

# **Rapid design using web based UI design tools, Case: Contriboard**

Heli Sutinen

Bachelor's Thesis  
December 2014

Degree Programme in Media Engineering  
School of Technology, Communication and Transport



JYVÄSKYLÄN AMMATTIKORKEAKOULU  
JAMK UNIVERSITY OF APPLIED SCIENCES



Description

Author(s) Sutinen, Heli	Type of publication Bachelor's thesis	Date 12.12.2014
		Language of publication: English
	Number of pages 75	Permission for web publication: x
Title of publication <b>Rapid design using web based UI design tools</b> Case: Contriboard		
Degree programme Media Engineering		
Tutor(s) Niemi Kari		
Assigned by JAMK University of Applied Sciences, Need for Speed Program, Rintamäki Marko		
<b>Abstract</b>  The thesis is assigned by JAMK University of Applied Sciences and Digile Need for Speed Program (N4S). N4S Program acts as an accelerator for a new ways of working from Agile and LEAN development to Real-time Delivery. The program focuses on research in three areas: real time value delivery, deep customer insight and mercury business - finding the new money. The purpose of the thesis was to research web based UI design tools and the effect of tools being part of the software development process. One of the main research questions was to discover if UI design tools speed the development process when used in designing.  The research of web based UI design tools was conducted by evaluating and selecting the most suitable ones to support the development process of a product called Contriboard. Contriboard is a brainstorming and collaboration tool and reference product for Need for Speed Program developed by N4S@JAMK team. The evaluation was executed by tripartite testing of UI design tools. The first part included testing of larger amount of tools with the help of three summer trainees. The results were gathered using a web-inquiry created by the author of the thesis. The inquiry results helped in choosing tools for closer testing. The second part of testing was implemented by testing functionalities required in rapid designing. Finally, the selection of most suitable tools for designing of Contriboard was tested in the design process.  The testing tools in action indicated that UI design tools not just enabled faster designing but increased the communication between stakeholders as well. Understanding of the right kind of design methods and using the right kind of UI design tools present an advantage for speeding design process.		
<b>Keywords/tags (<a href="#">subjects</a>)</b> UI design tools, wireframe, prototype, mockup, service design, user experience, Contriboard		
<b>Miscellaneous</b> Agile, DevOps, LEAN, LEAN startup methodology		



Tekijä(t) Sutinen, Heli	Julkaisun laji Opinnäytetyö	Päivämäärä 12.12.2014
	Sivumäärä 75	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x
Työn nimi <b>Rapid design using web based UI design tools</b> Case: Contriboard		
Koulutusohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Niemi Kari		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu, Need for Speed-ohjelma, Rintamäki Marko		
Tiivistelmä		
Työ toteutettiin Jyväskylän ammattikorkeakoulun toimesta Need for Speed-ohjelmassa (N4S). N4S-ohjelma toimii kihdyttimenä ketterien menetelmien sekä LEAN-ajattelun kautta tulleiden uusien reaalialkaisten liiketoimintamallien kokeilussa tuottaen arvoa reaalialassa syvällisellä asiantuntemuksella. Opinnäytetyön tarkoituksesta oli tutkia käyttöliittymän suunnittelun tarkoitettuja web-pohjaisia työkaluja osana suunnitteluprosessia. Opinnäytetyön tärkeimpänä tavoitteena oli löytää vastaus, onko työkalujen käytöllä vaikutusta tuotekehitysprosessin nopeuttamisessa.		
Selvitys toteutettiin arvioimalla käyttöliittymän suunnittelutyökaluja, rajaamalla parhaat vaihtoehdot ja käyttämällä niitä osana Contriboard-tuotteen kehittämistä. Contriboard on yhteistyötä edistävä ideointityökalu, jota kehittää N4S@JAMK-tiimi. Samalla tuote toimii referenssinä N4S-ohjelmassa. Web-pohjaisia suunnittelutyökaluja löytyy huomattavia määräriä pienillä ja suurilla eroavaisuuksilla. Selvitystyö toteutettiin kolmivaiheisella testauksella sisältäen nykytilanteen määrittelyn ja kyselyn työkaluista, yksityiskohtaisemman sisällön testauksen sekä testauksen käytännössä osana Contriboard-tuotteen kehitysprosessia. Ensimmäisen vaiheen kyselyssä oli mukana kolme työharjoittelijaa. Testauksen ensimmäisen ja toisen vaiheen perusteella valittiin työkalut kolmanteen vaiheeseen.		
Käytännön testaus osoitti työkalujen käytön parantavan suunnitteluprosessia, ei pelkästään nopeuttamalla, mutta myös parantamalla yhteistyötä sekä kommunikointia tiimin sisäisesti. Ymmärtämällä oikeanlaiset menetelmät ja käyttämällä oikeanlaisia käyttöliittymän suunnittelun tarkoitettuja työkaluja saavutetaan nopeammin tuloksia.		
Avainsanat ( <a href="#">asiasanat</a> ) Käyttöliittymän suunnittelutyökalut, ratalankamalli, luonnos, prototyyppi, palvelumuotoilu, käytettävyys, asiakaslähtöinen suunnittelu, Contriboard		
Muut tiedot Ketterä, DevOps, LEAN		

