Master's Programme in ICT Innovation SCI 3020

A lean UX Framework for creating a reusable product from start to finish

Writer: Nicolas Uggla, BSc
Supervisor: Mika P. Nieminen, DSc
Advisor: Tuukka Mustapää, MSc

Master's Thesis
30.7.2021

Acknowledgements

First off, I would like to thank the team at Contiot, that provided me with this opportunity, and the work that we conducted during the process. I would also especially like to thank Katri Hietala for the guidance in my work and Tuukka Mustapää for guidance and feedback on the thesis.

I also want to thank my friends and family. My girlfriend for the emotional support, my mother and brother who supported me through the process and gave me notes, and a special thanks to my grandmother, who went through the thesis multiple times and gave me notes and helped me with my grammar. Further many thanks to my friend "Baba", who gave me feedback and direction through my writer's block.

Finally, I would like to thank my supervisor, Mika Nieminen for his feedback, advice and guidance. Harsh truths can be hard to hear, but at the end of the day they need to be said. I also apologize for my early versions of the thesis, not to mention my long and wordy style of writing. Nonetheless, you still managed to give me useful feedback to get me through it.

Otaniemi, 30.7.2021

Nicolas Uggla

Author Nicolas Uggla

Title of thesis Lean UX Framework

Programme Master's programme in ICT Innovation 3020

Major Human Computer Interaction and Design

Thesis supervisor Mika P. Nieminen, DSc

Thesis advisor(s) Tuukka Mustapää, MSc

Collaborative partner CONTIOT

Date 30.7.2021 **Number of pages** 74 **Language** English

Abstract

Lean UX is a methodology that focuses on speeding up the UX process by getting rid of unnecessary elements, introducing new ways of working and thinking outside the box. The motivation behind this thesis is the lack of frameworks for Lean UX development as well as bridging the gap between design and development. This is done through introducing a framework for Lean UX development, that focuses on reusability throughout the process.

To create the framework, I conducted a literature review on Lean UX practices and similar works with either processes or frameworks that target the development aspects of UX. The framework was implemented as part of a project at a company called Contiot. Contiot aims to digitalize the calibration process through creating a platform for creating, sharing and verifying digital calibration certificates. The goal of the project was to create the design for the final product. The framework was evaluated using a survey within the team consisting of seven members, and walkthroughs of the implemented framework were conducted externally for validation.

The literature review in combination with the company's mission resulted in a framework consisting of four stages: The Define stage for determining the users, their usage and the structure of the platform in the form of a sitemap, the Ideate stage for creating tangible solutions and testing different ideas, the Prototype stage for creating a representation of the final product and finally the Handover stage for creating a smooth transition from design to development.

Through the evaluation it became evident that the framework was perceived as useful within the team during the process. The framework provides a methodical approach to the whole process and creates a shared understanding within the team. Throughout the process it became evident that there is a tradeoff between reusability and speed, where the more reusable something is the longer it usually takes to develop, which is important when considering creating different artefacts both in the long and short term. Turning the design into code proved more difficult than expected, and the current state of design to code software still need more research.

Keywords: Lean UX, Frameworks, Startup, Design tools, Design to code

Table of Contents

1 Introduction 1

1.1 Motivation 2

1.2 The project and the calibration industry 3

1.3 Introduction to the UX tasks 4

1.4 Scope and delimitations 5

1.5 Research questions 6

2 Background 9

2.1 Lean UX 9

2.1.1 Design thinking 9

2.1.2 Lean startup 11

2.1.3 Agile development 12

2.1.4 Defining and implementing Lean UX 13

2.1.5 POCs vs Prototypes vs MVPs 14

2.2 Frameworks 15

2.2.1 Lean UX frameworks for development 16

2.3 Bridging the gap between design and development 19

2.3.1 Design systems 19

2.3.2 Design to code software 21

2.3.3 The concept of reusability 23

3 Methodology 25

3.1 Research process 25

3.2 Framework development 26

3.2.1 Structuring the framework 26

3.2.2 Creating the different stages of the framework 28

3.2.3 Choosing the right tools to execute the methods and create the artefacts 34

3.3 Evaluation of the framework, artefacts and design to code software 35

4 Results 38

4.1 Construction of the framework 38

4.1.1 Define stage 38

4.1.2 Ideate stage 43

4.1.3 Prototype stage 46
4.1.4 Handover stage 49
4.2 Framework & artefact evaluation 51
4.2.1 The teams experience with UX 52
4.2.2 The scales of measurement 54
4.2.3 Define stage results 54
4.2.4 Ideation stage results 56
4.2.5 Prototype stage results 57
4.2.6 Handover stage results 59
4.2.7 Framework and outcome results 61
5 Analysis 65
5.1 What characterizes a good Lean UX framework? 65
5.2 What are the current best tools and how can they be implemented into the Lean UX framework? 67
5.3 How well does the chosen solution perform and how could it be improved? 68
6 Discussion 73
6.1 Limitations 73
6.2 Future work 74
7 Conclusions 76
References 78
Appendix 85
Framework & Artefact Survey 85

1 Introduction

User experience can be defined as a person's perceptions and responses that result from the use or anticipated use of a product, system or service (ISO, 2008). Law *et al.* (2009) agrees with ISO's definition, but admits it still needs further refinement. According to Vieira (2020) the concept of user experience and UX designers could be as old as 6000 years, dating back to Feng Shui and the importance of space. The term UX was coined by Donald Norman in 1993. Donald Norman was a cognitive scientist at Apple Computer, and after coining the term became the first person with the title of UX designer (Stevens, 2019). Even though the concept of UX might be thousands of years old, the newly defined science is still in its infancy when it comes to technical application. Vieira (2020) predicts that the field of UX design will continue to grow rapidly and by the year 2050 there will be a significant difference from what the industry looks like today. It is by no surprise that the field has such an expected growth due to the recent traction and benefits that companies can receive by investing in the satisfaction of their customers.

Some of the known benefits are increased customer acquisition and revenue generation on sites, optimized resources, insight from user engagement and reduced costs by avoiding errors (Partners, 2019). Companies failing to see the gap between their current UX practices and those required for growth driving UX tend to lead to simply more replication. This counteracts one of the fundamental principles of UX, which is innovation (Hokkanen, Xu and Väänänen, 2016). With the growth in the industry, designers will have to grow with the field to stay relevant. The typical role of the designer will become more connected with other disciplines such as coding, analytics and more. This means that there will be more job opportunities for UX designers, on top of current ones such as UX research, visual design, interaction design and frontend design (Sviryda, 2019). Other sources highlight even more responsibilities such as data analysis and even going as far as sales, because UX responsible people also have to justify UX tasks in order to sell them (Larusdottir et al., 2018).

However, when working at a startup, things take a different perspective. A startup is a young company founded by one or more entrepreneurs to create a unique (Fontinelle, 2020) or irresistible (Baldridge, 2021) product and bring it to market. Startups usually begin as shoestring operations, limited in both funding and resources (Fontinelle, 2020). Since there is no exact definition of startups, Wilhelm (2014) proposes a rule. The rule limits startups to number of employees, annual revenue and total worth in order to create clear criteria for defining what makes a company a startup. The rule is called the 50, 100 and 500 rule. The 50 stands for 50 million in revenue run rate for 12 months, the 100 stands for number of employees and lastly the 500 stands for total worth, however one wants to measure it. Overcoming any of these measurements would mean that the company is no longer a startup.

However, Cockayne, (2019) proposes another point of view, that the concept does not need to be so limited and dogmatic that it might not even be applicable to actual startups. Sometimes generalizability is a good thing. Startups, and especially very technical startups focus more on the technology of the product and making sure that the code works before considering any look and feel of the product. This leads to less resources and time allocated towards design of the product, meaning that designers working at startups will have fewer resources and more work to do, especially if they have to fill the role of UX alone. Mixing the different designer roles with a limited time frame and limited resources at one's disposal can make it a difficult task to manage without burning out.

Luckily there are several remedies concerning the issue. Lean UX is a recent methodology that combines elements from design thinking, development and startups (Aarlien and Colomo-Palacios, 2020; Gothelf, 2013). It makes a capable methodology of working in challenging and changing environments in terms of resources and time. Lean UX focuses on getting the user in the center of development, involving the development team in the process and getting rid of heavy documentation, processes, tasks and deliverables that do not produce enough value (Gothelf, 2013). Some of the benefits and goals are optimized work, improved internal communication and developing products with good enough market fit (Aarlien and Colomo-Palacios, 2020).

To ensure the success of lean UX adaptation, frameworks can be a powerful tool. Frameworks can be described as skeletal structures created to support or surround something (Definition of framework | Dictionary.com, n.d.). Frameworks give an initial structure that can be used and adapted when faced with new situations. There are many existing frameworks when it comes to UX. These frameworks target areas like user research, ideation, prototyping, and even in some cases the development process (Navarro et al., 2016). This shows how important frameworks can be when it comes to successful UX implementation. Frameworks give structure to the process with separate stages and milestones that can be followed along. In UX there are only few frameworks that target the development aspect of the work. According to companies adopting web design frameworks, ease of use and code generation abilities are some of the most important criteria (Almeida and Monteiro, 2017), highlighting a growing need for UX frameworks for development.

1.1 Motivation

Industrial motivation: There seems to be high demand for UX designers on the market, and especially in startups. To stay relevant designers must have a broad skillset and be highly productive to meet the demands from the industry. This thesis introduces a framework based on modern methodologies that have a proven track record as well as different tools

that might be expected at workplaces. The thesis also provides guidelines on how to use and improve the framework either through my personal application or in form of recommendations. The framework should be easy enough to understand and use for new designers but be so advanced that experts can make use of it as well. Creating such a framework allows startups to speed up the UX and development process as well as enhancing their collaboration within the team. The provided tools and process could give a good overview of what to expect during the process, better decision making, a clear picture of the expected outcome and a focus on the things that truly bring value. This can lighten the burden and hopefully help with stress and prevent burnout from work.

Academic Motivation: Diving deeper into lean UX and frameworks, there seems to be a gap between different methodologies and frameworks and practical use cases, and the current tools being presented in different frameworks seem to be out of date. This thesis aims at bridging the gap between academia and industry. By analyzing methods and implementing the insights, improvements and guidelines can be created for future use and research. Updating the toolkit used in frameworks could potentially improve future research as new software can bring new capabilities that are previously unknown in current UX research. Such software includes everything from simple wireframing to code generation abilities. This thesis aims to find them, analyze them, and recommend the most suitable ones.

1.2 The project and the calibration industry

Calibrations are an important part of the manufacturing industry. To ensure the quality of produced products and goods, manufacturing companies must measure the conditions of the manufacturing process. This is done through instruments that can measure for example pressure, heat, humidity and other factors that can affect the products in the manufacturing process. The process of ensuring the correctness of these devices is called calibration. This is done through comparing them with a traceable reference or standard with known accuracy and adjusting them accordingly (*What is calibration, and how can you manage your calibration reports?*, 2019). The field of calibration is a vast market with everything from national and international measurement institutes to private contractors down to manufacturing companies. With so many different players in the industry there are millions of calibrations being done annually, most of which are done manually. This makes for an attractive target for digitalization. Finland was ranked third in Digibarometer 2019, which is a measure for 22 countries digitalization progress. The United States and Denmark were ranked number one and number two respectively (Ali-Yrkkö, 2019). This suggests that Finland is doing well in the evolution of digitalization. There are already larger

Finnish companies that are investing in digitalization of the calibration industry. However, in such a vast industry there is still room for smaller players to have their cut of the market.

Contiot is a research to business (R2B) project that started in October 2019. The project has funding from both Business Finland and Aalto University. The target of the project was to spin off from Aalto University to become a startup. The project has functioned in a way similar to startups, which includes their ways of working and their disruptive technology. The company's mission is: "*With our Digital Calibration Certificate (DCC) Solution you can create, view and manage calibration certificates digitally. It fully transforms your paper-based or pdf calibration certificate handling processes to a digital process and makes calibration information machine-readable, authenticated and securely transferred between partners*." (CONTIOT – Continuity for IoT Data, n.d.). The goal is to build a digital calibration certificate hub, called the "DCC Hub" where certificates and information can be shared by the key players of the industry and provides them with the necessary tools. In addition to this, Contiot plans to provide a digital uncertainty and identity module to decrease calibration errors and to improve the data integrity and to reduce costs for the key players by reducing misinterpreted measurement data. Currently the main target is the DCC Hub.

The team of Contiot consists of an operational side, including a development team of 4 members, a product owner, a project leader and a metrology expert. The other, administrative side takes care of administrative tasks and advisory. These are people that either work on research projects, or that have been working in the project from the very beginning but are no longer participating in the operational work. The latest addition to the team was me, working as a UX designer in the project of creating the DCC Hub. Contiot has plans to hire more employees when the design is ready for development.

1.3 Introduction to the UX tasks

The main goal of the project is to create the DCC Hub. The current stage of the project is a demo that the development team are working on. The demo works as an MVP (see page 14) that can be showed to the clients and tested by them. The goal of the UX work is to design and test the final product of the DCC Hub. The scope of the work is to have a prototype of the final design within a timeframe of six months, with some time reserved for thesis writing and finalisation.

The design of the DCC Hub consists of the following three subtasks:

Sitemap: A sitemap is a mapping of the different screens and functionality of the artefact (Ann, 2019). The sitemap is used to ensure that all necessary

information is gathered and that the flow of the screens makes sense from an overview perspective.

Wireframe: A wireframe is an initial depiction of the product concept. It contains the essential elements of the product, providing a clear idea of the layout, functionality and structure (Jaye, 2021). Low fidelity wireframes can be simple sketches on paper and high-fidelity wireframes can be created digitally, called digital wireframes. The digital wireframes can introduce more realism with added pictures and some interactivity (*What is Wireframing?*, n.d.).

Prototype: "*A prototype is a physical or digital embodiment of critical elements of the intended design, and an iterative tool to enhance communication, enable learning, and inform decision-making at any point in the design process."* (Lauff, Kotys-Schwartz and Rentschler, 2018). The prototype is essentially the goal of the project. It gives a design that can be tested and evaluated after which the final product can be developed.

As a basis for the study conducted in this thesis, the project leader and the product owner had worked on tasks related to UX. Buyer personas, use-cases of the DCC hub and models of how the DCC Hub would be used by the different players had already been created. This meant that a large part of the user research had been done and UX development was the next step. Understanding the current models and material as well as the calibration process were critical requirements before starting the UX work.

1.4 Scope and delimitations

Scope: This thesis proposes a lean UX framework for developing software products at a R2B or a B2B (business to business) startup. The research will take a practical approach to UX frameworks, going from discovery and analysis down to implementation. The goal is to cover the whole scope of the frameworks, not only analyzing from a theoretical perspective but also the application itself. This means that a very broad range of aspects must be analyzed, narrowing down the research very quickly, starting from UX methodologies, moving on to methods, to different software tools that can be used to execute the methods and finally analysis of how the chosen solution performed. Trying to cover such a vast research area and vast amounts of instances means that some depth might be lost, and information might be missed in each level of the framework, when going from theory to practice so quickly. However, since this thesis addresses the implementation of a framework such matters are to be expected, though they should also not be overlooked. The idea is not to create the perfect solution for one instance, but to create a good enough solution that can be modified and improved to fit new instances perfectly.

To target the issue of overlooking important information and keeping the scope on point, the research can essentially be divided into an

operational and a strategic side. The strategic side will consider research, planning and analysis, and the operational side will consider the application during the thesis work. These two sides will coexist along the thesis going back-to-back from research to implementation to analysis, repeating these until the framework is done.

The project revolves around startups, and in this case more specifically R2B and B2B startups. Already established companies and B2C startups will not be addressed. Nonetheless, the results will not necessarily constrain themselves to B2B startups, if factors can be generalized then I will consider it a positive addition, if it is also applicable to the other types of companies. Lastly, another important aspect in lean methodologies is reusability, the thesis aims to preserve reusability throughout the different methods and tools. This means making trade-offs against whether to use different tools based on the short to long-term reusability, potentially undermining the rapidness of lean methodologies. By this I mean that focusing on reusability especially in the long term might take longer planning sessions or time to create, but it might eventually be worth it.

Delimitations: The operational side of the thesis will focus on development and creation. This excludes evaluation methods with users which are at the very core of the concept of UX, but due to the limited timeframe and the scope of the thesis they will not be included. However, in the strategic analysis they will still be used for evaluation of the framework and the different tools and methods. During the project there were also specific tasks that had to be completed which limited the range of methodologies and methods to choose from. This also affected the tools on some level. These factors will be mentioned and further analyzed later in the thesis.

The current Covid-19 situation will only be mentioned here briefly as it has had an impact on this thesis, the case as well as working conditions in general. It has affected all of them in the same way, the way of working is now remote. For this thesis it affected the chosen methodologies, favouring online solutions. For work it meant working from home, which both has its benefits and drawbacks that I will not go deeper into. Lastly in terms of UX work and in my case, this meant choosing tools and methods that would work better online. It remains to be seen if the current situation has changed the way humans work. It is likely that online tools have a better future no matter how the situation will advance.

1.5 Research questions

Creating a fully-fledged framework, it is important to take a top-down approach, first looking at the big picture of what frameworks are and what they can do. When a sufficient understanding has been developed, a deeper analysis of different areas of the framework can be executed. The research starts from methodology and methods in the first research question and

will then move on to the actual tools being used in the second research question. Finally, when the framework has been deployed and executed the final research question attempts to analyze the results of the framework, giving insight into how well the framework performed and how it can be improved.

Research Question 1: *What characterizes a good Lean UX framework?*

To understand what makes a good UX framework, different kinds of frameworks must be analyzed to find out how they are being used to have something to benchmark against. Then an understanding of the structure of a new framework and its different stages and parts can be understood. To answer the question, I propose the following sub questions to address the complexity of the question:

1a What kind of frameworks are currently being used in Lean UX?

1b What are the different stages in a Lean UX framework?

1c What are the different methods and steps to take in each stage?

Research Question 2: *What are the current best tools for UX designers and how can they be applied in the framework?*

As this thesis takes a highly practical approach it is important that not only the methodologies and methods are analyzed, but also the tools that bring them to life. Another reason for this research question is that Academia lacks research of different tools used in the industry and analyzing the current best tools is a way of contributing to academia.

This research question is split up into the different areas of the framework. I analyze the different tools separately in the following way where each section is a different phase of the design process that required different tools or ways of producing results:

2a What is the best tool for the define stage?

2b What is the best tool for the ideation & prototype stage, and can they be combined for efficiency?

2c What is the best tool or solution for an optimal handover?

Research Question 3: *How well does the chosen solution perform and how could it be improved?*

Finally, after the framework has been designed and applied to the project the results can be measured and analyzed to gain a deeper understanding of why the framework performed as it did, and if there is room for improvements. This question is especially valuable, because it can give feedback and insights from practical application in the industry

and measure the significance of the artefacts created in the framework, to help in improving the framework for future use.

To answer the research question, several sub questions are proposed to analyze the framework and to measure the performance of the artefact:

3a What is the perceived usefulness of the produced artefacts?

3b What is the perceived usefulness of the framework overall?

3c How can the framework be improved?

2 Background

In this chapter the theoretical background of the areas of interest will be analyzed. Key concepts will be introduced and defined to be used and discussed later in the thesis, starting off with Lean UX, which is a new and broad area. The concepts that Lean UX builds upon have already existed for a while, which means that there are clear definitions of the sub genres of Lean UX. These sub genres have both academic and industrial applications. These will be investigated to find out the current practices and standards as well as a clear definition to what Lean UX is. Frameworks in UX will then be studied to find what other practitioners and researchers have been doing to be able to benchmark the framework of this thesis. However, due to the lack of scientific material about Lean UX frameworks, the material will include some popular sources such as web articles from professionals in the industry as a base for the research and the decision-making process. Lastly the background of bridging the gap between UX design and development, including the concept of reusability will be examined to provide a basis for the analysis and discussion on design to code systems and software.

2.1 Lean UX

The methodology of Lean UX was born out of several other methodologies that evolved out of the fast-changing environment of software development. Previously design processes had been largely unchanged and followed long and rigorous processes due to costs of failing being high (Gothelf, 2013). However, this has changed due to a new reality where products can be distributed online much faster, and the manufacturing process can be done practically anywhere. Teams can now work in shorter cycles, learning as they go on (Gothelf, 2013). Lean UX consists of three methodologies, Design thinking, Lean startup and Agile software development (Gothelf, 2013; Liikkanen *et al.*, 2014). To understand exactly what Lean UX is, these three methodologies must first be defined.

2.1.1 Design thinking

Design thinking is an iterative process where knowledge is constantly being searched for and questioned. The goal is to redefine the problems and understand the users. Design thinking can also be used to define strategies and solutions that might not otherwise be apparent. Design thinking can even be seen as a way of thinking and working, as well as hands on methods. (Dam and Siang, 2020). Design thinking also solves the problem of ingrained thinking patterns. Humans develop patterns based on repetitive activities, which helps to quickly use them in familiar and similar scenarios. The problem can be, that they prevent us from easily

developing new ways of discovering, understanding, and solving situations. This is why it is also called thinking outside of the box, as it means thinking outside of currently developed thinking patterns (Dam and Siang, 2020). The five phases of design thinking are in nonsequential order empathize, define, ideate, prototype and test (Dam and Siang, 2021), with an additional stage implementation (Gibbons, 2016), as seen in Figure 1.

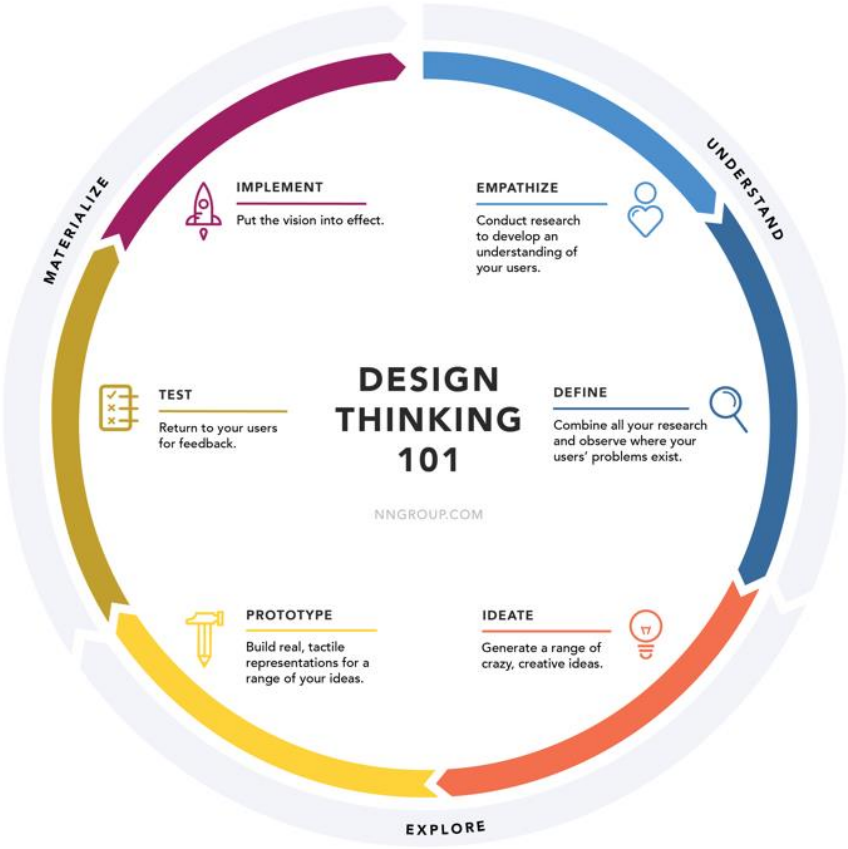

Figure 1: The 6 stages of design thinking presented in the Nielsen Norman groups article design thinking 101 (Gibbons, 2016)

Empathizing with users is crucial in user centered design. It allows designers to set aside their assumptions to get insight into users and their needs (Dam and Siang, 2021). This stage is about conducting user research into what users say, think and feel (Gibbons, 2016). The **define** stage is about pooling together the research to find problems, user needs and insights to form an understanding (Gibbons, 2016). In the **ideation** stage, generation of new solutions happen with complete freedom (Gibbons, 2016). The solutions are based on the research from the previous stage (Dam and Siang, 2021). With the understanding or problem statement definition from the previous stage, the team can start thinking outside of the box to identify a solution. In the **prototype** stage, tactile representations of the final product are built to understand which parts of the solution that are feasible (Gibbons, 2016). In the final **test** stage, the

prototype is rigorously tested, and the insights gathered from feedback can then be corrected or if it is an iterative process, it can be reused in any of the previous stages of the process.

The processes as described by both Gibbons (2016) and Dam and Siang (2021) are nearly identical. Yet, in the 6-stage process as described by Gibbons (2016), a final stage of implementation is included. Though, this I will not describe any further as it is not well described in his article. It can also be seen as redundant, because in any of the steps there will be implementation of the results from the previous stage.

2.1.2 Lean startup

Eric Ries introduced the methodology of Lean startup in 2011 (Edison, Wang and Abrahamsson, 2015). The methodology originates from the car manufacturing company Toyota's manufacturing environments. They managed to increase value while simultaneously reducing waste in their processes (Cyrillo, 2011). The methodology was designed to improve teams, reduce waste, optimize the process and keeping customers in the center of the decision making process (Aarlien and Colomo-Palacios, 2020).

Lean startup follows the 5 principles mentioned below (Edison, Wang and Abrahamsson, 2015):

Entrepreneurs are everywhere: The first principle highlights that anyone can be an entrepreneur, no matter in which busines they work in or they type of work that they do.

Entrepreneurship is management: The second principle is that startups are not only about product development but also about business development.

Validated learning: The third principle is that anything being developed should be validated by customers, otherwise there would be no need to develop it.

Innovation accounting: To improve outcomes, the real innovation must be measured empirically.

Build, measure, learn: Lean startup follows a repetitive 3 step process, build, measure and learn. Build prototypes and minimum viable products (MVPs) with rapid testing. An MVP is essentially a product consisting of only the core features needed to make the product work (Minimum Viable Product (MVP) and Design - Balancing Risk to Gain Reward, 2020). The term has been around for some time but was recently made popular due to two influential names, one of them being Eric Ries, the inventor of the lean startup methodology. The MVP can then be tested and evaluated for insights. The insights can then be implemented in the next iteration of the build, measure and learn process.

The idea of Lean startup is about improving the process as it goes on. Instead of focusing on a heavily documented business plan, it focuses on building a product quickly and shipping it to market for early learning (Edison, Wang and Abrahamsson, 2015). Ultimately the goal is to produce products as quickly as possible with minimal resources that still maintain enough quality to satisfy customer needs (Liikkanen et al., 2014). In recent years lean methodologies have started to become more common in software development environments under the term agile, which will be introduced in the next section.

2.1.3 Agile development

In the 1990s agile methodologies started getting traction in software development. Agile processes focus on speed, communication and collaboration by delivering functioning software early on in the process and building it continuously (Larusdottir et al., 2018). As seen in figure 2, the process of the agile looks very different from more traditional waterfall methodologies that were used before.

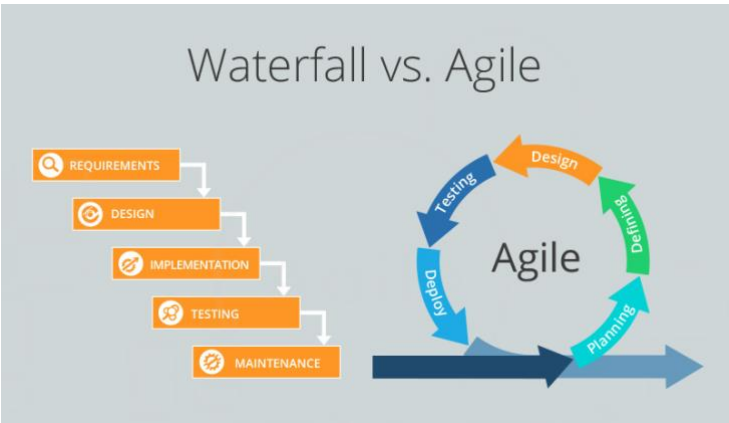

Figure 2: Comparing the waterfall and agile methodology side by side. (Singh, 2019)

In agile the process is done iteratively in repetitive short periods of time called sprints. Cross functional teams work on different parts of the solution at the same time, only focusing on that which brings the most value (*Effectiveness of Agile Compared to Waterfall Implementation Methods in it Projects: Analysis Based on Business Intelligence Projects*, n.d.). Traditional methods differ from agile in that they are done incrementally, which means that when one step is complete the process

moves on to the next stage. When this process is planned the document can look like in figure 2, as that of a waterfall. The problem with waterfall methodologies is that if one step in one increment takes a very long time or goes over budget, the other steps beneath it will be put on hold, which makes the process very slow and very reliant on previous steps being executed correctly.

Lean UX applies four of the core principles of agile development (Aarlien and Colomo-Palacios, 2020; Gothelf, 2013).

Individuals and interactions over processes and tools: To generate solutions quickly it is important to engage the whole team. Discussions should be held freely and often (Aarlien and Colomo-Palacios, 2020).

Working software over comprehensive documentation: This point essentially highlights that everyone will have different opinions and ideas on how to create the solutions. The challenge is to figure out which one is viable. The solutions can be evaluated sooner if they can be tested (Aarlien and Colomo-Palacios, 2020).

Customer collaboration over contract negotiation: Collaborating with customers and within the team creates a shared understanding of the problem space, this will make iterations faster and relieve the process of heavy documentation. (Gothelf, 2013)

Responding to change over following a plan: This principle embodies the saying "to fail often and early". Essentially it means that the initial assumptions are often wrong and they should thus be evaluated in order for us to change them in the right direction, instead of following through with assumptions that turn out to be wrong when it is too late. (Gothelf, 2013)

2.1.4 Defining and implementing Lean UX

From the previous research it becomes clear that the sub methodologies of Lean UX share many of the same traits, as described in the previous sections. There are however a few principles and aspects that stand out from each of the methodologies. Design thinking is about a mindset for thinking outside of the box as well as a process that includes users and ideation (Dam and Siang, 2020). Lean Startup is about discounting aspects of the process that do not bring enough value (Edison, Wang and Abrahamsson, 2015). Finally, agile is about improving the way of working through short sprints with high involvement from both the team and customers and evaluating the process to allocate resources to the areas that need them the most (Bruun *et al.*, 2018).

Nonetheless, an exact definition of Lean UX has not yet been formalized and there remain many different definitions coming from various sources (Cyrillo, 2011). Gothelf (2013) also restrains from providing a dogmatic approach to practicing Lean UX. This allows for some own interpretation of the area when practicing and implementing

Lean UX. According to Cyrillo (2011) following lean UX principles when implementing a UX strategy means that you are doing lean UX. The argument is that even using only some lean principles still makes the UX process leaner. However, the main specification is that everyone works together to solve the problems, especially designers and developers.

2.1.5 POCs vs Prototypes vs MVPs

In one of the previous sections, it was mentioned that in Lean startup, prototypes and MVPs are built for rapid testing. The words prototype, MVP and POC (proof of concept) are often used in the startup environment. These words are often used interchangeably, which can result in confusion. Therefore, it is important to know the difference between the words before diving deeper into product development in a startup.

An **MVP** can be defined as the least amount of things that you can build into a product that still gives the customer value, with a bonus of receiving value back (*Minimum Viable Product (MVP) and Design - Balancing Risk to Gain Reward*, 2020). Another definition from Eric Ries lean startup is: "*a version of a new product which allows a team to collect the maximum amount of validated learnings about customers with the least effort.*" (*Minimum Viable Product | Design Defined | InVision*, n.d.). Both definitions have in common the learning aspect and to bring customers value. However, giving customers value would require actual features to be built and coded. Simply creating a few mockups cannot bring customer any value unless it gives them joy from future hopes. This means that an MVP is something that customers can use to bring them value, in contrast to a prototype.

Defining a **prototype** can be even more challenging than an MVP as there is just as many definitions, but there are also different types of prototypes based on their resemblance to the product. This is called the fidelity of the prototype, of which there is a scale from low to high fidelity. A study was conducted to explore prototypes and included finding out a rigid definition for prototypes by Lauff, Kotys-Schwartz and Rentschler, (2018). The definition goes: "A prototype is a physical or digital embodiment of critical elements of the intended design, and an iterative tool to enhance communication, enable learning, and inform decision-making at any point in the design process.". This means that the prototype can be just about anything, from paper mockups to digital prototypes to even coded prototypes if it resembles the intended product. But it becomes clear that the prototype is not to be mistaken for the real product.

A **POC** is similar to MVPs and prototypes in that it helps save time and money and help verify assumptions, but they serve a slightly different purpose. POCs are often built to verify minor technological assumptions, in contrast to MVPs they often concern a small part of the whole system and they are verified internally (*MVP vs POC vs Prototype: What Does*

your Company Really Need?, 2018). Comparing POCs to prototypes, the POC is just a minor feature that is being tested, while a prototype enables multiple features such as design, functionality and usability to be tested. With a POC all of that is not possible. POCs are common in startup development because they allow the team to quickly validate assumptions.

As can be noticed these terms are all quite similar, but there are minor differences, and they all serve different purposes. To summarize, they can be ordered in the scale of readiness, an MVP is the actual product that can bring value to early users with real functionality. A prototype can look even more like the finished product than the MVP, but there is no real functionality which means that it cannot bring customers any value right now, but it can bring value to customers in the future in form of learning and insights. The usefulness of the prototype comes from bringing insights before something is included in the actual product, this is crucial for faster development because features are validated even before they are coded. A POC is just a small research project, and it can be about verifying a small function or feature in the MVP or prototype. Using both a prototype and an MVP means that teams can work on design and code at the same time.

2.2 Frameworks

As mentioned before frameworks are skeletal structures that are used to support something. Frameworks exist to provide direction and structure without being too rigid in contrast to processes (Ellis, 2008). The power of frameworks comes from providing guidance with flexibility and creativity (Gardner, n.d.), which means that it can be adapted to new situations and customized for specific needs (Ellis, 2008). Frameworks serve as more of guidelines where each part does not have to be included, only the ones deemed necessary. Especially for smaller organizations, frameworks can be more useful than methodologies as they leave room for more creativity and flexibility (Dafir, 2016). In the context of UX, to understand the structure of a framework, it is described as things that can be abstracted from a process. Using coding methodology, the method DRY can be used, which stands for Don’t Repeat Yourself. Things that are being done over and over can be consolidated into one location (Croft, 2007). Essentially this is the idea behind creating a framework, finding repetitive patterns or other aspects, and making an initial structure that can be used for future implementation. The real world benefit should also be considered when looking to include factors into a framework (Croft, 2007).

It is necessary to use frameworks and tools that are flexible enough to meet the specific team’s needs. Teams are less likely to start an activity if the presented tool is too complex, thus it is recommended to leverage tools that are already known, similar in nature or that are easy to learn (Krout, Carrascal and Lowdermilk, 2020). Some of the benefits of using

frameworks are added structure to processes, guidance in the design process and flexibility and innovation (Batterbee, 2020), repeatability and reduction in time and resources (Justinmind, 2018). Even though using a framework seems to generate mostly positive outcomes there are a few challenges such as hindered creativity and the actual creation and implementation can be time consuming if not done right.

There seems to be very few scientifically researched frameworks and processes that address Lean UX and UX development frameworks. Currently, most frameworks address the UX research process on how to discover, gather, and measure data. Though the frameworks are often created for these areas, the UX development aspects are just as important. These are the aspects that involve creating artefacts based on the gathered user data. There are very few frameworks that go down to tool level, and if they do, they do not recommend publicly available tools. This shows that there is a research gap for Lean UX frameworks, especially ones for the development aspects of UX.

2.2.1 Lean UX frameworks for development

Even though the area is to my knowledge under researched, some material could be found that matches the scope of the thesis. Two frameworks were found that touch on different aspects of the scope for creating a Lean UX framework for the development aspects of Lean UX. The first framework gives an overview that focuses on the UX process, using the stages of the design thinking process (Martins et al., 2020). The second framework attempts to solve the disconnect between designers and developers during the development phase (Navarro et al., 2016). In this section these two frameworks will be studied as a benchmark for future development of the framework presented in this thesis.

Lean UX Process by Martins *et al.* (2020)

Martins et al. (2020) propose a process based on the development of a Portuguese e-commerce platform. The process follows the 5-step design thinking process to create the product by including three of the five stages: define, ideate and prototype. As the process focuses on creating the visual solution it was still important to define the target audience through creating archetypes and a structure of the website. The structure and the archetypes inspired the creation of user flows and diagrams showcasing how users interact with the prototype. Then wireframes were created to show the visual outline of all the pages. Lastly two prototypes were constructed for both the buyer and the seller side of the platform. The prototypes were created in two stages, first a low fidelity prototype to simplify the flow and identify potential usability problems, then a high-fidelity prototype for the final design. The process also includes various forms of testing and evaluation of the artefacts that were constructed in

the process. As mentioned before these will be out of scope for this thesis as it focuses on the creation aspects and not the testing aspects of UX.

Design for development framework by Navarro *et al.* (2016)

The second study by Navarro et al., (2016), creates a new framework for generating UX artefacts in a way that focuses on reusability and designing as well as using tools that help in coding the product. In this approach prototypes are used to successfully include UX in agile development, but it is clearly noted that it is not without ease. The process is of an iterative nature, where feedback is gathered and integrated, thus the process is not entirely straightforward. The framework uses scripting languages to code prototyped behavior by designers. This means that prototypes are reused and iteratively incorporated into the final design, while the prototyped behaviors are removed. The framework improves communication between designers and developers because they can work on the same artefact at the same time.

UX used to be considered expensive in terms of time and human resources in some cases, which is why it usually was not considered crucial, or it was considered too expensive in Agile development. However, during the recent years the evolution of agile UX shows that UX can be implemented into the agile process. Prototypes are described as a good way of doing so, even though it comes with some difficulties. Designing software requires close collaboration between designers and developers, which is difficult because there are different tools for UX and coding. This makes it harder to reuse material produced in the UX process. Integration of different stakeholders and team members needs to be easy. Therefore, programs should be so simple that different stakeholders can take part in the process, this is called codesign. The reuse of artifacts is often not possible or neglected, after designs are evaluated, they are often thrown away, which wastes time and the resources spent on the artefact. Thus, digital prototypes should replace low fidelity prototypes because reuse of prototypes needs to be improved. The process of the framework can be seen in Figure 3.

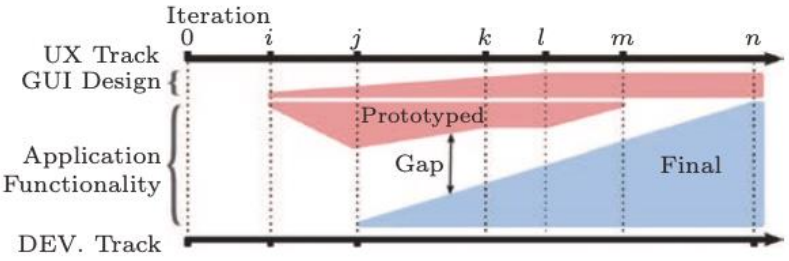

Figure 3: An overview of the development and design process going in parallel to each other. The approach is called parallel tracks.

Navarro *et al.* (2016) presents a framework called SIBAP or Script-Based Aspect-Oriented GUI Prototyping. The framework suggests reusing prototypes from the very beginning until they become the final product. The prototype becomes a common artefact for designers and developers

where their efforts are integrated. The main goal of the framework is to force communication between designers and developers by reusing the design when developing.

In **iteration *0* and *i*** requirements are gathered, and the first prototypes are made. In the following **iteration *j***, one prototype is chosen, and the development team starts making it. At the same time the UX team is refining features. In **iteration *k*** designers can test new behaviors. In **iteration *l***, prototyped behaviors are removed from the graphical user interface (GUI). In the second to last **iteration *m***, once all prototyped behaviors are removed, SIBAP is also removed from the framework. In the final **iteration *n***, the UX team tests the product with the end users, while the development team fixes glitches. In the beginning of the process the artefact is called the "prototype" and at the end it is called the "application".

The framework is introduced as a POC, meaning it is a small research project. To be able to use the framework designers should be familiar with scripting tools and possess advanced coding skills, compared to other tools where no coding skills are required. The framework proved beneficial in several ways. The framework enabled the team to produce cleaner code, improve communication when developers code in native functionality and enabled designers to create and test prototyped behaviors on the same artefact as developers.

2.3 Bridging the gap between design and development

There have been a few previous attempts at bridging the gap between design and development and not just in terms of collaboration, but also in designing for code or transforming designs into code. These attempts have been made to make the process easier for turning the design into working code. The two approaches are design systems and design to code software, which can be used together.

2.3.1 Design systems

Design systems are currently a trend within UX. As component based software architectures have emerged as well as agile development, a need for design systems was born (Vesselov and Davis, 2019). Design systems give an overview of the structure of the product, goals and vision. Design systems provide a visual representation of the interface in several stages, making it easier for the whole team, especially engineers to find their way through the design files (Vesselov and Davis, 2019). When finally large corporate organizations such as Google and Apple began adopting design

systems, the concept finally matured to the standard that can be seen today (Beck, 2017).

Design systems can be described as a collection of documented elements, components and regions that include both design and front end guidelines with underlying design rules and principles that help the team build products (Vesselov, 2019). Therefore, design systems can be seen as a collection of the whole style of the company's design with the purpose of bringing scalability, structure and giving a helping hand to the development team through instructions and style guides. A design system is a concept that is based on pattern libraries and atomic design (Beck, 2017), which will be introduced in the following sections.

Pattern libraries are collections of UI elements (Leeson, n.d.). Pattern libraries are often part of a larger design system, but they can be used independently. Pattern libraries ensure consistency in UIs, improves reusability and make maintenance easier (*How to create a pattern library and why you should bother*, n.d.).

Atomic design takes the concept of our chemistry and applies similar principles to web design, where elements can be broken down into smaller entities, like our human bodies (*Designing Systems | Atomic Design by Brad Frost*, n.d.). In atomic design, web pages can be broken down into the five following categories seen in figure 4, atoms, molecules, organisms, templates and pages.

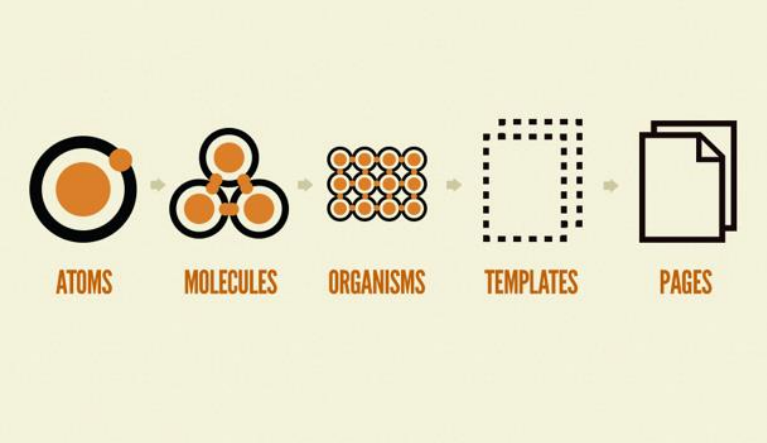

Figure 4: The five building blocks in atomic design, going from left to right the smaller elements are combined to make larger entities. (*Designing Systems | Atomic Design by Brad Frost*, n.d.)