## Content

<b>Abbreviations and terms.....</b>	<b>5</b>
<b>1 Introduction .....</b>	<b>7</b>
1.1 Background, definitions and objectives .....	7
1.2 N4S-Program .....	8
<b>2 Changes in development of web services .....</b>	<b>8</b>
2.1 Cloud computing .....	8
2.2 Waterfall model to agile models .....	10
2.3 LEAN startup methodology .....	13
2.3.1 Continuous integration to continuous delivery.....	13
2.3.2 MVP .....	14
2.3.3 Pivot .....	15
2.4 DevOps as a trend .....	16
2.4.1 Teamwork .....	17
2.4.2 Fast releases, fast responses, fast fails.....	17
2.4.3 Continuously improving reference production environment COROLLA .	18
2.5 Service design .....	19
2.5.1 Deep customer insight .....	19
2.5.2 Service design thinking.....	19
2.5.3 Service design tool: Customer journey mapping .....	20
2.5.4 Customer experience and social media marketing .....	21
2.5.5 Changing of the service ecosystem .....	22
2.6 Wireframes, prototypes and mockups .....	23
2.6.1 Wireframe .....	23
2.6.2 Prototype .....	24
2.6.3 Mockup .....	24
<b>3 Process and methods supporting N4S-Program .....</b>	<b>25</b>
3.1 Process cycle.....	25
3.2 Using UI design tools as a part of design process.....	27
<b>4 UI design tool testing requirements and implementation.....</b>	<b>28</b>
4.1 Requirements for evaluation.....	28
4.2 Comparison criteria .....	29
4.3 Cost structure .....	29
4.4 Test implementation .....	30

<b>5 Research and evaluation of UI design tools .....</b>	<b>31</b>
5.1 The status quo and survey.....	31
5.2 Deeper testing of features and functionalities.....	32
5.2.1 Concept.ly .....	33
5.2.2 Fluid UI .....	34
5.2.3 InVision App .....	37
5.2.4 NinjaMock .....	40
5.2.5 Proto.io.....	42
5.2.6 ProtoShare .....	44
5.2.7 UXPin .....	48
5.2.8 Comparison results of the second phase .....	50
<b>6 Case: Contriboard .....</b>	<b>52</b>
6.1 History and present of Contriboard.....	52
6.2 Service Design Workshop .....	53
6.2.1 Defining objectives.....	53
6.2.2 Gathering test data .....	53
6.2.3 Integrating observations and results.....	53
6.3 Testing UI design tool in design process of Contriboard .....	54
6.3.1 Low-fidelity mockups of UI with NinjaMock .....	54
6.3.2 UI mockups clickable with InVision App.....	56
6.3.3 Fast landing page mockup for Contriboard using UXPin.....	59
<b>7 Conclusions .....</b>	<b>61</b>
7.1 Research results .....	61
7.2 Research critique .....	62
7.3 Reflections on the research.....	63
<b>References .....</b>	<b>64</b>
<b>Appendices .....</b>	<b>67</b>
Appendix 1. Reference production environment COROLLA.....	67
Appendix 2. Customer Journey Canvas .....	68
Appendix 3. UI design tool survey questions .....	69
Appendix 4. UI design tool survey responses.....	70