Atoms are the smallest entities, these can be simple elements like buttons, inputs or other elements that serve minimal purpose and cannot be broken down any further. Molecules are simply several atoms grouped together. They can be for example a text input field combined with a search button. Organisms are more complex than atoms and molecules. They consist of collections of molecules and atoms, and even other organisms. Organisms

can be a search form consisting of several molecules and atoms. Templates and pages do not follow the same structure as the previous instances of atomic design. Templates and pages differ from one another in the way that templates focus on the content structure rather than the final content. Organisms can be applied to a specific template to give a specific structure. Finally, pages show what specific templates look like with real content in place. An example can be a template with real images and text that show the content in action. Pages are the best way of testing the underlying design system, if something fails, it is easy to go down the hierarchy and fix specific parts of the design system. (*Designing Systems | Atomic Design by Brad Frost*, n.d.)

2.3.2 Design to code software

During recent years new tools have emerged for converting design into code to minimize the gap between design and front-end development. Yet, existing conversion tools don't have mechanisms to generate low code artefacts, which are ones that can be used within low code platforms. Low code platforms are coding platforms with a graphical user interface where the developer can create artefacts without using code or using very little code. An issue with these tools is that they do not have a connection with design tools which makes for an inefficiency in the design to code process. Design tools such as Figma and InVision do have code generation abilities, but they are not used by professional teams as they do not export low code web technology. (Bexiga, Garbatov and Seco, 2020)

Luckily, there are other tools that can be used for this issue. These tools are used to export the designs from design tools as developer friendly code into the design to code (D2C) tool. One of the tools that has made progress during recent years is Anima, which provides a design to development platform for tools such as Figma and Adobe XD to export designs as code (Bekyarov, 2021). Anima was created by designers, developers and product managers, who understand the struggles that come with the handover process. Instead of seeing ideas materialize, too much time is spent on explaining every step to one another (Anima App, 2021). Anima aims to optimize the handover between designers and developers by helping designers create responsive prototypes with additional features that regular prototyping tools cannot offer. In figure 5 (next page), a component is created, and a text input functionality is added through the Anima plugin for the prototyping tool Figma. Auto layout is also added to the component which makes it responsive, which means that the component responds to changes made to it or during interaction.

Figure 5: In the upper left corner is a design component from the project. Below the component is a screenshot of how it looks in the layer panel of Figma. To the right is the editor panel of Figma, with auto layout enabled.

Developers can then export these designs through the Anima plugin to the Anima platform where they can access the designs in either play or code mode (Bekyarov, 2021). Play mode is similar to regular prototyping showcases, where you can interact with the prototype, the code mode enables the inspection of the code. The code can be exported in either React, which is a JavaScript and user interface (UI) library and tool built for creating components, or regular HTML. The functionality of the React code can be seen in figure 6 with the styling underneath in the figure. This makes for a better starting point for development where they can either use components entirely or copy and paste certain elements from the exported code.

Figure 6: In this figure the CSS and JSX code of the component from the previous figure has been exported to Anima App. To the left is the CSS code (cascading style sheet) and to the right is the JSX code, which allows HTML to be put into JavaScript.

2.3.3 The concept of reusability

Reusability is very often mentioned in many of the works in Lean UX and frameworks. Yet, the application of reusability is never fully explained, which seems to mean that a complete definition or instructions do not exist. For example, frameworks in their very nature are reusable. Once a framework is created it can be adjusted and improved to a new process which means that the framework is being reused. However, it is not the reusability of the framework that will be considered in this section, but the reusability of design artefacts.

Reusing a framework is quite straightforward, while reusing an artefact can be quite challenging. For example, Navarro *et al.* (2016) proposes a framework where designs can be reused for development, but never fully explains how it is done. He mentions that designers and developers can work on the same artefact, but how it is done is not detailed enough to give a proper understanding. It could mean that designers know how to code, thus they are working on the front end of the artefact, but it could also mean that they are working on design at the same time as developers are working on functionality. In terms of pure reusability, it would mean that designs can be turned in to code, meaning that the very same designs are completely reused. However, this is doubtful. There is clearly some vagueness to the term that needs addressing.

Reusability in software engineering is somewhat more tangible and easier to understand. It was discovered in programming that grouping together data with functions that work on the data made the code clearer and easier to understand. This eventually became known as object-oriented programming. The fundamental principle is that programs are being designed around the data that is being used. This leads to flexibility and reusability where programmers can create modules that can be reused for several things and the modules can be changed without affecting the rest of the program (Yevick, 2005). Another example of reusability in software development is atomic design, where smaller components or sections of code can be grouped to form larger elements, where the same small elements can be used multiple times (Saring, 2020).

The definition of reusability is very broad (Pakkanen *et al.*, 2016). Design reuse can be explained as building new applications and tools by reusing already developed designs, including logic and data. The logic and data can come in form of components such as code segments, structures, plans and reports. The goal is to help developers build better products. A prerequisite is to make designs reusable and store them in a way that they can be easily found and used. A design for reuse model was introduced by Duffy, Duffy and MacCallum (1995). The model focuses on reusability itself rather than methods and processes of reusability. To construct the model existing practices were researched and classified into three general processes, that are design for reuse, design by reuse and domain

exploration. Design for reuse is finding and extracting knowledge fragments and design by reuse is using existing knowledge fragments in new situations. (Pakkanen *et al.*, 2016). Domain exploration is as the name suggests, researching design domains to identify reusable fragments of knowledge that can be located, extracted, saved and then used to develop new designs. (Duffy and Ferns, 1998)

A standard principle is that the more components can be reused the quicker and better the software design is. When designing new systems it is essential to make decisions concerning reusability and adaptability (Netinant, 2013). Pakkanen *et al.* (2016) mentions some of the benefits of reusability, which are improved productivity, reduction in effort and risk, avoiding errors and uncertainty and it helps teams familiarize themselves with design. All of these in turn lead to cost reduction, quicker time to market, faster testing and better quality. Some of the drawbacks and challenges are time consumption in creating reusable parts, biases in creating components to be reused and case specific designs that are not standard designs (Pakkanen *et al.*, 2016). Hayes (2008) points out the reusability versus the usability, it is important to plan the reusable components well. Otherwise, if too many elements are created to be reusable it will be nearly impossible to keep track of them all, thus affecting the maintainability.

3 Methodology

The purpose of this chapter is to go through my research process, presenting step by step how data was collected, measured and analysed, as well as the decision making in the process. The process of using the data to structure it into a framework is presented, along with how the framework and artefacts were evaluated.

The objective of the research is to answer the research questions presented in the first chapter, which were to characterize what makes a good framework, the different methods and tools for the framework as well as evaluating both the framework and the produced artefacts for improvements.

3.1 Research process

The research goes hand in hand with the design process of the project at Contiot. The study takes an action research approach, where I as a researcher am not only an observer, but I also interact with the research object (Costello, 2003). In this case the research object is the framework itself. The environment of the study was the project of creating the DCC Hub. The end goal of the project was a complete prototype of the final product based on the business needs of Contiot. A design system with documentation of the design as well as libraries and guides were added to the end goal. The decision for adding the design system came from discussions with the development team and the project leader of how to create a smooth transaction for the design to the development team.

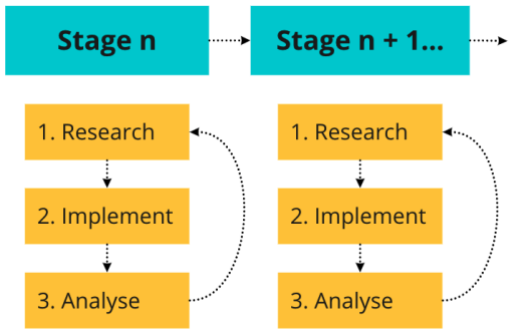

Figure 7: The research took an iterative approach where the same process was repeated for each stage of the process

The research of each stage in the framework followed the same structure, as seen in figure 7. Each stage was first researched, then implemented and lastly analyzed, before the same process began for the following stage of the framework. The research essentially follows an iterative waterfall structure, where the process repeats itself, and

knowledge flows from the previous stage to the next one. Each stage of the research and planning of the framework essentially functions the same way. First, there is selection of the right method, followed by selection of the right tool, after which the implementation is done and lastly an analysis and an evaluation is performed. User evaluations of the artefacts started in later stages of the framework, where concepts and artefacts were first evaluated in internally but as the fidelity grew higher external stakeholders were involved in the process.

3.2 Framework development

As the framework in this thesis was created based on the project work and the tasks at hand, a planning session of the process had to be done before starting the work. This was done through analyzing the design thinking process and selecting the steps needed to fit the different tasks that were provided. The planning was done together with the project leader to ensure that the planning of the UX process was in harmony with the rest of the company processes. As mentioned before the deliverables of the work were a sitemap, wireframe and a prototype.

It was decided at the beginning to keep the UX process separate from the development to not interfere with development which consisted of completely unrelated tasks most of the time. However, discussions and workshops were held including the development team in the beginning of the process. The time frame of the work was aimed at five months for the whole process from start to finish with time for adjustments at the sixth month. It was initially planned that each stage of the framework would take about one month, meaning that slight variations in time would not matter, as long as the project was kept within the 6-month time frame. The process of selecting the different steps to take in the process followed the lean startup principle of value. This meant that tasks, processes and other aspects that were believed to not bring enough value were discounted from the process, and time and resources were only focused on parts that could take the process to the next stage.

3.2.1 Structuring the framework

To structure the framework, I did a literature review to find out what principles and characteristics that were needed to create a Lean UX framework and analyzed existing frameworks and processes for inspiration and benchmarking. This allowed the framework to rely on and learn from previous experiences with a proven track record. The other important aspect of the literature review was to find out what kind of characteristics a good framework has and more specifically a Lean UX framework for development. This was done through analysis of the three subgenres of Lean UX which are lean startup, agile methodologies and

design thinking. Principles from these subgenres served as the foundations of the framework.

The structure of the framework was based on studies from the previous chapter, also drawing inspiration from the double diamond model, which has become a cornerstone in modern UX (Ball, 2019). I found two studies that were similar in scope and objectives. The first one was by Navarro *et al.* (2016), which is a framework for designing reusable artefacts at the same time as the development team to make the process faster and increase collaboration. The other similar study was by Martins *et al.* (2020), which was done on a design thinking process for designing an e commerce platform, where the product being developed was very similar to the product being developed at Contiot. Based on these two approaches I planned the design process with the help of the project leader who had some experience in UX design. The result of the planning can be seen in figure 8, which eventually ended up being the stages of the framework.

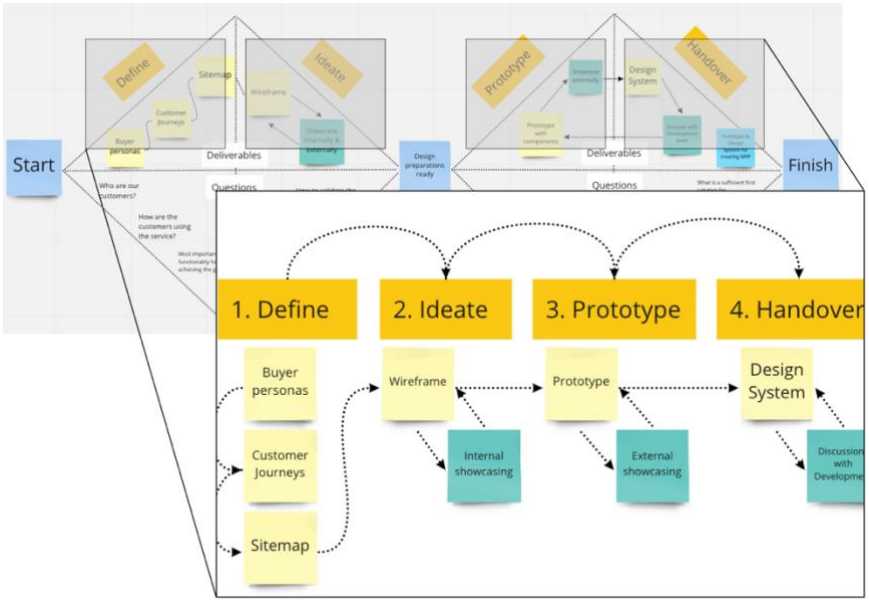

Figure 8: Overview & summary of the framework. The structure of the framework shares similarities with the structure of the Double diamond model. The summary consists of the stages and most important deliverables and methods for the respective phase.

The structure of this framework follows the design process proposed by Martins *et al.* (2020). This includes the steps in the design thinking process used for creation, excluding the first research stage and the last testing stage. Since the process did not include anything related to development a final step was introduced, that was inspired by Navarro *et al.* (2016), about handing over the designs to be implemented. The idea behind the framework was also to include the different deliverables as milestones in the framework to keep track of the process and help in guiding the next steps. These were integrated in the different stages of the

framework. The final structure of the framework thus ended up being **Define**, **Ideate** and **Prototype** from the design process including a final step called **Handover** for on bridging the gap between design and development.

3.2.2 Creating the different stages of the framework

Choosing the right methods can be challenging. However, in this case some of the deliverables were predetermined. As mentioned in the introduction the project consisted of creating a sitemap, a wireframe and a prototype for the DCC Hub. There were several methods that were implemented before reaching these outcomes. Creating these artefacts and the decision-making process is considered in this section.

Define stage

The end goal of the define stage is to form an understanding and form a problem statement. In this case the problem statement came in form of how the DCC Hub should work, and who the customers would be. The first deliverable was a sitemap, which would eventually help in answering these questions. However, forming the knowledge to be able to create a sitemap required some steps to be taken before it could be done.

There were many resources to go through before an understanding of the industry, product and customers could be made. Choosing the methods and tools for this stage of the framework meant that several artefacts had to be created and they needed to bring enough knowledge to take the process to the next stage. We chose methods from service design tools (SDT)[1] together with the project leader who had experience in using them from before. These methods were then implemented in workshops with the rest of the team to understand the problem and solution from different angles and to create a shared understanding within the team.

Three different workshops were held at the beginning of the design process. Their topics were in the following order, buyer personas, customer journeys and lastly the sitemap. **Buyer personas** are examples of archetypes of real users, initially used by marketing teams to sell their products. They investigate what makes buyers want to use their service instead of competitors. The service of Contiot had many buyer personas, but they were summarized as users (Revella, 2015). **Customer journeys**, also referred to as customer journey maps, are used to plot out the customer relationship with the organization. It breaks down their interactions into several stages and explores it from the buyers perspective to identify the key interactions with the organization from start to finish (Villani, 2018). **Sitemaps** show how the different pages will be linked together during development and function as to-do lists for wireframes,

[1] https://servicedesigntools.org/

showing which screens that need to be produced. They can also help in mapping out the user journey (Caddick and Cable, 2011).

The workshop for **buyer personas** was held to determine the potential buyers of the DCC Hub. An example of a buyer persona can be seen in figure 9. I created templates and filled them out with examples that we used to correct and fill out during the workshop. This helped in having a starting point for the discussions as well as giving me time to form my own understanding of the customers, which would eventually help in facilitating the workshop. In this workshop the project leader and product owner were present as they were the ones talking with the potential buyers.

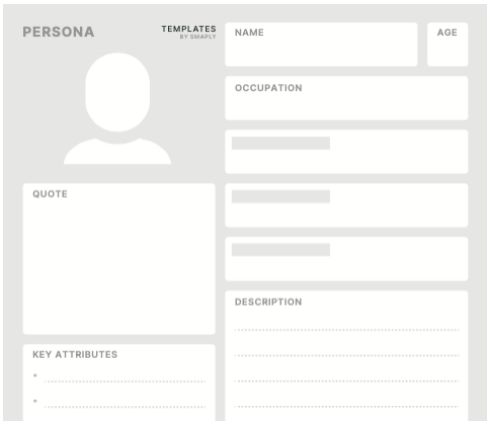

Figure 9: An empty template of a buyer persona. The different fields are usually filled out through UX research about the buyers or users (Cramer, 2020)

The workshop for **customer journeys** was split into two occasions, determining the customer journey for the company doing calibrations and the company ordering calibrations. An example of a customer journey for a retail scenario can be seen in figure 10. These were called issuer and customer. These customer journeys were essentially formed by the knowledge gathered from the buyer persona workshop, where several different buyers were summed up in two personas and some were left out for future development.

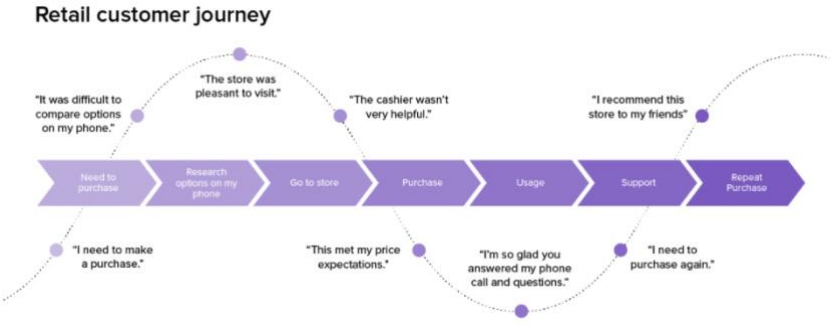

Figure 10: An example of a customer journey in retail. The journey starts from the left, the dotted lines portrays the users thoughts and the structure in the middle shows the steps they need to complete the process. (‘Customer journey map: What it is and why you need one’, 2021)

Finally, the last workshop was held about the sitemap. An example of a sitemap can be seen in figure 11. I had built a structure based on the customer journeys that I presented to the team, including the development team. In the workshops feedback was collected on improving and correcting the structure as well as locating which of the screens that would also work as APIs, which could help the development team. (Gibbons, 2016)

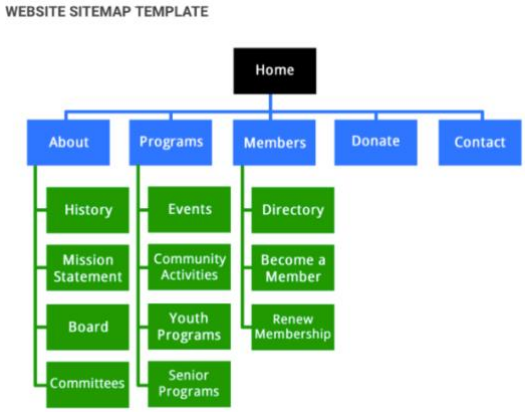

Figure 11: An example of a sitemap used in a website. The sitemap shows an abstraction of the screens in the webpage and has a hierarchical structure, where the user lands on the top page and can then go deeper into the application. (*5 Easy Steps to Creating a Sitemap For a Website*, 2019)

Ideate stage

At this stage the learnings from the previous chapter had been summed up in a sitemap. This gave all the necessary information to start creating the wireframe, with all the needed screens and the needed overall functionality within these screens. A wireframe of Youtube is presented below in figure 12, with the structure and functionality without any real content or design.

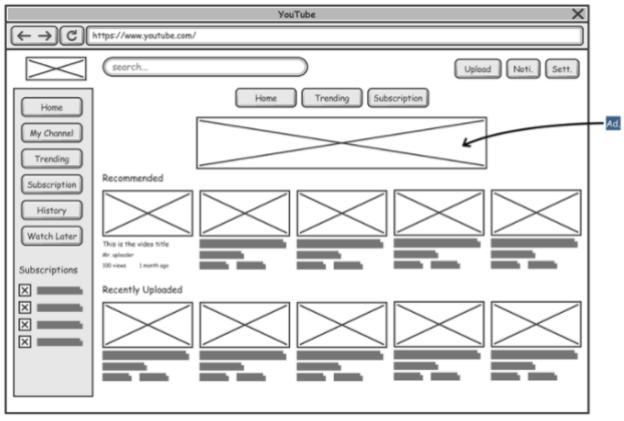

Figure 12: A wireframe depicting Youtube's website. All elements are shown, essentially without design and data, but the structure shows how the page should look. (Daniel, 2019)

With the creation of the wireframe an important choice had to be made, whether to make physical wireframes or move directly to digital wireframes. This was an easy choice at the time because the office was moving home, which meant that online capabilities were crucial for being able to showcase and gather feedback for further improvements. At the start of the ideation stage, it was decided to move the UX process to more agile ways of working. This was to get me closer to the development team that was already working in short sprints and focusing on new or different areas every week. Thus, I created a backlog, which is a set of smaller tasks that can later be selected to be done, which can be seen in figure 13 with a summary of the time schedule. The backlog was discussed with the product leader and the product owner to make it coherent with the rest of the team's schedules. These plans would then be evaluated at the end of every week to be able to plan the efforts of the following week, following the structure of the agile development process (Gothelf, 2013).

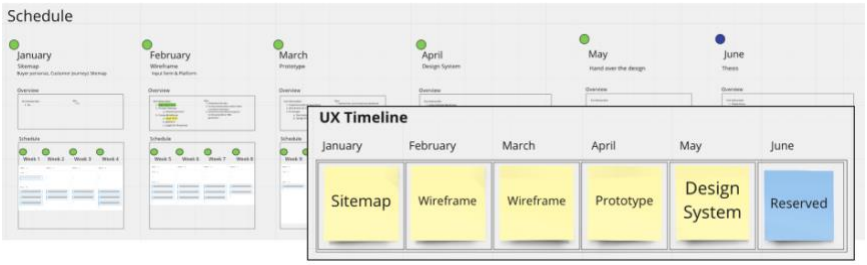

Figure 13: The bigger picture below shows the project plan and backlog as a combined structure. The picture on top shows a summary of the expected time schedule for each phase.

Prototype stage

When the wireframe was deemed good enough, it was time to start the prototype stage. At this stage the agile processes had been implemented and external showcases had become a standardized process. The team required the final design in a form that would help the development team, not just in terms of looking and playing around with the prototype. Thus, it was decided to create a **design system** in combination with the prototype. The creation of the design system will be discussed further in the handover phase as it was not the main focus at this stage.