## Figures

Figure 1. Growth of cloud benefits in 2014 .....	9
Figure 2. Waterfall model .....	10
Figure 3. Elements of using Scrum for agile software development .....	13

Figure 4. Right side of the environment shows tools used for continuous integration and delivery .....	14
Figure 5. Pivoting when using the build-measure-learn feedback loop .....	16
Figure 6. Venn Diagram illustration showing the collaboration enabling agility .....	17
Figure 7. Visual image of the DevOps Cycle.....	18
Figure 8. Main touchpoints of Contriboard .....	21
Figure 9. Wireframe made using NinjaMock UI design tool .....	24
Figure 10. Mockup merged inside photo of a platform.....	25
Figure 11. Innovation process.....	26
Figure 12. Compounded Agile DevOps-like process cycle .....	27
Figure 13. Main view of Concept.ly .....	33
Figure 14. Simple user interface, library icons on left .....	35
Figure 15. Screen diagram of Fluid UI demo.....	36
Figure 16. Projects view.....	38
Figure 17. Comment and drawing inside comment is added .....	39
Figure 18. Project management in NinjaMock .....	40
Figure 19. Outlook of NinjaMock UI in wireframing phase .....	41
Figure 20. Project preview and comments .....	41
Figure 21. Proto.io has simple UI .....	43
Figure 22. Browser testing with dark device .....	43
Figure 23. Project view of ProtoShare .....	45
Figure 24. ProtoShare UI workflows for finding library elements .....	46
Figure 25. Some of the library elements of ProtoShare .....	47
Figure 26. UXPin user interface has many buttons.....	49
Figure 27. Comments are shown as colored circles.....	49
Figure 28. Illustration showing the transformation of Contriboard .....	52
Figure 29. Edit/delete mode functionality and number of taps .....	55
Figure 30. Idea of select to delete ticket .....	56
Figure 31. Screenshot of ticket view left and boards view right.....	57
Figure 32. Mockups needed to include two sidebar ideas .....	58
Figure 33. Activity view of InVision App .....	58
Figure 34. First wireframe of landing page of Contriboard .....	59
Figure 35. All the elements used in mockup landing page .....	60
Figure 36. High-fidelity mockup and final landing page .....	61

## Tables

Table 1. Summary of UI design tool grading points from inquiry .....	32
Table 2. Plans and pricing for Concept.ly.....	34
Table 3. Plans and pricing for Fluid UI .....	37
Table 4. Plans and pricing for InVision App .....	39
Table 5. Plans and pricing for NinjaMock .....	42
Table 6. Plans and pricing for Proto.io.....	44
Table 7. Plans and pricing for ProtoShare.....	48
Table 8. Plan content and pricing for UXPin .....	50
Table 9. Summary of UI design tools price categories.....	51

## **Abbreviations and terms**

### **Cloud computing**

Also referred as “the cloud” is the delivery of on-demand computing resources which covers everything from applications to data centers over the internet on a pay-for-use basis.

### **Cloud services**

IT resources, as like services dynamically offered to customers via internet.

### **Development and operations (DevOps)**

Software development method. Emphasizes communication, collaboration and integration between developers and operations professionals in IT companies.

### **IaaS = Infrastructure as e Service**

### **Kanban**

Tool for managing software development process. Kanban-board is a whiteboard and post-its on board indicate features developed in projects.

### **MVD = Minimum Viable Design**

### **Minimum viable product (MVP)**

Strategy used in product development for fast and quantitative market testing of a product or feature of a product.

### **Mockup**

The visual design which represents the structure of information, visualizes the content and demonstrates the basic functionalities in a static way.

### **PaaS = Platform as a Service**

### **Portability**

Measure of a system’s ease of moving from one platform to another.

### **Prototype**

Middle to high-fidelity representation of the final product, which simulates user interface interaction. It should allow user to experience and test content and interactions in a way similar to the final product.

### **Public cloud**

Solution available to the general public, for example Google Docs.

### **Software as a service (SaaS)**

Software licensing and delivery model. Software is licensed on a subscription basis and is centrally hosted on the cloud.