Initially the idea was to build the prototype out of the wireframe, creating a copy of the document and reusing the whole document for the prototype. However, this proved a difficult task, and a tradeoff was introduced. Creating a wireframe should be a quick process, getting as much insight as fast as possible, but turning it into a reusable prototype is difficult if the wireframe is not done according to proper prototyping standards, which on the other hand takes time. To be able to build the prototype in a way that satisfies current standards as well as makes the design to code better, it needs to follow a certain structure. This structure

needs to be planned to work well, which includes discussions with the development team as every coder usually follows their own code structure. For these reasons it was decided to throw away the wireframe, but all the information including structure, data and functionality was reused in the prototype. An example of a prototype of a banking app can be seen in in figure 14. The app shows all the functionality of the final product as well as the design but has no real usage.

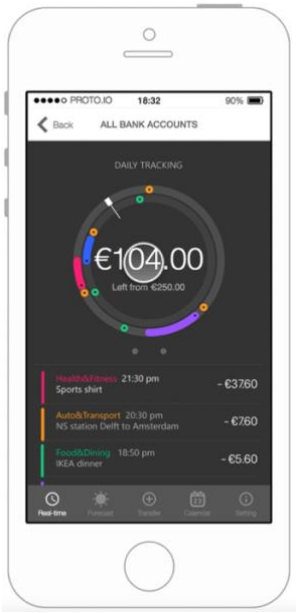

Figure 14: An example of a prototype of a banking app. The prototype shows the final design and, in this case, also the final functionality. The look and feel of the product is almost identical to that of the real product. (Ibragimova, 2016)

The end goal of the prototype stage was to have a deliverable design for development. Similarly, to the banking app in figure 14, it did not need to have any real functionality, but it needed to show all the functionality and data as well as the design.

Handover

As the outcome was discussed of the UX design process, it was decided that anything that could help the development team implement the prototype would be necessary for the project. Thus, the handover became a step that needed to be carefully planned and executed. Design systems and design to code software were introduced to bridge the gap between design and development. As they are both very new concepts it was decided to include them in the research to see if they could indeed help in transforming designs into code and helping the development team bring life to the designs.

The creation of design systems did not follow conventional methods. This was because it can take a very long time since several systems must be combined, and it all must be planned for the exactly right components to be created. Thus, I decided to create a design system solely in Figma, including the designs, components, styles as well as guidelines for how to use the design system. This of course, included the prototype which shows the interactions and functionality in action. The creation of the design system followed atomic design guidelines (*Designing Systems | Atomic Design by Brad Frost*, n.d.). This means that the components were structured and created in a specific manner that would resemble React code structure, which is shown in the figure 15 below with examples of potential structures. The logo libraries were created according to development recommendations. This included using some already existing library out there, which I was free to choose.

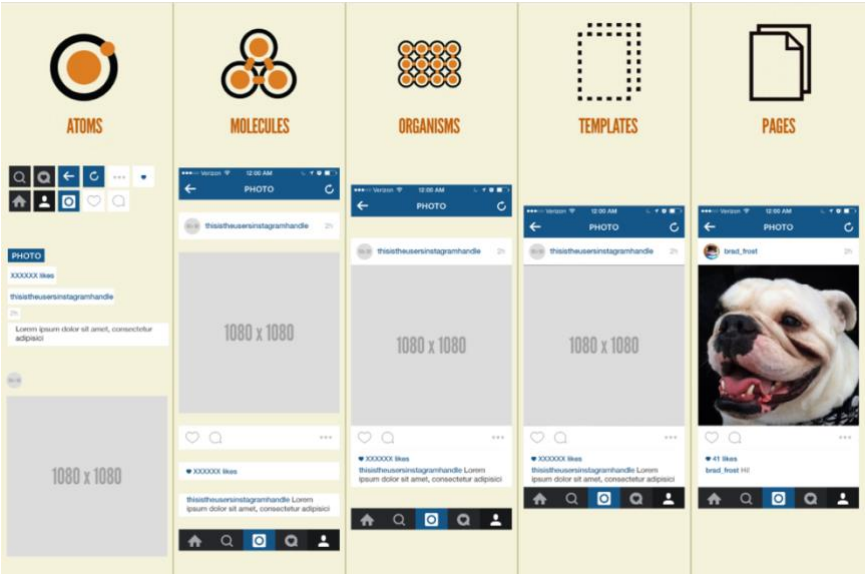

Figure 15: Atomic design of a social media smartphone app. Going from left to right elements are grouped together to create larger entities until the final structure is complete. (*Atomic Design Methodology | Atomic Design by Brad Frost*, n.d.)

For the design to code aspects, I decided to go a new route that has previously not been researched very much. This route included using design to code software for exporting the designs to code, to be easily implemented by the development team. I chose a tool based on reviews and recommendations from professionals in the industry on the social video platform YouTube. Usually when a new tool or feature is created or implemented, they are the first ones to know due to their reach and exposure to potential customers. Therefore, they are a good source for finding new potentially good tools or features. The D2C tool was first tested and discussed with the development team lead to see if it showed any promise. A plan was created to hire more employees at the completion

of the design. The new team members would focus on creating components out of the design, which is why the D2C functionality could be very important. The expectation was that it would be able to give the development team a foundation for their coding, and even usable code that they could then further improve. It was recommended to structure components with atomic design, and in a way that makes them responsive. This aids development in being able to copy code that in turn makes the design responsive in the web page. Responsivity means that as the user is adjusting the size of the page, the elements will adjust themselves accordingly. This meant that the components and elements would need to be structured and named in ways that would make sense for development.

3.2.3 Choosing the right tools to execute the methods and create the artefacts

After the methods and ways of working had been chosen for the framework, it was time to select the right tools to execute the selected methods. To choose the right tools several criteria were constructed based on the two inspirational studies for this framework and important principles from Lean UX. The tools were found through web analysis and recommendations from peers in UX as well as personal experience.

The tools for each stage of the framework followed a comparison analysis to get familiar with different facets that make tools good for the specific stage. Choosing the criteria to analyze the tools were picked from the previous chapter that would best suit the Lean UX approach. The criteria can be split into two categories, Lean UX qualities and case specific qualities. The lean UX qualities are ones that were present in each scenario of choosing the tools, such as pricing or ease of use, these are more general. The case specific qualities were qualities that helped the tool do more specific tasks in stages of the framework, for example transforming design to code. For the comparison analysis several tools for the different tasks of the framework were put into a comparison table and compared with the lean UX qualities as well as the case specific ones. This was done separately for each section of the framework, except for the ideation and prototype stage because the same tool was used for both of them.

The lean UX criteria followed principles from Lean UX. The following criteria were analyzed, pricing, ease of use, collaboration and functionality based off Lean UX criteria (Gothelf, 2013). Another criteria that was considered that was not evaluated per se, was the transformability. The transformability is about how well the tool can transform designs into code. This is important for reusability and for faster deliveries. If the development can either export components or use the code from the tool, then this can greatly improve reusability and thus the speed at which the design can be implemented. However, the software that focused on transforming designs to code were still evaluated.

The tool for the design system was selected according to the previous stage, as it was the only solution that made any sense. When creating the prototype, the components of the design system need to be in the same space to be able to use them quickly and smoothly, thus the same tool was used for both. The selected tool for the design to code software was as mentioned before, selected through recommendations by professionals in the industry.

3.3 Evaluation of the framework, artefacts and design to code software

After the process was over and the handover to the company was done, the framework and the produced artefacts were evaluated. This was done to verify if the idea behind the framework stood any ground and if the chosen approach was in the right direction.

Qualitative interviews are good when trying to find new insights (Hammarberg, Kirkman and de Lacey, 2016), but for confirmation of assumptions or evaluation, surveys can be an even better option. **Surveys** can be described as "the collection of information from a sample of individuals through their responses to questions" (Check and Schutt, 2011). Surveys are a flexible research method, they allow for a variety of methods to choose participants and collect data (Ponto, 2015). Surveys allow for both quantitative research through methods such as numerically rating aspects, and qualitative research through open ended questions or both, which is called using mixed methods (Ponto, 2015). More specifically the method I used was a **questionnaire**, which typically include various items that reflect the research aims.

Sampling a population is important for any research involving users. The goal is to find a sufficient sample that portrays the population of interest. Thus, it is important to identify the population of interest. (Ponto, 2015). For the survey, the most important aspect was that the participants would have experience with the case that I was working, and a background in computer science as well as experience with startups were important criteria as well. The number of people that had experience with the UX project at Contiot was low, considering that the operational side consisted of seven people at the end of the project. To get a larger sample I included five external participants that all had a degree in computer science or that were working with development. All the external participants had some experience with startup activity from either work, personal or school

projects. Most importantly they had some knowledge in frontend development.

The most important benefits of using surveys in this case are the collections of empirical data through real life observations and that large amounts of data can be collected in a short amount of time at low cost Kelley *et al.*, 2003). Therefore, it is easier to scope the timeframe for the research. Surveys also helps in getting rid of interviewer bias, which could have been an issue considering the action research approach that I am taking. Instead the bias is spread amongst the participants themselves. Finally, the requirement for using methods that work online made surveys a suitable approach.

Surveys do come with some disadvantages as well. There is less opportunity to ask and especially discuss answers leading to a lack of depth (Kelley *et al.*, 2003). However, I tried to combat this with asking participants to explain their answers and opinions, for example when rating a feature on a scale. Another problem with surveys can be a low answer rate (Kelley *et al.*, 2003), which is why surveys are generally sent out to as many viable participants as possible. However, in this case a low answer rate was not an issue as it was sent internally in the company with a smaller population, but people that would with a very high certainty answer the survey. On the other side the survey was sent to the participants chosen for the walkthrough of the process, which means that they all answered the survey.

The goal with the survey was to help me answer the third research question in evaluating the framework and artefacts and being able to improve the framework. The survey starts with asking the participants for their consent and presenting the structure for them to aid them in following the survey. Firstly, the survey captures the teams experience with UX, this gives an idea to why the process might have performed the way it did. A team with much experience in UX might be able to help more in the process and they might also show a more positive attitude. On the other hand, a team with no experience might show less interest, which could affect the process. The survey goes on to evaluate the four different stages of the framework separately, with the deliverables, methods and tools in the focus. The survey also evaluates the handover separately from the development team's perspective, to give insights in how the handover can be improved and whether design systems and design to code software show any promise. Lastly the participants are asked about their general opinion of the framework with the ability to add comments freely.

For the external participants, that did not have experience with the project, walkthroughs of the whole project were held. First an introduction to the whole project was presented, then each of the stages where participants could use the different tools and look around in the files after which they were prompted to answer the questions for that stage. Before answering the questions, the participants were asked to think of themselves as an external developer joining the project. The walkthroughs

lasted for an hour to an hour and a half, including answering the questions directly in the questionnaire. The walkthroughs were especially useful, as they participants could ask questions during answering the questions and motivate their answers more.

The questions of the survey are especially directed towards the third research questions on how the framework performs and how it could be improved. The different deliverables are evaluated in terms of their usability in the perspective of the different team members. This is to determine how much the created artefacts helped the team in gaining insights and moving forward. The tools, that were used in the process, were evaluated to see how they were perceived and to see if they can be recommended for future usage in the framework. The design to code aspects were also evaluated specifically by the development team to see how they perform and how they could be used. Finally, questions about the overall performance were asked evaluating the perceived outcome and the pleasantness of the design process.

In the final handover stage, the design system is evaluated among both of the design to code options, Figma and Anima App. The ease of use and the usefulness of the design system was examined along with the usefulness and potential of the design to code solutions. The handover stage was directed towards the developers, but non developers could still answer the questions if they had found the stage helpful or otherwise. After the handover the framework and process were evaluated to find out overall opinions and at this point participants could give feedback freely, which could help if some important question had been left out.

4 Results

In this chapter the construction of the framework will be presented, based on the literature review and decision-making process introduced in the previous chapter. The implementation will be presented as well, along with the results of the evaluation survey of the framework.

4.1 Construction of the framework

The construction of the framework followed principles from Lean UX and took inspiration from other works that are similar in scope and objectives. Continuous learning was attempted all along the process by asking questions and making assumptions that could be validated along the way. (*A Simple Introduction to Lean UX*, n.d.).

The design process was mostly held separate from the other processes at the company. This was due to missing infrastructure in the company, which would have made the design process much harder to do if they had been combined. Instead, the design process was done in parallel with the development process, where knowledge from either process could help in the other. Both processes followed agile methodology working in short sprints, focusing on the customers, and allocating resources on the areas that needed them the most.

The final structure of the framework was four stages that would focus on the development side of UX, this excluded doing user research and testing with users. Instead, the framework follows the three middle phases of the design thinking process define, ideate and prototype which are designed to use the previous research and form solutions and designs of the final product. The structure included a final step called handover which was designed to make a smooth transition from design to development.

4.1.1 Define stage

The first phase of the framework is the define stage. In this stage all knowledge and resources are pooled together to create an understanding and formulate problem statements (Gibbons, 2016). Thus, it is in the define stage where the UX development starts in this framework, when the customer research phase is done, and the information is ready to be conceptualized.

Selecting the tool for the Define stage

Selecting the right tool for the define phase was done through comparison tables including several criteria to rate the different solutions on. The chosen contenders were **Miro, FigJam** and **Microsoft Whiteboard**. Miro and Microsoft Whiteboard are already known tools in the industry, while

FigJam is still in its infancy. FigJam is gaining traction fast due to its connection to the popular design tool Figma, as the tools share the same creators. Contiot uses Microsoft Teams, because of this I decided to add their solution of this type of software to the comparison. I was the sole contributor to the evaluation of tools.

The general features that were compared were pricing, ease of use, functionality, integrations and collaboration. All these features are also applicable to other types of organizations when comparing tools, but in terms of Lean UX they are especially important. For pricing I considered options that had a free version, because startups might not have the same resources to spend on tools as larger organizations. The ease of use is important because having a tool that is easy to use and learn can greatly speed up the process as the team can spend less time on learning to use the tool. Functionality is perhaps a more overall feature, but no less important. Integrations are important because they can improve the quality of designs, help development and speed up the process as external solutions, like for example icon libraries through plugins can offer better quality icons that are easy to use that might also offer developers the same library. Finally, collaboration is crucial for Lean UX as better collaboration can create a shared understanding within the team and online collaboration capabilities is required at the time of this thesis.

The scale of rating the tools goes from **one (poor)** to **three (great)**. A one on the scale means that the feature is below average, a two means that it is good in that aspect and can be above average and a three means that the tools is great in that aspect and at the very top compared to other tools. In table 1 the comparison can be seen with FigJam receiving the highest rating.

Overall Criterion	Miro	FigJam	Microsoft Whiteboard
Pricing	3	3	1
Ease of use	3	3	2
Functionality	2	3	2
Integrations	3	2	2
Collaboration	2	3	3
Total	**13**	**14**	**10**

Table 1: A comparison table between the different tools in the define phase.

In terms of **pricing**, FigJam and Miro scored the highest with providing a free solution, it is however unclear if FigJam will continue to be free after the full release of the tool. But considering their free option with Figma it is highly likely that some version of the tool will be free. Microsoft Whiteboard scored very low as they do not reveal their pricing, which means that it might be higher or it might be lower, but unnecessary

measures must be taken to find out when there are already free solutions out there with high performance.

With the **ease of use** both Miro and FigJam scored the highest. They are both very easy to take into use both the browser and desktop application. The Microsoft option on the other hand requires more effort to take into use. While using the tools they are all similar. Therefore, Miro and FigJam scored higher than Microsoft Whiteboard.

All three options share similar **functionality**. However, the functionality in FigJam scored higher than the rest due to its capabilities of showcasing designs directly from Figma. The other tools are equally good at the other features, but due to this functionality especially considering lean UX principles, this feature can potentially give a lot of value. If certain design elements can be created in Figma to be discussed separately with the help of FigJam this could save time and add creativity to the process.

Miro has the best options when it comes to **integrations**. Miro offers many different options, including Adobe XD which is another prototyping tool. The other tools might offer better integrations, but for FigJam it offers great integrations for Figma, but otherwise not many. On the other hand, Microsoft Whiteboard offers many good integrations, but it is optimized for strictly other Microsoft products. This means that either option could be an even better alternative, but only if the team favors either Microsoft products or if Figma is the main tool for UX development.

Perhaps one of the most important aspects is **collaboration**. Both FigJam and Microsoft Whiteboard got the highest score in terms of collaboration. Microsoft Whiteboard offers a live video call option for the whiteboard, which is simply a good aid for the collaboration. However, FigJam offers other functionality to compensate, with a cursor chat. The cursor chat allows users to write comments which shows up where their cursor is placed, which can be seen in real time by other users in the same space. The Miro tool also offers great collaboration but did not score as high due to it lacking unique features specifically designed for better collaboration. The current standard of live video calls might also undermine the importance of Microsoft's solution, but in terms of collaboration it is a good addition and might come with unexpected benefits.

Interestingly, FigJam scored the highest of the three collaboration tools. However, at the time of choosing a tool, FigJam was still in its beta release stage. This would make the outcome of using FigJam uncertain. The team also had previous experience with Miro, which is why Miro was chosen even though it received a lower rating than FigJam. Nonetheless, I decided to include FigJam in the comparison, this was partly due to FigJam gaining much traction on YouTube, where several professionals in the industry have been showcasing their experience with the beta release. For this project Miro was the best tool to use and satisfied all the

criteria. However, in the future FigJam will be a potential contender, especially if a company is considering an all-in-one solution.

Implementation of the Define stage

The implementation of the define stage started with the goal of a Sitemap in mind. But, as mentioned in the previous chapter several steps had to be taken to create the sitemap. The selected methods to form conclusions about the data were buyer personas, customer journeys, and then finally the Sitemap could be created.

The buyer personas were used to identify the most important customers of which there were sixteen. These were in turn summarized into five different categories of which two were chosen to be the most important ones needed for the platform to make sense, which can be seen in figure 16. These two categories were **customers and issuers**. Customers would be the ones that needed their instruments to be calibrated and issuer would be the ones to do the calibrations and issue calibration certificates to the customers. This would be similar to the buyers and sellers in the process by Martins *et al.* (2020).

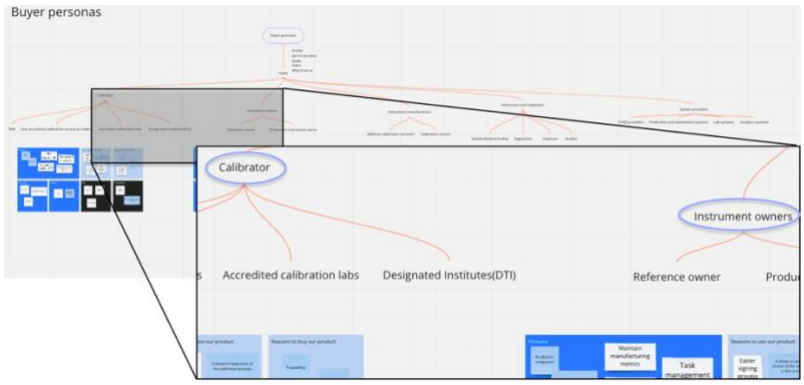

Figure 16: Buyer personas, sixteen types of customers were identified and grouped into five different categories, two categories were selected as the key players for the platform to work.

For both categories customer journeys were constructed in two different workshops, two for the customer and one for the issuer. For this workshop the development team lead was included to give insights into the feasibility of certain steps. **Three customer journeys** were constructed to represent the process of **ordering, delivering and receiving** a calibration certificate. All three journeys started from logging in to the service completing the task through all the necessary steps, to the signing out of the service. The customer journey of the delivery can be seen in figure 17 on the next page as “create and send DCC”, which are the actions needed to deliver it.

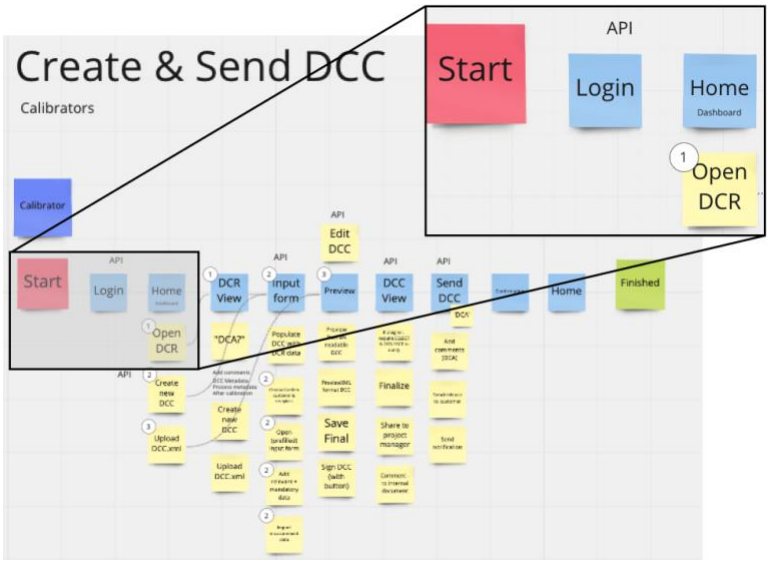

Figure 17: One of the three customer journeys, where the user logs in to the service to perform their key activity.

After the buyer personas had been constructed to identify the most important customers and their respective journeys with the platform had been created, the knowledge that was needed to create the sitemap was found. I created an example structure for the sitemap and a workshop was held to improve and correct the structure. The initial structure of the sitemap included all the screens and functionality from the customer journeys abstracted into the structure that can be seen in figure 18. The process of validating the sitemap and finalizing it happened in an iterative nature. The structure that was achieved in the final workshop in the define stage was not the final structure of the sitemap. The sitemap had to undergo several restructurings during the ideation stage as new knowledge was found when visualizing the sitemap in a wireframe.

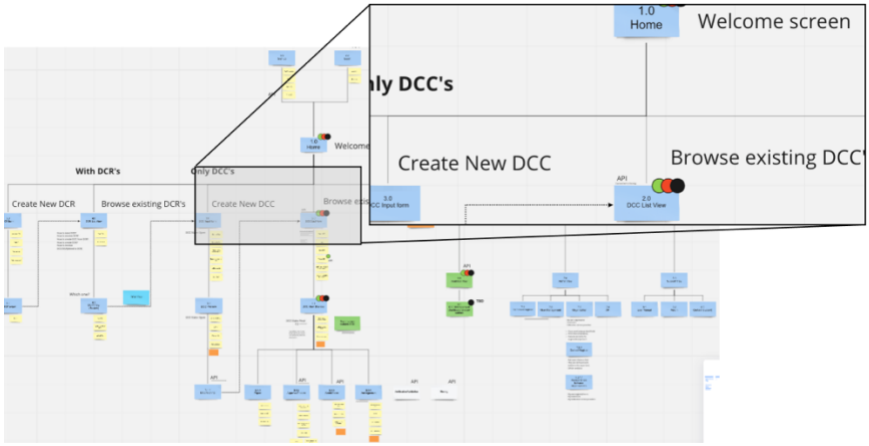

Figure 18: The sitemap that was created, Screens are marked with blue, functionality yellow, popups orange and content with green. The screens are numbered, which helps in navigating and understanding the logic in the sitemap

The define stage helped me as a UX designer to get into the project and understand the industry, which in R2B and B2B startups is especially crucial. In R2B and B2B startups the threshold of failing is much lower as there are often more data to handle, and the stakes are much higher due to requirements from industry partners and other stakeholders. After the define phase was done the users had been identified, their use of the platform had been mapped out and an abstraction of the whole platform was created. This helped in generating many useful discussions and new questions were asked that otherwise would not have been found.

4.1.2 Ideate stage

At the beginning of the ideation phase the knowledge from the user research had been formed into an understanding or problem statement. In the ideation stage creation of new solutions happens based on the previous stage with complete freedom (Gibbons, 2016; Dam and Siang, 2021).

Selecting tools for the Ideation and Prototype stage

The tools chosen for comparison were perhaps the most popular tools in industry **Figma**, **Adobe XD** and **Sketch**. Adobe XD is created by Adobe, which is already a giant in the industry. Figma was created by people working specifically in UX design and has a strong user community. Sketch has been around for longer and has an even larger community, but Figma seems to be getting much traction due to their all-in-one solution. All the tools are especially good at prototyping, but they also make very good options for doing digital wireframes.

The criteria for selecting the right tool are the same as the overall criteria from the previous section. All the features in the comparison take Lean UX principles into account when comparing the different features, seen in table 2. This comparison is done for both the ideation and prototype stage. The same scale from the previous comparison is used here as well, going from one to three.

Overall criterion	Adobe XD	Figma	Sketch
Pricing	1	3	2
Ease of use	3	3	2
Functionality	3	3	2
Integrations	3	3	3
Collaboration	2	3	2
Total	**12**	**15**	**11**

Table 2: A comparison between the different tools for wireframing and prototyping

Figma is the only one that offers a free solution as far as **pricing** goes. Then comes Sketch with a hundred dollars a year, and finally Adobe XD with twelve dollars a month. Considering Lean startup principles, Figma gets a full score with unlimited editors within three design files for the free version, which is more than enough for many small projects. For larger teams Sketch and Figma might even out considering the paid options, but not considering trying to keep the process and costs to a minimum.

Considering the **ease of use** they are all state of the art and very similar to each other. However, Sketch received a lower rating due to its reliance on many plugins, even though this is made easy, it can still be time consuming searching for plugins or knowing what to look for.

For **integrations** each tool has many good integrations. All can be installed and easily used within the application. Figma however, offers a plugin called Anima for Figma, that enables better exportation of code. The other tools do offer similar solutions, which is why they were rated equally.

For **collaboration** Figma came out on top with the real time editing in the files. This allows for better codesign of all files within Figma as well as showcasing as all participants can edit the files or show what they mean with their cursors.

The three of these tools compare well with each other and they are all state-of-the-art industry leaders at what they do. This means that choosing either of these tools will likely grant good or similar results. However, in my case I chose Figma as a tool because I had previous experience with both Figma and Adobe XD. However, Figma offered better functionality with external links in the prototype, a feature which Adobe XD strangely enough does not offer. This at the time was thought to be an important feature that could help in combining the wireframe and prototype with the work of the development team. According to Krout, Carrascal and Lowdermilk (2020), it is crucial to choose familiar tools to achieve faster and better results, this further enhanced the decision of using Figma.

Implementation of the Ideate stage

The pre-requirement in the ideation stage was a wireframe. However, no specific requirements or criteria were given for creating it, which left room for some creative freedom in choosing how to do it. The most important choice was between creating a digital or a physical wireframe. The digital option was chosen for better online collaboration possibilities as well as providing higher reusability.

The wireframe was constructed using the selected tool Figma. The information from the previous stage was implemented into the wireframe. Constructing the wireframe first consisted of creating all the screens that were defined in the sitemap along with their defined functionality. Due to the choice of using a digital wireframe, this happened at a very fast pace.

As the wireframe grew and insights were gathered the sitemap from the previous stage was updated. The home screen of the wireframe can be seen in Figure 19.

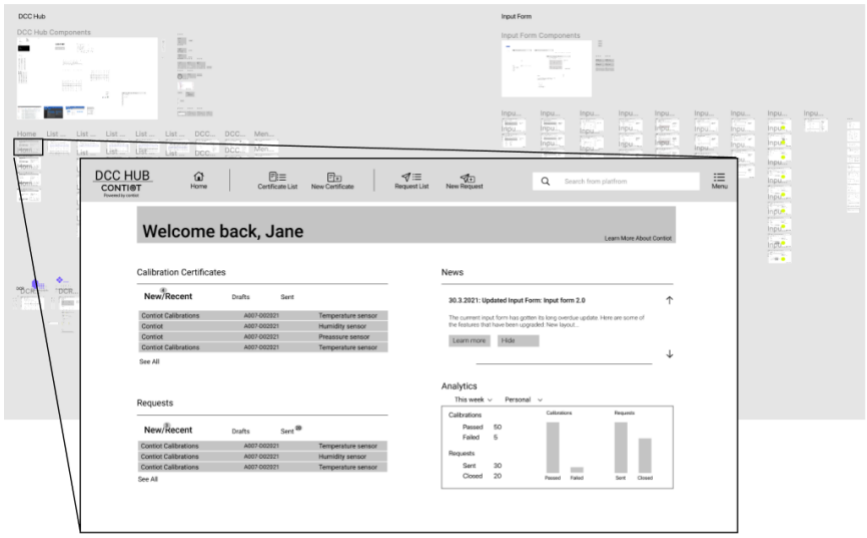

Figure 19: A zoom-in of the home screen of the wireframe. At the top is the navigation bar with links around the wireframe. Below four different components depict a list of calibrations (top left), a list of calibrations requests (bottom left), news (top right) and statistics from the usage of the platform (bottom right).

The validation of the wireframe happened two times a week internally for the start of the process, once at the start of the week at once at the end in a fashion similar to that of agile standup meetings. As the fidelity got higher it was decided to include stakeholders into the process. This meant finally being able to validate decisions with real users, which also helped in getting many new insights into the project. The time it took to get to validation with actual users was partly due to expectation management. This essentially meant that the product could not be shown if it was far from finished because this would lower the expectations of stakeholders too much. On the other hand, the product should not show anything that looks too good if the rest of the development is too far behind because it creates a false sense of the readiness in the project. This can be crucial in the success of a startup because otherwise it might lead stakeholders to expect too much or too little of the product. Nonetheless, a showcase with stakeholders at the middle of the week was introduced. The stakeholders consisted of partners from the industry with many different backgrounds and positions in the calibration industry. In the weekly showcase different functionalities and data would be introduced through the wireframe to then be validated by stakeholders, greenlighting, or correcting the wireframe.

The final evaluation of the wireframe happened with the product leader and the product owner after many weeks of creating the wireframe and getting the data right. At this stage several factors had to be in check, all

the data had to be correct, all functionalities had to be present, and the structure needed to be clear and make sense. At this stage it would not make sense to continue developing the wireframe unless it would be transformed into the prototype. Thus, it was concluded that no more time would be put in the wireframe and the next stage would start.

The implementation of the wireframe was not entirely problem free. There was much internal feedback that was continuously being implemented and the data needed for creating calibration certificates was a problem for keeping the process fast. New steps in the earlier define phase had to be created repetitively to create understandings of the data. These would then support the creation of the wireframe. Updating the sitemap proved helpful in creating the wireframe. The assumptions in the define stage could be validated in the ideation stage to provide a more truthful abstraction of the platform. The updated sitemap helped in providing a good overview of how the screens in the wireframe should be connected.

Another factor that proved problematic in the ideation stage was updating the screens in the wireframe. The whole idea of wireframes is to be as fast as possible, which proved very difficult due to the criteria of getting all the data correct. The problem came from many similar screens needing to be updated, which in the prototype stage was not an issue as they could be changed from a single location. However, in the wireframe screens were not built to be easily updated and modified, they were built in at a fast pace, gathering knowledge and getting things in the right place. Essentially the takeaway becomes that the data should be placed in the prototype stage, unless the wireframe is built in a way that makes it easily modifiable similarly to the prototype stage.

4.1.3 Prototype stage

In the prototype stage a tactile representation of the final product is built (Gibbons, 2016). At this stage the wireframe, which can also be seen as a low fidelity prototype was ready to be developed to look and feel like the final product. This meant taking the structure from the wireframe created in the ideation stage and building the final design based on the learnings from both previous stages.

Selecting the tool for the Prototype stage

The selection of tools followed the same comparison as the ideation phase because I decided to use the same tool for both wireframing and prototyping. This was because Figma as a tool offers great capabilities for digital wireframing, but it is essentially a prototyping tool. The choice was made for a smooth transition from the wireframe to the prototype, resulting in high reusability, and usability of designs and knowledge. Some components and many smaller elements could be completely reused by copying and pasting them from the wireframe to the prototype.

Implementation of the Prototype stage

For the prototyping, the requirements were to make the final design, containing all the right information and something that could be shown to all potential stakeholders. This meant that the fidelity of the prototype was required to be high. Another requirement was that two different prototypes had to be made for the two main types of customers to be able to sell the product to both of them.

The construction of the prototype was done with the same tool from the previous stage, Figma. The results can be seen in figure 20, where the home screen from the wireframe has been turned into the home screen in the prototype. Creating the prototype was a moderately fast process as all the information and design decisions were mostly done. The plan had originally been to create the wireframe, copy the whole thing and add styles to it to make it a prototype. But this option ended up failing due to missing principles when designing the wireframe. It ended up being a tradeoff between usability and speed. Designing the wireframe without thinking of the future structure made the process very fast, but it ended up costing some of the reusability of the wireframe for transforming it into a prototype. Only some elements and structures could be reused, but the wireframe worked as an excellent guide for the whole process of creating the prototype.

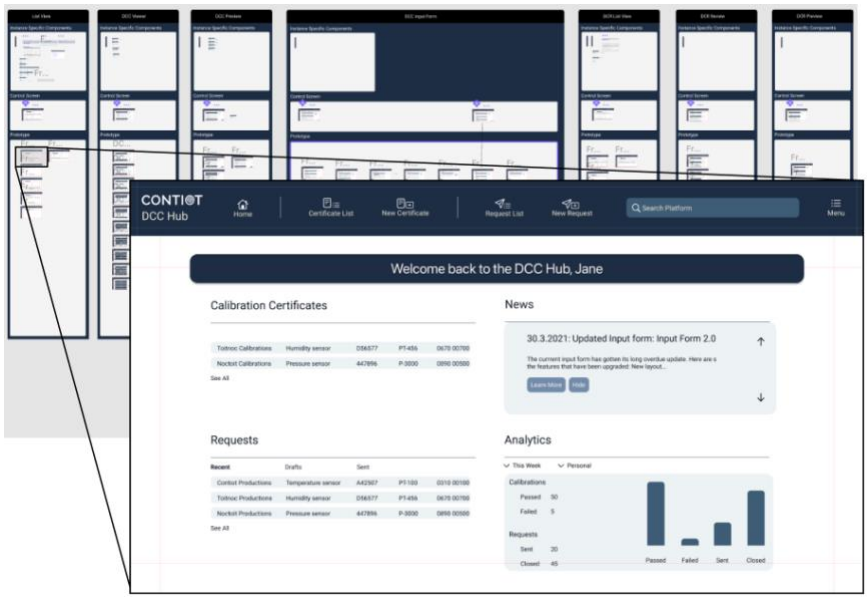

Figure 20: A zoom-in of the home screen in the prototype. All the same elements of the wireframe have been styled and minor functionality as links from the individual elements such as calibration certificates have been added for a better feel of how the full platform works.

At the prototype stage the development team was being included more in the discussions. Here it would be decided on factors like how to create the final design such as color schemes, components and styles. These elements would be implemented in the prototype during development. But if elements were being duplicated and reused, they were added from here to the design system. Eventually the development of the prototype happened through reusing elements and components according to atomic design, where smaller elements are being reused to make larger components to eventually form reusable templates and pages.

The prototype was of high fidelity, which meant that it had a high resemblance to the final product. The evaluation with stakeholders from the previous stage continued in the prototype stage. After the prototype was completed, the designs were ready for implementation, but they could have used even further refinements. The process essentially followed the build, measure and learn iteration loop from Lean Startup where each week new features and styles were added, after which they were evaluated and improved. This allowed for a fast development of the prototype where validation would come quickly, and the prototype could go on being developed without any hinders in time.

The transition from wireframe to prototype with Figma proved to be very useful, even though there were many problems. Essentially both tools can be found in the same place and use the same resources without any trouble of transferring files or styles. Because the wireframe was so high in fidelity, making the design decisions was a fast process. However, considering the problem with the wireframe in the last stage, it would be advisable to create a simpler wireframe without data, then in the prototype stage create a few of the screens and then based on the design start creating a design system and placing all the data. This is doing it according to the design reuse principles with design by reuse, which is designing using old designs or data, and design for reuse, which is designing elements to be reusable. Doing it this way would have sped up the process and leaving more time to get the final design right.

Finally, to get the most out of the ideate and prototype stage, it is advisable to start early with discussing with the development team. This means that the way the development team works can affect your design structure to be more similar to their coding structure, which affects how well they can extract and use code from the prototype and design system. Doing this will also help developers become more accustomed to the design system and all the design files from the very beginning. It can be hard to jump into the design system at the end because it is very vast and includes many different components.

4.1.4 Handover stage

The final stage of the framework is based on the framework proposed by Navarro *et al.* (2016). This stage is called Handover. This means that when the design is finally ready, be it a wireframe, a prototype or a whole design system, the design can be handed over in a smooth and well-prepared manner. This is especially important in startups where development teams might not be very large, or they might be very specialized in other coding aspects than the front end. Having a good design handover can help in building the final application, reduce waste and save time.

Selecting the tools for the Handover stage

There are many different definitions and versions of design systems. During this stage there was not much time left for the handover. With this in mind, I decided to create a simple design system within Figma, instead of using several other tools for the handover. I decided to choose only one tool that converts design into code called Anima App. The choice for creating the design system in Figma was because all the other design elements had been created there. The choice of the design to code tool came from professionals in the industry who promoted Anima App, which at the time seemed to fit the needs of the project.

Anima app is a design to code plugin for Figma. It allows designers to select the frames they want and export them to Anima. In Anima a project needs to be created, after which all the files can be conveniently exported there. Developers can then join Anima App with an invitation to the same project, there they can play around with the exported elements and inspect their code in various ways and formats. However, for this project the most important feature was the ability to get the components transformed into React code.

Implementation of the Handover stage

The method for the design handover was chosen according to industry standards, which meant creating a design system to ease the process. The structuring of the design system was done according to atomic design, where all elements were saved conveniently in a pattern library, seen in Figure 21 on the next page (*Atomic Design Methodology | Atomic Design by Brad Frost*, n.d.). This meant starting off with creating styles, which consist of fonts and colors. These can then be easily found in Figma in the team's own style library. After the styles had been created, the construction of elements began.

To create the atoms or elements as they are called in Figma, external plugins and libraries were imported to the design space to use already created logos and icons to speed up the process and improve the design. At this point the design system consisted of color and font styles, icon and logo libraries as well as font components which were used to add colors

to the fonts which is currently a missing feature in Figma, these can be seen in figure 21. In addition to this, simple buttons were created as the final addition to the atoms of the design system.

Figure 21: On the left the navigation of Contiots' design systems is shown, with the color palette as an example on the right side.

The atoms were used to make larger molecules, which are called components in Figma. The molecules are items such as the top bar, sidebars and other smaller components that consist of a few smaller elements. An example of the top bar and sidebars can be seen in figure 22.

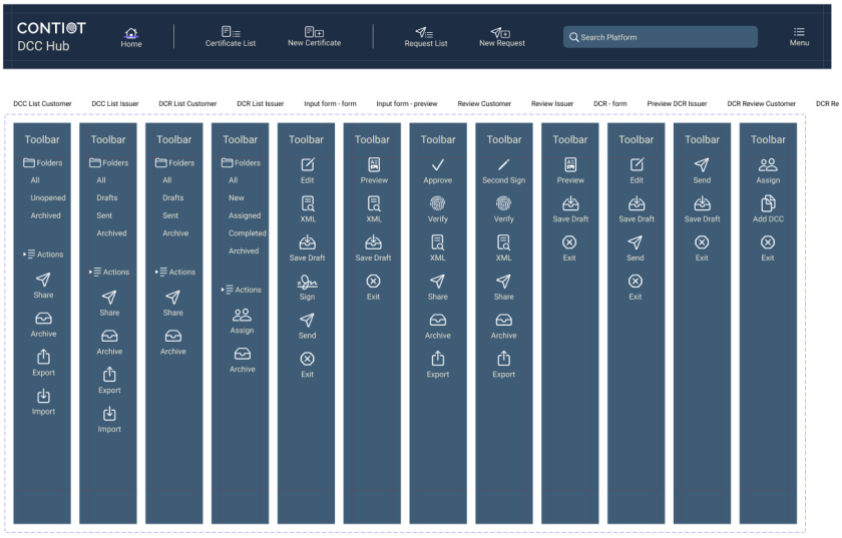

Figure 22: Molecules that have been made with smaller atoms. Here are all the different sidebars that the platform uses as well as a single top bar that is used throughout the platform.

The molecules were then used to create collections of components called organisms. In figure 23 the molecules of the home screen can be seen, which have been grouped together to form an organism. This organism can then be grouped together with the top and sidebars from the previous figure to form a template for the home screen. These templates are called control screens in the design system because they can be used to control all the screens that use the same template. This makes changing the designs very fast.

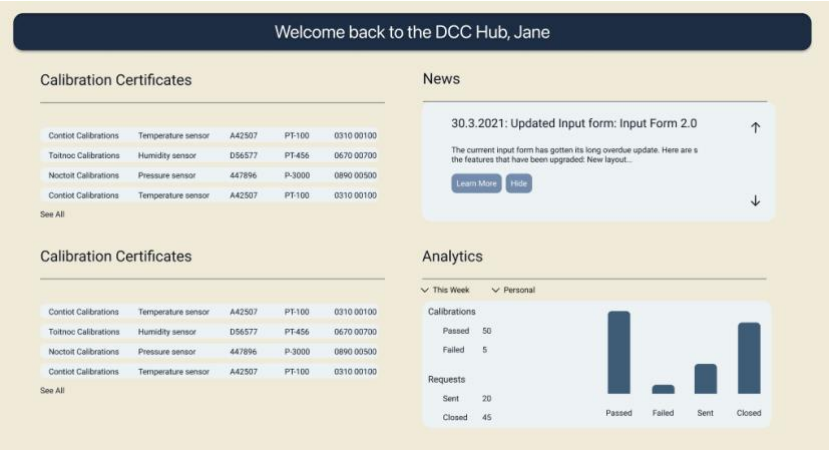

Figure 23: Four different molecules make up a larger organism. Here are four smaller components plus the welcome header that together make up a larger entity.

From the implementation it became evident that the design system helped in creating the prototype. Changing elements after a design was implemented became much faster because they could be easily found in the design system and since all instances that used the same design would be changed as well.

Something else that became apparent when exporting the design to Anima App, is that for the code to be any good, a lot of thought and planning needs to go into creating the components in the design system. For a component to make any sense as code in Anima, the component must be structured the same way it would be coded. Therefore, it would not be advisable for beginners in Figma or frontend development to use Anima. Only more experienced UX designers that have access to developers in their team should take on the task of exporting design to code.

4.2 Framework & artefact evaluation

This chapter introduces the results of the survey that was sent out after the design process at Contiot was over. The population of the survey was the

operational team within Contiot, which at the time of the survey was seven team members. The other part of the population consisted of five external participants with a background in computer science, startups and frontend development. The survey investigated their experience with the UX process, their view of the different deliverables in each phase and the different artefacts created, as well as their view of the framework. The questions of the survey can be found in the Appendix, but they will also be presented separately in this section for a clearer presentation of the results.

4.2.1 The teams experience with UX

The first section of the survey was created to determine the teams experience of UX. The team did have some experience of UX which can be seen in chart 1. Only two participants had more experience and knew how to use related tools. Five participants had some experience from smaller projects, and the final five only knew what it was. The team had experience with Figma Miro, Balsamic, Adobe XD and Photoshop, with methods such as user stories, site mapping and the double diamond model. Finally, some mentioned a few frontend languages, tools and concepts.

When asked about the value of UX, the participants believed that it brings much value. It was mentioned that it is the easiest way to deliver an MVP, which is very useful for startups. It was said that it brings user satisfaction, retention and engagement through the inclusion of customers into the process. It was also mentioned to be very important in every business, and that it is a great differentiator from competitors. However, many mentioned that it can often be forgotten in startups, but considering all the value it can bring, it should get more attention.

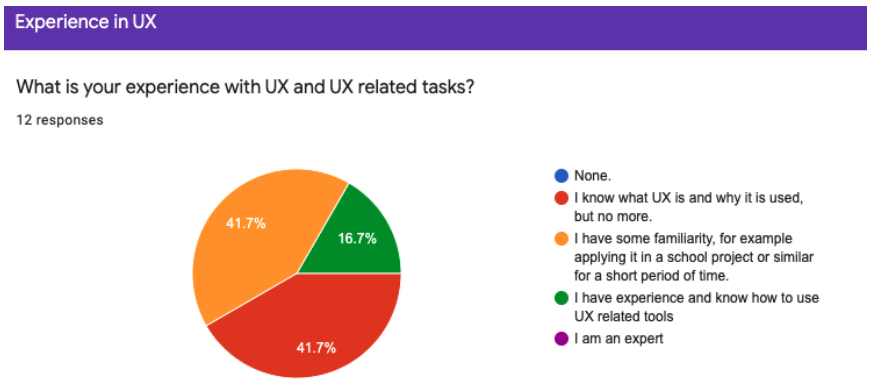

Chart 1: The participants experience with UX. Only two participants had extensive experience with knowledge of using related tools, but they all had at least some experience.

4.2.2 The scales of measurement

All the sections in the survey followed the same scales of measurement. The deliverables were rated in terms of their usefulness to the team. The scale of rating the deliverables went from one to five accordingly:

1. **Not useful**, it did not give me anything, no new insights or knowledge and i could not use it for anything.
2. **Somewhat useful**, It gave me some insights, I could potentially find some use for it.
3. **Useful**, It gave me insights or knowledge, I could use it for some aspects of my work.
4. **Very useful**, It gave great insight or knowledge, I could definitely use it for my work.
5. **Exceptional**, It is crucial for the success of the operation, it is of great value and is able to find new insights and knowledge that would otherwise not have been found. I use or will use for my work.

The rating of the different tools also followed a scale from one to five measuring the performance in different aspects. They went as following:

1. **Poor**, it fails in several aspects, I would not use it again.
2. **Fair**, it gets the job done, but may be lacking in some aspects, I might not use it again.
3. **Ok**, It gets the job done in most aspects, I would use it again.
4. **Good**, It gets the job done in all aspects, I would use it again, it compares better than most similar tools.
5. **Excellent**, It gets the job done perfectly, in this aspect it might be your tool of choice. I consider it at the top with maybe one or two other tools in this aspect.

4.2.3 Define stage results

The rating of the **usefulness** of the sitemap in the define phase received a **mean value of 3.6**, which is a good rating considering the scale. The scores can be seen in chart 2 on the following page. This meant that everyone found some insight from it and most either found it useful for at least some part of their work, but some even found great insight and could definitely use it for their work.

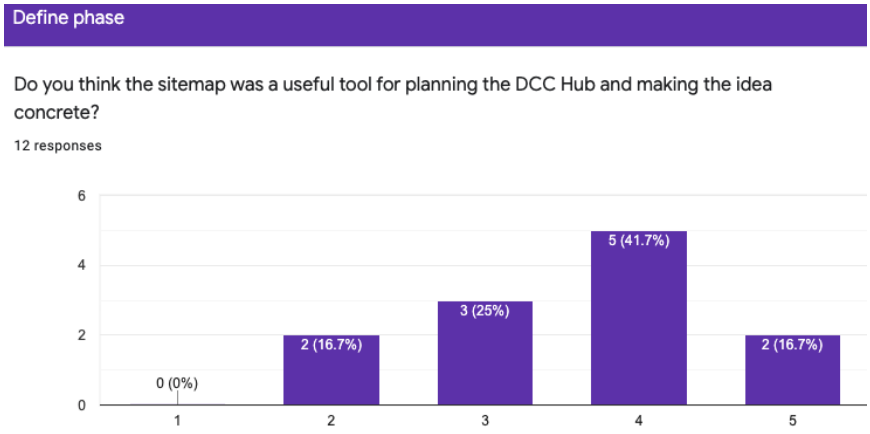

Chart 2: Usability of the sitemap within the team.

When asked about why it was useful the participants responded with ensuring that all the needed screens had been captured, the structure and functionality was clearly linked, it gave a good holistic view of the platform, and it helps all the people involved to understand how the platform should be built. Some reported that they already knew the concepts, but it was a good bonus, and someone even mentioned that it was too complicated. However, this was due to the development team being involved in creating the first iteration of the sitemap, but they never saw the later iterations. The first iteration was indeed very complicated and in need of restructuring, and only after a few iterations and learnings from the ideation stage the final version of the sitemap was created. Nonetheless, the evaluation yielded positive results and a definite usability to the project. This makes it is a necessary addition to the framework, also in terms of the UX designers own needs, that were presented in the previous chapter in the define stage. The needs were essentially to get into the project and learn about the industry.

Miro as the tool for the Define stage

The tool Miro was rated according to three features, functionality, ease of use and collaboration. The **functionality and ease of use** both received a **mean rating of 3.9**. The **collaboration** received a **mean rating of 4.1**, but with two less answers. This lands Miro on an overall rating of four, meaning that it is a good tool that gets the job done and compares well to most other similar tools. All the participants responded that they would use it again.

When asked to motivate their answers about Miro the participants responded with it being very good for collaboration and remote work. It was mentioned that it worked very well for the specific project, especially for internal communication, but not as much external showcasing or demoing. It was said that it was easy to use, but that it can get quite cluttered, because all documentation is in the same place. However, on the

contrary it was also mentioned that it was easy to find things without having to go through multiple folders and files. Essentially it is up to the designer to keep the file tidy in order for everyone in the team to find their way through the file.

4.2.4 Ideation stage results

The **usefulness** of the ideation phase received a **mean rating of 4.3**, which makes it the most useful of all the stages to the rest of the team. This meant that they found it very useful and could certainly use it for their work. The rating is seen in chart 3.

It was reported that it was intuitive and easy to use, even for the participants with very little previous knowledge. From a developer perspective it was a good start to try different things and it could be used as a base for implementing the real product. The wireframe gave good requirements, defined all the needed data and facilitated many discussions so that team members did not have to imagine the same thing in meetings, which can often go wrong. Overall, the functionality was very good, and many insights were gathered from the wireframe, and it was even mentioned that it could satisfy many MVP needs. One person reported that it is not advisable to make the designs from scratch, leading me to believe that they would have wanted some external libraries and components to be used in the wireframe stage. However, this would be more convenient in the following stage as the wireframe can conveniently show what types of external libraries could be used in the first place.

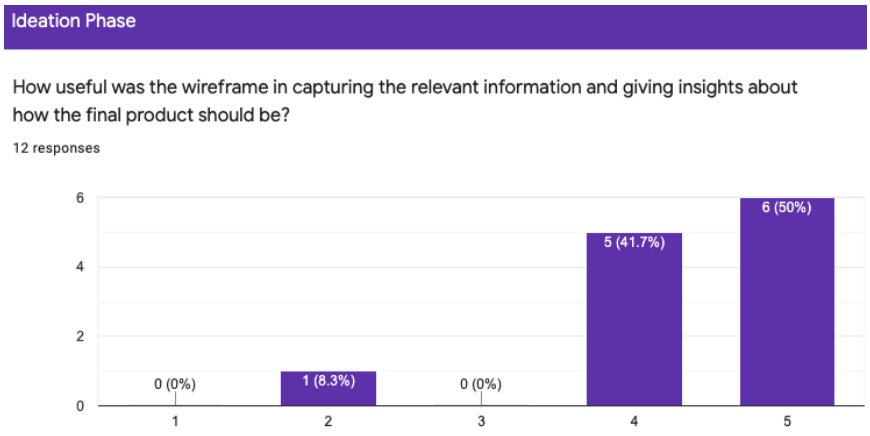

Chart 3: The perceived usefulness of the Ideation phase within the team.

Figma as a tool for the Ideation stage

Figma received a **mean rating of 4** when asked about how well it worked for showcasing the product and ideas. This means that it got the job done and that the team would use it in the future.

The participants were asked about what went well and what did not go well with showcasing with Figma, and they mostly responded positively. Figma was said to be the industry standard, and one said they would not want to see any other tool used. The outcome was said to be impressive from a business perspective, it was clear and gave a great overview on the project. Figma was very good for showcasing with no lag in preview mode, clickable objects were easy to find, and it even worked on the phone. However, it was mentioned that it could be slightly complicated without previous use. One of the participants mentioned that it might be better for larger organizations, as products can easily be shipped to market to then receive feedback if done on a budget. However, this can be the case in B2C startups, but in B2B the data needs to be correct before delivering the product in most cases.

Figma enables the creation of almost anything, which helps in imagining, discussing and planning the product. The data structures and layouts were easily defined and implemented, but the relations between the frames (screens) was a little laborious according to a participant who had been working with me in defining many of the screens. Finally, from a development perspective, Figma is a good starting point, but if something looks good in Figma, it might be hard to develop. This often leads to redesigning elements in Figma that did not work due to obstacles in development.

4.2.5 Prototype stage results

For the usability of the prototype, I chose to evaluate it from two perspectives, the usability for showcasing it to stakeholders and the usability for the development team, which can be seen in chart 4 on the following page. The usefulness in **showcasing to stakeholders** received a **mean rating of 3.9**, while the usefulness for **development purposes** received a **mean rating of 3.1**. This essentially means that it got the job done and that the team and external participants would use it again.

The participants were asked to motivate their ratings and suggest improvements after rating the perspectives of the prototype. It was mentioned that it can ease the life of developers. It tells exactly where things need to be and how the actions should look when something is clicked. Developers can then easily replicate the design. It was mentioned that it would be an improvement to have more overlap with the development. This could be done through doing the work in the same sprints as development. As the design is very fast to make, the designs could be made simpler, but faster so that the development team can implement the simple designs quicker and then later add more extensive designs and functionality. Another improvement for development is the usage of existing design libraries that have already been coded. This can improve deign and speed up development. This was the reason for the difference in ratings between the showcasing and development, as two of

the developers reported that using existing design libraries would have made development much easier and faster.

Chart 4: Usefulness of the prototype stage within the team both for showcasing for stakeholders (above) as well as for development (below). Both ratings follow the same scale but due to limitations with Google forms they do not show the same measurement on the Y axis.

The participants reported the prototyping as giving value through allowing testing with customers early on, making sure they can adopt the product. They strongly recommended doing so before any coding is done. The prototype even allows testing for very detailed content topics.

The prototype has high **internal value** as it can visualize what needs to be developed, and it can allow for the team to work on backend and frontend at the same time. The prototype gives an overview of the product in its early stages, which is helpful because development can take a long time.

The prototype can show **external value** through showcasing with customers and giving stakeholders a pleasant way of using and testing the product. It can also help externally through marketing and even in business negotiations. The prototype gives a realistic expectation of the final product, compared to similar tools such as Microsoft's Power Point. This can help with the expectation management problem, which is that if stakeholders have too high expectations, it can make the delivery that much harder because they expect too much of the product. On the other

hand, too low expectations might make stakeholders lose interest. This is why a prototype can give realistic expectations of the final product, as long as it is presented as a prototype. Otherwise, it might fool customers that the product is closer to being done than it really is.

Essentially it boiled down to the fact that anything that can give insights or generates deeper understanding brings value, and that is exactly what the prototype does. The cost of creating the prototype is also low compared to developing the actual solution.

4.2.6 Handover stage results

In this section the developers in the project as well as the external participants were asked to answer this section, the ones from the project that did not do development did not have to answer this section, but they were allowed to. The usefulness of the design system was rated with a **mean of 3.6**. This meant that the development team and external developers found it useful and would use it in their work. The ratings can be seen in chart 5.

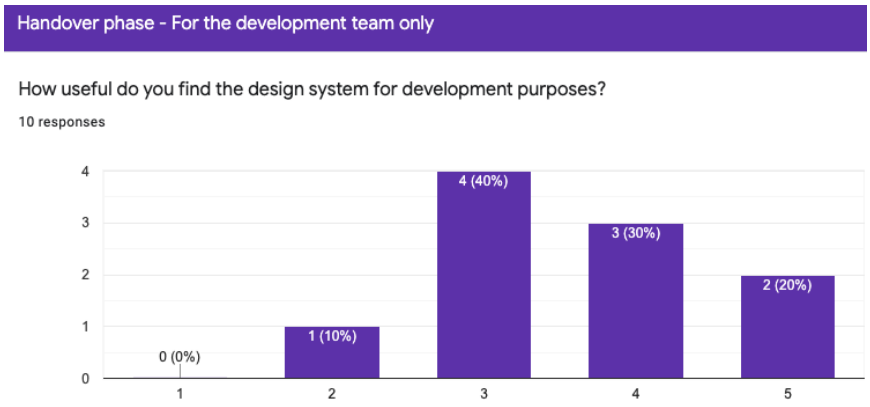

Chart 5: The usefulness of the design system.

The participants were asked to tell which parts they found the most helpful and which parts that could be improved. By far the most useful part according to the number of comments was the color schemes and font families, it was said that they could be very easily copied and pasted into development. The library was also found to be useful, and the navigation was said to be easy to understand and use. The component-based design was also seen as a plus. Lastly, showcasing the handover stage was seen as useful, because it showed many new elements and things to take into consideration.

For improvements it was mentioned that more existing libraries should be used such as Bootstrap and other component libraries like Material design by Google would have been welcomed in the design system.

Another improvement would be the collaboration between design and development. Naming the different sections and creating the design system more according to the developers needs would have made coding the solution more effortless.

The **ease of use of the design system** was rated with a **mean of 3.7**. This means that it was close to good to use, meaning that it almost gets the job done in all aspects and they would use it again. The results can be seen in chart 6.

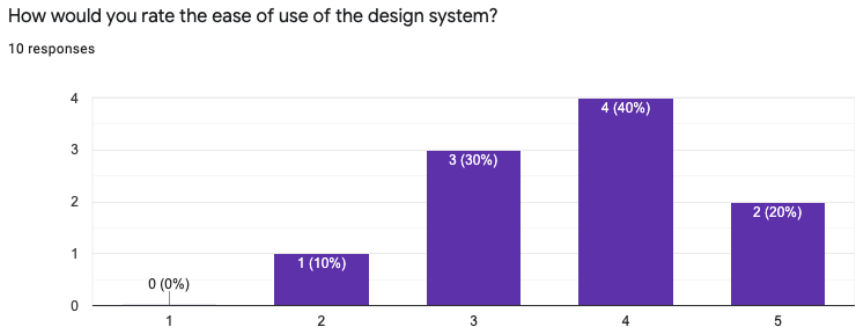

Chart 6: Ease of use of the design system.

When asked what worked well and what could be improved participants responded with the design system providing a good overview of all the components. Some would have preferred to have all things in the same page, but others did not. However, this is largely due to the size of the project, where there would otherwise be too much content on one page in the design system. A problem with the design system, especially within the team was that they did not really have the time to use it and implement the designs, which made it harder for them to respond and to learn it in the first place. But it was said that the developers should get to learn Figma a little bit better for the design system to give a bigger advantage. Nonetheless, after the walkthrough it was mentioned by the external developers that they could definitely make use of the design system, and it was said to be simple and clean.

Figma's D2C solution received a **mean rating of 3** with very high deviation. **Anima App** received a **mean rating of 2.9**, also with high deviation in the answers. Both were perceived as very useful by some and not at all useful by others. This was due to high variation in the backgrounds of the coders. All had previous experience in frontend coding, but the internal developers did not have much experience with frontend development compared to the external ones. This means that the results from Figma's own design to code solution and Anima App should get more proper testing before a real conclusion can be made. There were however many insights into their performance.

It was mentioned that **Figma's design to code** solution lacks the HTML code, which would be helpful. Some mentioned that it could be a good starting point, but most likely the code would need to be rewritten again due to its simplicity. Other believed this code to be a must in frontend development, because trying to implement design solely by visual coding often leads to anomalies that aren't present when using Figma's design to code solution properly. Figma's solution seemed very straight forward to newer coders, and they said that it could save time when coding.

Anima App in contrast to Figma, offers full code conversion, meaning that the code can be transformed into CSS, JavaScript, HTML as well as React code. The participants were asked about their experience with Anima as well as if they thought it showed potential. As previously mentioned, the answers had a high deviation. Some mentioned the code as being similar in use to Figma, which they did not see much use of, or that the code simply did not seem useful, and may be used for very simple components. They thought of it as giving more guidelines than actual usage. On the other side of the spectrum, people were intrigued to use it and could see themselves adopting it, but only after thoroughly going through it to reduce the margin of error. They thought the code seemed simple and easy to use and they saw value through having a component with all the classes already linked to it, which would save time when developing. A few things to take into consideration were that it would have been preferred with an all-in-one solution, where developers would not have to use two different software to look at designs. More importantly, the better the designs are created specifically for code exportation, the better they looked in the code. When designing for code exportation one must be very careful with naming of different elements and smaller components, otherwise the code might not even be readable. Also, the components need to follow standard coding practices such as responsive design if they are to be used by development. This is a tradeoff when deciding on the outcome of the design process. Either create designs for the purpose of designs, which is faster, or to create designs for export, which is slower but can generate better results in the long run depending on the development team.

Finally, when asked about what went well in the handover it was mentioned that it went well overall and good insights were given into what could be expected and how to achieve it. More instructions into using Figma was a wish for future projects because it seemed quite complicated for many.

4.2.7 Framework and outcome results

The outcome of the design process was rated according to the same scale as the usefulness of the different stages and deliverables. **The outcome of the design process** received a **mean rating of 3.8**, which can be seen in chart 7 on the following page. The team that had experience with the

process were also asked to rate their **own experience with the UX process**, which got a **mean rating of 3.7,** only the team at Contiot answered this question as they were the only ones with experience with the process. Most of the team had very little experience with UX and they stated that they learned much during the process. They also thought the process was useful, but they did not have the time to implement it during the time of the thesis.

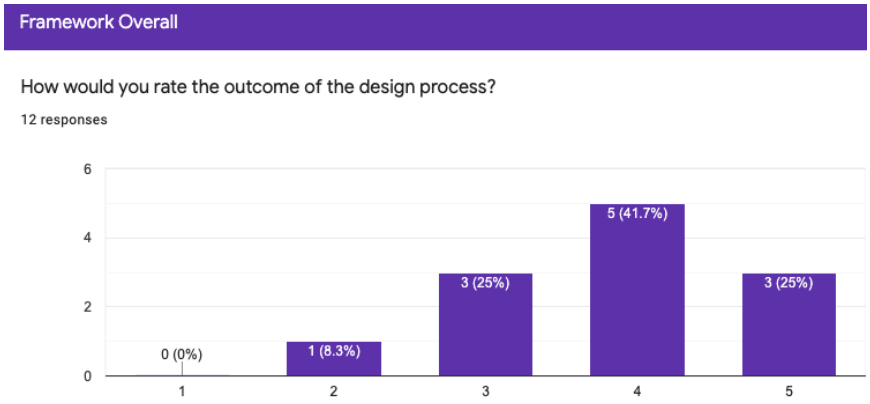

Chart 7: The outcome of the design process.

When asked about the outcome of the process a large variety of answers were presented about different aspects, so I decided to summarize them in bullet points to make them more visible. Below are listed the things that worked well in the framework:

- The framework speeds up the design process and brings on important discussions naturally.
- The framework provides a clear path from ideation to development.
- The outcome was good with designs ready for implementation, which will be useful for implementing the solution.
- The outcome was excellent considering the given time and resources available. It definitely showcases the usefulness of Lean UX in a startup environment.
- "The framework gave insights into what was being made and how to achieve it. After seeing this any investor or coder would have a good idea of what to expect and how it will work."
- "The approach was very methodical, and the designer realized his vision clearly, it would be pleasant to embark on the journey. It was straight forward from start to end. It could work well for both experienced and inexperienced people. "

The participants were also asked directly on what they thought could improve the framework. This was asked for more open-ended answers so that the participants could give feedback more freely. The improvements could be summarized into five different points. They are listed below in no particular order:

- Usage of more ready-made components.
- It might have been too early in the development process, or perhaps more work with development might have helped the process.
- “Coming from B2C the process might be a little bloated, and time to market could be cut shorter by skipping some steps of the wireframe. “
- The handover could be made smoother and perhaps more overlapping with development.
- “An all-in-one tool would be more pleasant to work with. “

5 Analysis

This chapter is for analysing the implementation of the framework as well as the results from the survey to answer the research questions. The analysis presents the research questions separately for clarification, and for the specificity of the research questions two and three.

5.1 What characterizes a good Lean UX framework?

1a *What kind of frameworks are currently being used in Lean UX?*

There are currently not many frameworks in Lean UX for designing for development. Though I was able to find a few when analyzing the background and the foundation of Lean UX in chapter two, background. The three foundations of Lean UX are **design thinking, Lean startup and agile development.** Analyzing these methodologies, I found a process where three steps of the design thinking process were implemented to create a similar platform to the one in my project. The three steps focused on the development aspects of Lean UX, skipping users research and Implementation. I found another framework that is created specifically to bridge the gap between design and development. Both the process and the framework, and even the double diamond model were used as inspiration, and for structuring the framework of this thesis.

1b *What are the different stages in a Lean UX framework?* & **1c** *What are the different methods and steps to take in each stage?*

Analyzing the background, I found that frameworks are a very loose term which means that the guidelines of creating one are almost nonexistent. This is partly because they vary so much between different areas that they are being applied to. Nonetheless, inspiration can be found for the task by looking at other works. Guidelines for Lean UX on the other hand do exist, they are also loose, but important principles can be applied. **Lean UX** as a methodology is **value driven**, which means that processes and artefacts are created based on the value that they are expected to bring. If a process, method or artefact does not add value, it should not be included in the process. Secondly **implementing different methods from the subareas of Lean UX means that the process becomes leaner**. Following these two principles the stages of the framework should be implemented only if they bring value and they include methods and elements from Lean UX, but the latter is not mandatory. According to the principles above I constructed my framework and tailored it to fit the scope of the thesis work. The stages included in my framework were **Define**, **Ideate**,

Prototype and **Handover**. The first three stages are based on the design thinking process in the study by Martins *et al.* (2020) and the final stage is based on an existing framework for bridging the gap between design and development by Navarro *et al.* (2016). A summary of the stages and their end deliverables can be seen in figure 25.

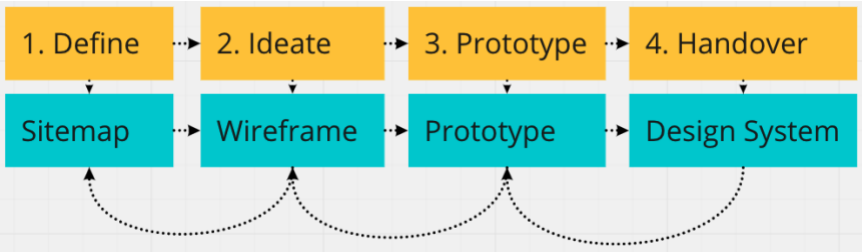

Figure 25: The stages of the framework. The phase is above in orange with the deliverables of each stage below in turquoise. Knowledge from each step is used for the next phase but the knowledge can also move backwards to be used again in the previous phases.

The **define stage** is used to **pool together all the collected data about users** and **create an understanding** or a **problem statement**. This can be done through creating for example buyer personas or customer journeys. In my process this included both and the process ended with a sitemap that was a pre requirement for the project work.

The **ideate stage** is used to **create tangible solutions** based on the previous stage. This can be done through **wireframing**. In this thesis the choice was between a digital and physical wireframe, but the digital solution was chosen due to working conditions and the ability to showcase to customers more easily. Wireframes make sure that the structure and flow between pages is right, and that all the data is correct. The last step of ensuring the data can also happen in the prototype stage.

The **prototype stage** is used to **create a version of the final design**. In this stage the solution is validated externally from stakeholders to make sure it works in the hands of the end users and that they like the look and feel of the product.

The **handover stage** was specifically created for an **optimal handover of the design to the development team** drawing inspiration from Navarro *et al.* (2016). This stage includes creating a **design system** including pattern libraries, and in this case following atomic design principles when creating design components. This stage also included using **design to code software** for a smoother handover to development and to speed up the process.

5.2 What are the current best tools and how can they be implemented into the Lean UX framework?

2a *What is the best tool for the define phase?*

For the define stage the best tool was **Miro**, which was found out in the results in chapter 4.1.1. Miro is a tool that essentially works like an **endless whiteboard** which is great for coming up with **new ideas, planning** and different types of **workshops**. In the comparison the newer tool FigJam received a higher rating but was overruled due to it still being in its infancy, and the team having previous experience with Miro. For future implementations, if the team is **using Figma, FigJam would be the recommended** tool for the define phase. Nonetheless, each of the compared tools are all state of the art and choosing either of them would get good results. The difference in final rating was very small and the actual use also depends on the preferences and previous experience of the users. It is also **recommended to use tools that are familiar** to the team to skip any introductory phase, this is done to save time and to capitalize on the team's previous skillset.

2b *What is the best tool for the ideation & prototype phase, and can they be combined for efficiency?*

Figma was the best tool for both the **ideation and prototype stages** due to its superiority in the comparison in chapter 4.1.2. The other tools are good as well, but for this project Figma's rating was the highest. Figma offers a free solution which is a clear advantage over the other tools. The other advantage was the collaboration that is better than Adobe XD, but equal to Microsoft Whiteboard. However, due to online video calls being so easy nowadays, the advantage of video calls within the tool loses some attractiveness. For optimal efficiency **digital wireframes** can be constructed using Figma, this makes them **easily convertible to a prototype** in the prototype stage. However, there are tradeoffs that need to be taken into consideration when striving for optimal efficiency. These will be mentioned in the later sections.

2c *What is the best tool or way for an optimal handover?*

The best tool or way of doing a handover was a more multifaceted question than was anticipated. It depends on the specific situation with factors such as the **developer's preferences, knowledge, skillset, the size of the product and the specificity of the final deliverable**. In this case the final deliverable was a prototype with the requirements of containing all the necessary data and having the final design, also including the most necessary functionality that would be needed to make the product work.

A **design system** was chosen as the optimal way of doing the handover, due to the scope of the company as they were going to start developing the final product at the end of the design process. The tool for the job in this case was Figma, but selecting the right tool heavily depends on what the rest of the project has been created by. **It is recommended to use the same tool as when creating the rest of the designs**.

The design to code tool **Anima App** was selected for use in combination with the design system as it showed great promise in the beginning. Figma's own D2C functionality was also selected for evaluation. For optimal use of both these solutions the practitioners should **apply coding principles** and **knowledge** when designing, this means that discussions with development becomes very important when using these tools. Otherwise, these types of tools could produce unreliable results if the design is not structured in a way that mirrors the way the code should look like. Therefore, it is only **recommended to use Anima and Figma's D2C functionality** if the designer has **coding experience** or if they can **discuss with the development** team before making the components and designs.

5.3 How well does the chosen solution perform and how could it be improved?

3a *What is the perceived usefulness of the produced artefacts?*

Define Stage

The Sitemap produced in the define phase had good usability for planning the DCC Hub and visualizing the solution to the team and everyone involved which was seen in the results in 4.2.3. With a **rating of 3.6** out of 5, there is still room for improvements. To most participants it ranged from good to excellent. Unfortunately, within the team some of the team members only saw the first iterations of the sitemap. During the construction of the wireframe many new insights were gathered that helped make a clearer structure for the sitemap. For this reason, the sitemap was a too complicated for some. Nonetheless, it shows the relation between the stages and how useful it can be to update the knowledge from previous stages based on new findings. This also **shows the importance of including the rest of the team in the process** to keep a shared understanding of the problem space and the artefacts. The most important functionality was the **overview of the platform** which clearly **shows all the screens needed** for the platform and **how everything is linked**.

The chosen tool **Miro** for the define stage was more than enough for this project, and it received a **mean rating of 4** when combining all of the evaluated aspects. However, for future projects **FigJam would be recommended if Figma is the chosen tool for the following stages**, because it can offer better synergy with Figma and having fewer different

programs in the Lean UX process is always a benefit to minimize the workload.

Ideation Stage

The wireframe was considered the most useful out of all the artefacts with a **mean rating of 4.3**, which was discovered in the results of the ideation stage in 4.2.4. The team had the highest exposure to the creation process of the wireframe and the possibility to give feedback during several occasions. The wireframe gave the team value through **giving new insights** that would otherwise not have been found, it enabled the **finding and including of all the needed data**. Though the ensuring of the data could also have been done in the prototyping stage, which could have shifted the perceived usability between the ideation and prototyping stage. The wireframe **gave some needed requirements** and **facilitated many discussions**. The choice for making a **digital wireframe** was an important step in the process. This **enabled the whole team to easily access the wireframe** and give feedback and **allowed external stakeholders to be involved** even though they were busy and could not physically attend.

Figma as a tool for both the ideation and prototype stage received a mean rating of 4. Using the same tool for both of these stages was a good idea based on the results of the survey, however this introduced certain decisions that have to be made. **Before starting to create the wireframe, the fidelity should be compared to the scope of the project.** If the scope is to deliver a B2C solution where it does not matter if the UX is not great or some data is missing, then skipping the prototype stage and only going for a digital wireframe can be enough. However, in most cases where it is important to get the data right, the solution must be good enough according to requirements then a prototype is needed. In this case it is good to use the same tool, but the fidelity of the wireframe can be kept lower to speed up the process.

Prototype Stage

The prototype was rated through two perspectives. Through the perspective of **showcasing stakeholders**, it received a **mean rating of 3.9.** For this purpose, it was perceived as very useful. The value for stakeholders comes from **visualizing the solution in a realistic way**. This helps with expectation management, that can otherwise be an issue when trying to showcase solutions to stakeholders.

From a **development perspective** it received a **mean rating of 3.1**. The prototype helps in **showing the team what needs to be built and how**. With a rating of 3.1, there is still room for improvement and additions to the prototype. Doing the wireframe with lower fidelity would have left more time to create the prototype. This could have benefitted especially the development team as more **components and libraries could have been discussed**. This further strengthens the finding of the **tradeoff between the wireframe and prototypes fidelity**. That was

mentioned in the previous section. When designing for reusability, throw away designs can still add much value. They can tell what needs to be designed and how, this makes the design for reuse much easier in the prototype stage if done correctly in the wireframe stage. Nonetheless, the prototype allowed for **showcasing with stakeholders very early on in the process**, and it was still recommended to create the prototype before any coding was done.

Handover Stage

The design system received a **mean rating of 3.6**, seen in the previous chapter at 4.2.6. Though the rating is still moderately high, it is likely that a higher rating could have been achieved had the team and external developers gotten to use the design system for a longer period. At the time the team only received a presentation after they checked through the system and the external developers went through it in the walkthrough for a limited time period as the walkthrough took at most an hour and a half.

It became evident that the whole design system was useful for both me as a designer when creating the prototype, but also for developers looking to implement the designs. The **color** and **font styles**, **navigation** and **component-based designs** were deemed the most useful aspects of the design system. In addition to the design system Figma's own CSS code was available as well as Anima App which converted the designs to React code.

Both D2C software's had very high deviation in their ratings and answers. The sample that answered the question also had different background and experience with frontend coding, which does affect the results. The results do however show that there are clear benefits to using these solutions, and that they show promise for future usage. The ratings from chapter 4.2.6 showed a **mean rating of 3 for Figma's own** code and a **2.9 for Anima Apps solution**. Considering this, they both seem to be very similar in performance, whereas Figma's solution seems easier to use being integrated and found directly in Figma, but Anima App offers more extensive code. To end this section, it needs to be said that **both tools require further testing** with more users to provide more reliable results but considering the novelty of the software's it is impressive what they have managed to do.

3b *What is the perceived usefulness of the framework overall?*

The perceived **process outcome** was rated with a **mean of 3.8** and the **teams own experience** was rated with a **mean of 3.7** in chapter 4.2.7. The framework received very positive feedback overall, when overlooking minor parts or features of it such as the design to code options. Overall, the participants were happy with the outcome. They thought that it **speeds up the process, brings on important discussion naturally, gives**

designs ready for implementation, insights into how the solution works and how to achieve it with a methodical approach. The results show that the Lean UX approach to the project was a good decision and that using a framework helped in the process by structuring the process. This helped everyone in the team follow along the process, especially with the end deliverables in each phase.

3c *How could the framework be improved?*

For improving the framework, I will only focus on the larger scale things that could be improved. This is because most small-scale details are very specific to the project, and as is often said in the design industry, the design is never ready, which means that it could always be improved. The whole reason for a framework in the first place is to abstract things for them to be replicable or reusable to another case that is different from the current one.

The **communication** in each stage of the framework should include discussions with the development team. This includes planning the design together for it to include principles and assets that the development team can later use. This became apparent from the survey where the development team wished for these aspects in the design. This is recommended in agile methodologies, but in my case working separately from the development team this was easier said than done. Nonetheless, for the framework the communication with the development team is a crucial aspect for a better handover and overall reusability in the process.

Another improvement is a clear step for **making the decision for the transition from wireframe to prototype**. If the wireframe is done physically this step is unnecessary, but if using a digital wireframe, it is important to consider. Firstly, to determine the fidelity of the wireframe, **staying too long in the wireframing stage can slow down the process**. As the architecture and structure of the design file may or may not have been considered at this stage it may be difficult to make large changes to the wireframe. Therefore, I recommend to either transition fast from the wireframe to the prototype if the wireframe will be thrown away afterwards. Otherwise, it is recommended to create certain parts of the wireframe without thought on structure, but after a few screens are created the document should be structured, elements should be made reusable and connected through the design file to make editing faster and easier. If the later of the options is chosen and the wireframe is done correctly the wireframe can potentially be turned into a prototype. This adds a great deal of reusability to the process on a step that can otherwise be time consuming.

Codesign and designing with ready-made components are other important improvements, especially creating the design system with the development team. Unlike the first improvement which is merely about communication and planning. The idea is to have sessions to carefully plan

and create the components in the design, including selecting different pattern and component libraries that are currently out there or better yet, ones that the development team already uses. This improvement could save time, improve quality, and produce a more failproof design. However, this makes for a tradeoff between quicker design or better-quality design as well as potentially losing some innovation in the design process. Though the level at which the development team is included can be chosen according to the designer's opinion and experience.

Lastly, an improvement to the framework would be using fewer tools and looking at more **all-in-one solutions**. Doing this helps the team in keeping focused on only one solution instead of having to download multiple solutions and learn each of them. Luckily Figma now offers possibilities for all the phases with the addition of the plugin Anima for Figma. However, it is very likely that Figma will create their own design to code solution considering how fast they update their solutions and how much they listen to their user-based forums.

6 Discussion

In this chapter I discuss the implications of the results of the study, the limitations that the study had, as well as recommendations and future work that could be done in in the area.

A Lean UX framework was successfully constructed at the project of this thesis. Both the project and the thesis reached their goals according to the scope. The reusability was one of the main targets of the framework and while I had expected a higher reusability through the framework, this study highlights how it can potentially be done through the implementation and the consideration of certain tradeoffs. This thesis highlights the usefulness and the need for future research of these types of frameworks.

The framework that was created shows how Lean UX principles can and should be included in software development projects, and more specifically in startups. Along the design process of the thesis work, Lean UX principles are considered also in the decision-making process throughout the thesis. The framework could be applicable to other types of companies as well considering the Lean UX approach, though the framework is specifically directed towards startups. It was seen and mentioned in the results that participants could also see the framework in B2C startups, but with minor adjustments. The framework was evaluated through a questionnaire and shows good potential for future implementations. However, there is still room for improvements of which the most important ones were presented as answers to the third research question.

6.1 Limitations

The most pressing limitation of this study was the **time frame**. The framework was researched, implemented and evaluated, but any actual implementation of the outcome never happened at the time of the thesis. This specifically affected the results of the handover, and more specifically the D2C software's. To give more valid results the team should have been given a longer time frame to get acquainted with the software's as well as Figma itself, and to implement the designs. This would have given data from real usage instead of assumptions made from just looking at the code.

Another limitation was the **population of the study**. Only including 12 people does not offer very high external validity to the study. The team of Contiot only consisted of seven members at the time of the questionnaire, which made for 100% of the sample. This is very low, but it was evened out with conducting walkthroughs with five external participants with a similar background. The external participants essentially worked as a test group for the sample, adding more weight to

the results. The study of the questionnaire in most parts was in more of a discoverable nature, thus the sample was still able to give meaningful insights and concerns. This makes for very accurate contextual results, but with limited generalizability. Nonetheless, the generalizability could potentially also be part of future work, determining if the process at Contiot could be applied to other types of companies and fields and testing out how well the process translates into them.

As mentioned in the beginning, the **pre-requirements** also limited the direction in which the study could go as certain deliverables had to be created. This limited the freedom to choose methods and tools that could be used throughout the process. Nonetheless, the pre-requirements were quite common to UX projects in general, which does improve the generalizability of the study.

The way the UX process was executed, **working in a silo** of sorts, limited the way that agile methodologies could be executed in the project. This allowed for less collaboration, thus less knowledge from the development team could be transferred into the process.

Lastly, a limitation and a problem in UX research in general, is the **lack of scientific articles about certain areas**, and **too many definitions of the same thing or similar definitions about different things**. UX is especially directed towards industry, which means that most practitioners are not UX renowned experts with a high enough academic background. This leads to a lack of scientific articles on areas and an abundance of web articles where authors stipulate their own definitions about things, which leads to the problem of too many definitions. This is problematic for a field that lacks structure and lexical definitions of much terminology. Standardization in the field of UX can potentially reduce freedom for designers, but in the current state of the field some standardization would be necessary as it can affect research in making it slower and potentially leading to misunderstandings.

6.2 Future work

There were still many things that were left out of this thesis or that deserve to be researched further. **D2C software** are new tools that show an interesting and novel way of doing things. They could potentially change the UX and development process in the future by automating or helping in the stage where design is implemented. This could significantly increase the power of UX designers to fulfill their designs and to help in projects. In this thesis they were not analyzed to their full extent. I firmly believe that they will have a large part to play in the future of UX and that they deserve more research.

Design for reusability is another factor that was only a small part of the thesis. This area is very interesting and especially valuable in Lean

UX, if artefacts can be reused throughout the process. Reusability was introduced in this thesis, but to get to the roots of the subject, more research is needed. This could be looked at in combination with comparing low and high fidelity in wireframing and prototyping with high and low reusability. In this thesis it became evident that to an extent they share the same curve, but at a certain point they go in opposite directions, especially in case of the prototype.

Finally different methodologies, frameworks and processes for **bridging the gap between design and development** could be very important for the evolution of UX. In the background chapter it becomes evident that there is a research gap, which indicate that the area could use some more research.

7 Conclusions

This study was conducted to build a Lean UX framework for generating a reusable product from start to finish. I conducted a **literature review** to find out what characterizes a good lean UX framework. Based on the literature review a Lean UX framework was **constructed and implemented** into Contiots project of constructing an online platform for creating, sharing and using digital calibration certificates. The implementation of the framework was analyzed through a **survey** collecting data on the perceived performance of the process and the deliverables.

The framework that was created was inspired by two existing frameworks and processes as well as the double diamond model, and was further based on principles from the subgenres of lean UX. The different stages of the framework consist of three of the five phases of the design thinking process **Define**, **Ideate** and **Prototype**. An additional stage was added called **Handover**. The handover stage was added by inspiration from a successful framework of bridging the gap between design and development as well as whishes from the project company.

The recommended tools for the framework were **Miro or FigJam** for the define stage. These tools are essentially used as endless digital whiteboards, giving teams the ability to brainstorm, test hypotheses and store information conveniently. **Figma** was recommended for both the ideation and prototype stage due to it giving abilities of creating both **digital wireframes and prototypes**. Figma was also eventually also used for the handover stage as it helped in creating the **design system** in combination with the prototype. For the D2C functionality both Figma and Anima App were evaluated. **Figma's own D2C solution** is recommended as it gives easy access to the CSS code of the design. **Anima App** is also recommended, but only with previous research on how to use the tool as it can be more complicated, and designers need to have some frontend knowledge or plan the design together with development to use it properly.

The results of the survey clearly showed the usefulness of the framework both in the perspectives of the present team members as well as external developers. The framework makes for a **methodical approach** that includes the team and creates a **shared understanding** of the problems and artefacts. To improve the usage of the framework there must be a focus on **communication between designers and developers** to ensure the quality and usefulness of the designs. **Codesigning** with stakeholders as well as the development team ensures that the designs are valid, and that usable pattern and component libraries are used. **An all-in-one solution** is recommended for improved efficiency in the framework as it reduces time spent on learning different tools. Finally, a **tradeoff between reusability and speed** needs to be considered when implementing the framework. This is especially seen in the wireframe to prototype stage, and it is introduced in the results of both as well as the analysis. It is also apparent in the handover stage, where designers must

choose between creating designs fast, or making the designs reusable for the development team with code export from the designs.

Finally, the study reveals a **research gap in the area where UX design meets development.** The UX design profession is evolving fast and more knowledge in frontend development is needed, especially in startups where time, resources and knowledge are limited. This study highlights an increasing need for research on frameworks and software for bridging the gap between UX design and development, as well as the demand for UX designers with the knowledge on how to use them.

References

5 Easy Steps to Creating a Sitemap For a Website (2019). Available at: https://www.quicksprout.com/creating-website-sitemap/ (Accessed: 8 July 2021).

A Simple Introduction to Lean UX (no date) *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/article/a-simple-introduction-to-lean-ux (Accessed: 20 March 2021).

Aarlien, D. and Colomo-Palacios, R. (2020) 'Lean UX: A Systematic Literature Review', in Gervasi, O. et al. (eds) *Computational Science and Its Applications – ICCSA 2020*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 500–510. doi: 10.1007/978-3-030-58817-5_37.

Ali-Yrkkö, J. (2019) *Digibarometri 2019: Digi tulee, mutta riittävätkö resurssit?*, *Etla*. Available at: https://www.etla.fi/julkaisut/digibarometri-2019-digi-tulee-mutta-riittavatko-resurssit/ (Accessed: 10 April 2021).

Almeida, F. and Monteiro, J. A. (2017) 'Approaches and Principles for UX Web Experiences: A Case Study Approach', *International Journal of Information Technology and Web Engineering (IJITWE)*, 12(2), pp. 49–65. doi: 10.4018/IJITWE.2017040103.

Anima App, A. (2021) *We've moved inhouse*, *Medium*. Available at: https://blog.animaapp.com/weve-moved-inhouse-293b1e2e3f5a (Accessed: 2 July 2021).

Ann, A. (2019) *The Marriage of Content Strategy & Sitemaps*, *Medium*. Available at: https://medium.com/ashleycrutcher/the-marriage-of-content-strategy-sitemaps-ae8bd6f3d66f (Accessed: 8 May 2021).

Atomic Design Methodology | Atomic Design by Brad Frost (no date). Available at: http://atomicdesign.bradfrost.com/chapter-2/ (Accessed: 8 July 2021).

Baldridge, R. (2021) *What Is A Startup?*, *Forbes Advisor*. Available at: https://www.forbes.com/advisor/investing/what-is-a-startup/ (Accessed: 11 May 2021).

Ball, J. (2019) *The Double Diamond: A universally accepted depiction of the design process*, *Design Council*. Available at: https://www.designcouncil.org.uk/news-opinion/double-diamond-universally-accepted-depiction-design-process (Accessed: 8 July 2021).

Batterbee, I. (2020) *Design frameworks: what are they and why you should be using them*, *Medium*. Available at: https://uxdesign.cc/design-frameworks-what-are-they-and-why-you-should-be-using-them-f480a7a5b349 (Accessed: 2 July 2021).

Beck, C. (2017) *On the Current State of Design Systems in UX*, *Medium*. Available at: https://medium.com/better-product/on-the-current-state-of-

design-systems-in-ux-4cd0aa1fad71 (Accessed: 11 May 2021).
Bekyarov, M. (2021) *From Design To Developer-Friendly React Code In Minutes With Anima*, *Smashing Magazine*. Available at: https://www.smashingmagazine.com/2021/01/design-developer-friendly-react-code-animaapp/ (Accessed: 2 July 2021).

Bexiga, M., Garbatov, S. and Seco, J. C. (2020) 'Closing the gap between designers and developers in a low code ecosystem', in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. New York, NY, USA: Association for Computing Machinery (MODELS '20), pp. 1–10. doi: 10.1145/3417990.3420195.

Bruun, A. *et al.* (2018) 'The role of UX professionals in agile development: a case study from industry', in *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. New York, NY, USA: Association for Computing Machinery (NordiCHI '18), pp. 352–363. doi: 10.1145/3240167.3240213.

Caddick, R. and Cable, S. (2011) *Communicating the User Experience: A Practical Guide for Creating Useful UX Documentation*. Hoboken, UNITED KINGDOM: John Wiley & Sons, Incorporated. Available at: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=698119 (Accessed: 8 July 2021).

Check, J. and Schutt, R. K. (2011) *Research Methods in Education*. SAGE Publications.

Cockayne, D. (2019) 'What is a startup firm? A methodological and epistemological investigation into research objects in economic geography', *Geoforum*, 107, pp. 77–87. doi: 10.1016/j.geoforum.2019.10.009.

CONTIOT – Continuity for IoT Data (no date). Available at: https://contiot.io/ (Accessed: 8 May 2021).

Costello, P. J. M. (2003) *Action Research*. London, UNITED KINGDOM: Bloomsbury Publishing Plc. Available at: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=472768 (Accessed: 8 July 2021).

Cramer, A. (2020) *User Personas: The Basics | Smaply Blog*. Available at: https://www.smaply.com/blog/personas (Accessed: 8 July 2021).

Croft, J. (2007) 'Frameworks for Designers', *A List Apart*, 12 June. Available at: https://alistapart.com/article/frameworksfordesigners/ (Accessed: 27 May 2021).

'Customer journey map: What it is and why you need one' (2021) *Delighted*, 7 January. Available at: https://delighted.com/blog/guide-to-customer-journey-mapping (Accessed: 8 July 2021).

Cyrillo, M. (2011) 'Lean UX: Rethink Development', *InformationWeek*, (1316), p. 40,42,44.

Dafir, J. (2016) *(1) What is The Difference Between Methodology and Framework? | LinkedIn*. Available at: https://www.linkedin.com/pulse/what-difference-between-methodology-framework-dafir-jacob/ (Accessed: 8 June 2021).

Dam, R. F. and Siang, T. Y. (2020) *What is Design Thinking and Why Is It So Popular?*, *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular (Accessed: 6 May 2021).

Dam, R. F. and Siang, T. Y. (2021) *5 Stages in the Design Thinking Process*, *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process (Accessed: 25 May 2021).

Daniel, A. (2019) *4 Reasons why Wire-Frames are Important during Website or Mobile App Development*, *Medium*. Available at: https://medium.com/fbdevclagos/4-reasons-why-wire-frame-is-important-during-website-or-mobile-app-development-46fabdf47190 (Accessed: 7 July 2021).

Definition of framework | Dictionary.com (no date) *www.dictionary.com*. Available at: https://www.dictionary.com/browse/framework (Accessed: 21 May 2021).

Designing Systems | Atomic Design by Brad Frost (no date). Available at: http://atomicdesign.bradfrost.com/chapter-1/ (Accessed: 4 June 2021).

Duffy, A. and Ferns, A. (1998) 'An analysis of design reuse benefits'. (1998) An analysis of design reuse benefits. In: Proceedings of the 12th International Conference on Engineering Design (ICED '99). Design Society, pp. 799-804.
ISBN 392297953X

Duffy, S. M., Duffy, A. H. B. and MacCallum, K. J. (1995) 'A design reuse model', in. Heurista, pp. 490–495. Available at: https://strathprints.strath.ac.uk/18791/ (Accessed: 5 July 2021).

Edison, H., Wang, X. and Abrahamsson, P. (2015) 'Lean startup: why large software companies should care', in *Scientific Workshop Proceedings of the XP2015*. New York, NY, USA: Association for Computing Machinery (XP '15 workshops), pp. 1–7. doi: 10.1145/2764979.2764981.

Effectiveness of Agile Compared to Waterfall Implementation Methods in it Projects: Analysis Based on Business Intelligence Projects (no date). doi: 10.1515/fman-2017-0021.

Ellis, S. (2008) *Frameworks, Methodologies and Processes*, *VS ELLIS*. Available at: https://vsellis.com/frameworks-methodologies-and-processes/ (Accessed: 8 June 2021).

Fontinelle, A. (2020) *What Exactly Is a Startup?*, *Investopedia*. Available at: https://www.investopedia.com/ask/answers/12/what-is-a-startup.asp (Accessed: 8 May 2021).

Gardner, J. (no date) *Brand Development Process vs. Framework*, *Jeremiah Gardner*. Available at: http://jeremiahgardner.com/blog/brand-development-process-vs-framework/ (Accessed: 8 June 2021).

Gibbons, S. (2016) *Design Thinking 101*, *Nielsen Norman Group*. Available at: https://www.nngroup.com/articles/design-thinking/ (Accessed: 25 May 2021).

Gothelf, J. (2013) *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly Media, Inc.

Hammarberg, K., Kirkman, M. and de Lacey, S. (2016) 'Qualitative research methods: When to use them and how to judge them', *Human reproduction (Oxford, England)*, 31. doi: 10.1093/humrep/dev334.

Hayes, L. (2008) *Reusability vs. Usability: Where to Draw the Line?*, *StickyMinds*. Available at: https://www.stickyminds.com/article/reusability-vs-usability-where-draw-line (Accessed: 7 June 2021).

Hokkanen, L., Xu, Y. and Väänänen, K. (2016) 'Focusing on user experience and business models in startups: investigation of two-dimensional value creation', in *Proceedings of the 20th International Academic Mindtrek Conference*. *AcademicMindtrek'16: Academic Mindtrek Conference 2016*, Tampere Finland: ACM, pp. 59–67. doi: 10.1145/2994310.2994371.

How to create a pattern library and why you should bother (no date) *Paul Boag - User Experience Advice*. Available at: https://boagworld.com/design/pattern-library/ (Accessed: 4 June 2021).

Ibragimova, E. (2016) *High-fidelity prototyping: What, When, Why and How?*, *Medium*. Available at: https://blog.prototypr.io/high-fidelity-prototyping-what-when-why-and-how-f5bbde6a7fd4 (Accessed: 7 July 2021).

ISO (2008) *ISO 9241-210:2010(en), Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. Available at: https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en (Accessed: 6 July 2021).

Jaye, H. (2021) *What's The Difference Between A Wireframe, A Prototype, And A Mockup?* Available at: https://careerfoundry.com/en/blog/ux-design/difference-between-wireframes-prototypes-mockups/ (Accessed: 23 May 2021).

Justinmind (2018) *Lean UX: how to get started*, *Medium*. Available at: https://uxplanet.org/lean-ux-how-to-get-started-bb3771697e2 (Accessed:

28 May 2021).

Kelley, K. *et al.* (2003) 'Good practice in the conduct and reporting of survey research', *International Journal for Quality in Health Care*, 15(3), pp. 261–266. doi: 10.1093/intqhc/mzg031.

Krout, K., Carrascal, J. P. and Lowdermilk, T. (2020) 'Lean UX research at scale: a case study', in *Proceedings of the Conference on Mensch und Computer*. New York, NY, USA: Association for Computing Machinery (MuC '20), pp. 53–59. doi: 10.1145/3404983.3405587.

Larusdottir, M. K. *et al.* (2018) 'UX in Agile before and during development', in *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. New York, NY, USA: Association for Computing Machinery (NordiCHI '18), pp. 984–987. doi: 10.1145/3240167.3240268.

Lauff, C. A., Kotys-Schwartz, D. and Rentschler, M. E. (2018) 'What is a Prototype? What are the Roles of Prototypes in Companies?', *Journal of Mechanical Design*, 140(061102). doi: 10.1115/1.4039340.

Law, L.-C. *et al.* (2009) 'Understanding, scoping and defining user experience: A survey approach', in. *Proc. CHI '09*, pp. 719–728. doi: 10.1145/1518701.1518813.

Leeson, J. (no date) *A Guide to Pattern Libraries: The Key to Design Consistency*, *Aiimi*. Available at: https://www.aiimi.com/insights/guide-to-pattern-libraries-the-key-to-design-consistency (Accessed: 4 June 2021).

Liikkanen, L. A. *et al.* (2014) 'Lean UX: the next generation of user-centered agile development?', in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. New York, NY, USA: Association for Computing Machinery (NordiCHI '14), pp. 1095–1100. doi: 10.1145/2639189.2670285.

Martins, N. *et al.* (2020) 'E-Marketplace as a Tool for the Revitalization of Portuguese Craft Industry: The Design Process in the Development of an Online Platform', *Future Internet*, 12(11), p. 195. doi: 10.3390/fi12110195.

Minimum Viable Product | Design Defined | InVision (no date) *Minimum Viable Product | Design Defined | InVision*. Available at: https://www.invisionapp.com/design-defined/minimum-viable-product/ (Accessed: 28 May 2021).

Minimum Viable Product (MVP) and Design - Balancing Risk to Gain Reward (2020) *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/article/minimum-viable-product-mvp-and-design-balancing-risk-to-gain-reward (Accessed: 6 May 2021).

MVP vs POC vs Prototype: What Does your Company Really Need?

(2018) *Skelia*. Available at: https://skelia.com/articles/mvp-vs-poc-vs-prototype-company-really-need/ (Accessed: 28 May 2021).

Navarro, P. L. *et al.* (2016) 'A Script-Based Prototyping Framework to Boost Agile-UX Developments', *Journal of Computer Science and Technology*, 31(6), pp. 1246–1261. doi: http://dx.doi.org/10.1007/s11390-016-1695-6.

Netinant, P. (2013) 'Design Reusability and Adaptability for Concurrent Software', *AASRI Procedia*, 5, pp. 133–139. doi: 10.1016/j.aasri.2013.10.069.

Pakkanen, J. *et al.* (2016) 'Achieving Benefits with Design Reuse in Manufacturing Industry', *Procedia CIRP*, 50, pp. 8–13. doi: 10.1016/j.procir.2016.04.173.

Partners, F. T. (2019) '5 Benefits of integrating UI/UX Design to your process', *FAHM Technology Partners*, 4 December. Available at: https://www.fahmpartners.com/5-benefits-of-integrating-ui-ux-design-to-your-process/ (Accessed: 11 May 2021).

Ponto, J. (2015) 'Understanding and Evaluating Survey Research', *Journal of the Advanced Practitioner in Oncology*, 6(2), pp. 168–171.

Revella, A. (2015) *Buyer Personas: How to Gain Insight into Your Customer's Expectations, Align Your Marketing Strategies, and Win More Business*. Somerset, UNITED STATES: John Wiley & Sons, Incorporated. Available at: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=1895818 (Accessed: 8 July 2021).

Saring, J. (2020) *How to Build Faster with Reusable UI Components in React*, *Medium*. Available at: https://codeburst.io/how-to-build-faster-with-reusable-ui-components-in-react-482397dec818 (Accessed: 7 June 2021).

Singh, M. (2019) *Moving From Waterfall To Agile With Kanban*. Available at: https://www.digite.com/es/blog/waterfall-to-agile-with-kanban/ (Accessed: 25 May 2021).

Stevens, E. (2019) *The Fascinating History of UX Design: A Definitive Timeline*. Available at: https://careerfoundry.com/en/blog/ux-design/the-fascinating-history-of-ux-design-a-definitive-timeline/ (Accessed: 11 May 2021).

Sviryda, A. (2019) *The Product Designer role: evolution or revolution?*, *Medium*. Available at: https://uxdesign.cc/product-designer-evolution-or-revolution-44e94e14891f (Accessed: 8 May 2021).

Vesselov, S. and Davis, T. (2019) 'The Rise of Design Systems', in Vesselov, S. and Davis, T. (eds) *Building Design Systems: Unify User Experiences through a Shared Design Language*. Berkeley, CA: Apress, pp. 1–11. doi: 10.1007/978-1-4842-4514-9_1.

Vieira, T. (2020) *A brief history of UX design and its evolution*, *TNW |*

Syndication. Available at: https://thenextweb.com/news/a-brief-history-of-ux-design-and-its-evolution (Accessed: 8 May 2021).

Villani, I. (2018) *Transform Customer Experience: How to Achieve Customer Success and Create Exceptional CX*. Newark, AUSTRALIA: John Wiley & Sons, Incorporated. Available at: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=5649350 (Accessed: 8 July 2021).
What is calibration, and how can you manage your calibration reports? (2019) *Netilion Blog*. Available at: https://netilion.endress.com/blog/what-is-calibration/ (Accessed: 10 April 2021).

What is Wireframing? (no date) *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/topics/wireframing (Accessed: 8 May 2021).

Wilhelm, A. (2014) 'What The Hell Is A Startup Anyway?', *TechCrunch*, 30 December. Available at: https://social.techcrunch.com/2014/12/30/what-the-hell-is-a-startup-anyway/ (Accessed: 11 May 2021).

Yevick, D. (2005) *A first course in computational physics and object-oriented programming with C++*. Cambridge ; New York: Cambridge University Press.

5 Easy Steps to Creating a Sitemap For a Website (2019). Available at: https://www.quicksprout.com/creating-website-sitemap/ (Accessed: 8 July 2021).

Appendix

Framework & Artefact Survey

Experience in UX

This section is for determining the teams familiarity and experience with UX.

What is your experience with UX and UX related tasks?

- ◯ None.
- ◯ I know what UX is and why it is used, but no more.
- ◯ I have some familiarity, for example applying it in a school project or similar for a short period of time.
- ◯ I have experience and know how to use UX related tools
- ◯ I am an expert

What are some tools and methods related to UX that you know of or have tried?

Your answer

What do you think is the value of UX in startups?

Your answer

What is your work title?

Your answer

Define phase

The outcome of the define phase was the sitemap. The sitemap states all the needed pages in the platform as well as their structure related to each other.

The evaluation of the method/deliverable goes by the following scale:

1. Not useful, it did not give me anything, no new insights or knowledge and i could not use it for anything

2. Somewhat useful, It gave me some insights, I could potentially find some use for it

3. Useful, It gave me insights or knowledge, I could use it for some aspects of my work

4. Very useful, It gave great insight or knowledge, I could definitely use it for my work

5. Exceptional, It is crucial for the success of the operation, it is of great value and is able to find new insights and knowledge that would otherwise not have been found. I use or will use for my work

The evaluation of the tool goes by the following scale:

1 - Poor, it fails in several aspects, you would not use it again

2 - Fair, it gets the job done, but may be lacking in some aspects, you might not use it again

3 - OK, It gets the job done in most aspects, you would use it again

4 - Good, It gets the job done in all aspects, you would use it again, it compares better than most similar tools

5 - Excellent, It gets the job done perfectly, in this aspect it might be your tool of choice. You consider it at the top with maybe one or two other tools in this aspect.

Do you think the sitemap was a useful tool for planning the DCC Hub and making the idea concrete?

	1	2	3	4	5	
Not useful	○	○	○	○	○	Exceptional

Why was or wasn't it useful?

Your answer

How did you like the functionality of Miro?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

How would you rate the ease of use with Miro

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

How would you rate the tool Miro for collaboration?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Would you use Miro for future projects?

- ○ Yes
- ○ No

Why? You can talk about the topics above in doubt.

Your answer

Ideation Phase

The outcome of this phase was the wireframe. The wireframe was created to bring life to the sitemap and create a flow of the screens listed in the sitemap. It was also used to create the initial structure of the pages, all the needed elements as well as getting the right information. This section is for determining the usability of the wireframe and the goodness of Figma as a tool for showcasing ideas and the product.

The evaluation of the method/deliverable goes by the following scale:

1. Not useful, it did not give me anything, no new insights or knowledge and i could not use it for anything

2. Somewhat useful, It gave me some insights, I could potentially find some use for it

3. Useful, It gave me insights or knowledge, I could use it for some aspects of my work

4. Very useful, It gave great insight or knowledge, I could definitely use it for my work

5. Exceptional, It is crucial for the success of the operation, it is of great value and is able to find new insights and knowledge that would otherwise not have been found. I use or will use for my work

The evaluation of the tool goes by the following scale:

1 - Poor, it fails in several aspects, you would not use it again

2 - Fair, it gets the job done, but may be lacking in some aspects, you might not use it again

3 - OK, It gets the job done in most aspects, you would use it again

4 - Good, It gets the job done in all aspects, you would use it again, it compares better than most similar tools

5 - Excellent, It gets the job done perfectly, in this aspect it might be your tool of choice. You consider it at the top with maybe one or two other tools in this aspect.

How useful was the wireframe in capturing the relevant information and giving insights about how the final product should be?

	1	2	3	4	5	
Not useful	○	○	○	○	○	Exceptional

Could you tell a bit more about your experience with the wireframe?

Your answer

How did you find Figma as a tool for showcasing the product and ideas?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

What worked well with showcasing with Figma and/or what did not work?

Your answer

Prototype phase

At this stage the information and structure of the platform had been determined. The prototype was created to resemble the final product and to get all the functionality right as well as the look and feel of the product. This section is for determining the outcome of the prototype, how well it portrays the final product as well as

The evaluation of the method/deliverable goes by the following scale:

1. Not useful, it did not give me anything, no new insights or knowledge and i could not use it for anything

2. Somewhat useful, It gave me some insights, I could potentially find some use for it

3. Useful, It gave me insights or knowledge, I could use it for some aspects of my work

4. Very useful, It gave great insight or knowledge, I could definitely use it for my work

5. Exceptional, It is crucial for the success of the operation, it is of great value and is able to find new insights and knowledge that would otherwise not have been found. I use or will use for my work

The evaluation of the tool goes by the following scale:

1 - Poor, it fails in several aspects, you would not use it again

2 - Fair, it gets the job done, but may be lacking in some aspects, you might not use it again

3 - OK, It gets the job done in most aspects, you would use it again

4 - Good, It gets the job done in all aspects, you would use it again, it compares better than most similar tools

5 - Excellent, It gets the job done perfectly, in this aspect it might be your tool of choice. You consider it at the top with maybe one or two other tools in this aspect.

How useful did you find the prototype in terms of showcasing stakeholders.

	1	2	3	4	5	
Not useful	○	○	○	○	○	Exceptional

How useful did you find the prototype in terms of development? (for development team)

	1	2	3	4	5	
Not useful	○	○	○	○	○	Exceptional

Why and what could be improved?

Your answer

Do you think the prototype gives the company value, and if so how?

Your answer

Handover phase - For the development team only

This last section is for evaluating the design system, handover and design to code program.

The evaluation of the method/deliverable goes by the following scale:

1. Not useful, it did not give me anything, no new insights or knowledge and i could not use it for anything

2. Somewhat useful, It gave me some insights, I could potentially find some use for it

3. Useful, It gave me insights or knowledge, I could use it for some aspects of my work

4. Very useful, It gave great insight or knowledge, I could definitely use it for my work

5. Exceptional, It is crucial for the success of the operation, it is of great value and is able to find new insights and knowledge that would otherwise not have been found. I use or will use for my work

The evaluation of the tool goes by the following scale:

1 - Poor, it fails in several aspects, you would not use it again

2 - Fair, it gets the job done, but may be lacking in some aspects, you might not use it again

3 - OK, It gets the job done in most aspects, you would use it again

4 - Good, It gets the job done in all aspects, you would use it again, it compares better than most similar tools

5 - Excellent, It gets the job done perfectly, in this aspect it might be your tool of choice. You consider it at the top with maybe one or two other tools in this aspect.

How useful do you find the design system for development purposes?

	1	2	3	4	5	
Not useful	○	○	○	○	○	Exceptional

Could you mention further on which aspects were or could be the most helpful and what should be improved?

Your answer

How would you rate the ease of use of the design system?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

What worked well and what could be improved?

Your answer

How would you rate Figmas own design to code option?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Could you tell more and also about how it can be used?

Your answer

How would you rate Anima App

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Could you tell more about your experience with Anima?

Your answer

Do you think that Anima shows future potential and if so how?

Your answer

What went well with the handover and what could be improved?

Your answer

Framework Overall

This section is for determining the overall satisfaction with the design process in accordance with the framework.

The evaluation of the outcome and your experience goes by the following scale:

1 - Poor, it fails in several aspects, you would not use it again

2 - Fair, it gets the job done, but may be lacking in some aspects, you might not use it again

3 - OK, It gets the job done in most aspects, you would use it again

4 - Good, It gets the job done in all aspects, you would use it again, it compares better than most similar tools

5 - Excellent, It gets the job done perfectly, in this aspect it might be your tool of choice. You consider it at the top with maybe one or two other tools in this aspect.

How would you rate the outcome of the design process?

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Why?

Your answer

How would you rate your own experience with the UX process?

	1	2	3	4	5	
Poor	○	○	○	○	○	Exellent

Why?

Your answer

Is there something that you would like to add about any of the steps in the UX framework or otherwise?

Your answer