### **UI = User Interface**

**User Experience (UX)**

A person's perceptions and responses that result from the use or anticipated use of a product, system or service.

**Wireframe**

Serves as a plan how to build the user-interface. Low-fidelity representation of a design, showing only the main groups of content, structure of information and a description and basic visualization of the user-interface interaction.

**WP = Work package**

## 1 Introduction

### 1.1 Background, definitions and objectives

Today when the Lean Startup and the DevOps (development and operations) way of thinking is widely known, it becomes nonprofessional and unprofitable to build a web based product before evaluating it quickly. This thesis is about researching does using UI design tools support rapid designing and speed up the development process. Thesis also helps in mapping out the right kind of UI design tools to test the product before going further to developing. Thesis is made inside the Need for Speed-Program, authored by JAMK University of Applied Sciences.

User interface design differentiates of the print design mostly of its interaction part. Considering the author's background as a print designer, motivation for this thesis came up with the urge to provide faster designing and better user experience through deeper customer insight and understanding in the field of IT industry.

Using UI design tools as a part of the design process is an area that has some general publications and visible proof. Continuous progression of tools and development of new ones is still making every research count. The high speed of future development is driving to faster development of products. Using one tool for designing wireframes, prototypes, and mockups would be ideal and rapid. The research questions focus on speed and agility in designing and development.

The questions to be researched in this these are as follows:

- Is there a right kind of UI design tool to be used in every situation?
- Does the designer provide more effective results and better collaboration by using UI design tools?
- Do UI design tools provide a more effective way to increase the speed of development process?

## 1.2 N4S-Program

Need for Speed-Program is a four-year program (2014–2017) established by Digile and funded by Tekes. The consortium consists of 11 large industrial organizations, 15 small and medium-sized businesses, research institutes and universities including JAMK University of Applied Sciences. Need for Speed (N4S) is determined to create the foundation for the Finnish software intensive businesses in the digital economy. N4S adopts a real-time experimental business model, and provides capability for instant value delivery based upon deep customer insight. (Paradigm Change – Delivering Value in Real Time 2014.)

The program has three major areas of focus (Järvinen 2014.):

WP 1: Delivering value in real time

WP 2: Deep customer insight — Better business hit-rate

WP 3: Mercury business — Find the new money

JAMK is participating for N4S-Programs packages with several innovative demos and theses. The subject of this thesis and reference product for N4S (introduced later in its own chapter) that the team is developing is most close to work package 1. Work package 1 provides approaches, methods and tools for agile and inexpensive designing, when creating and evaluating new products and services. Intention is to enable potential customers to evaluate product before the company invests heavily into actual development. (Turunen 2014.)

## 2 Changes in development of web services

### 2.1 Cloud computing

The popularization of the term cloud computing can be traced back to the early 2000s; however, the idea of ICT marketing going to service side is much older. In the year of 1960 a computer scientist, father of artificial intelligence John McCarthy developed the idea of networking, which allowed storing data on a remote server accessible via the internet. He predicted that in the future IT would be offered to

companies and consumers as services, not as products. The networking innovation acted as a vanguard for cloud computing. (Childs 2011.)

The history of web service development includes many key terms and main concepts that have been the driving force towards the success of cloud services. This large group of related factors, such as SaaS (Software as a Service), IaaS (Infrastructure as a Service), PaaS (Platform as a Service), virtualization, grid computing and utility computing has enabled transformation to cloud. (Salo 2012, 34.)

Clouds service architecture is divided in three layers; IaaS, PaaS and SaaS. Infrastructure (IaaS) creates a foundation for service platform (PaaS) and over the top can be built software (SaaS). Software as a Service is most known concept. SaaS means using software as a service instead of buying, installing, maintaining and possessing it. (Salo 2010, 22.)

In the year 2014 IT organizations see cloud as a huge benefit and a critical component of growth and profits. The 2014 survey of State of the Cloud Report shows results of the benefits of cloud (see Figure 1). The most significant benefits that organizations are realizing are greater scalability, faster access to infrastructure, higher availability and faster time to market for applications. The barriers to adoption are disappearing. (Rightscale, 2014.)

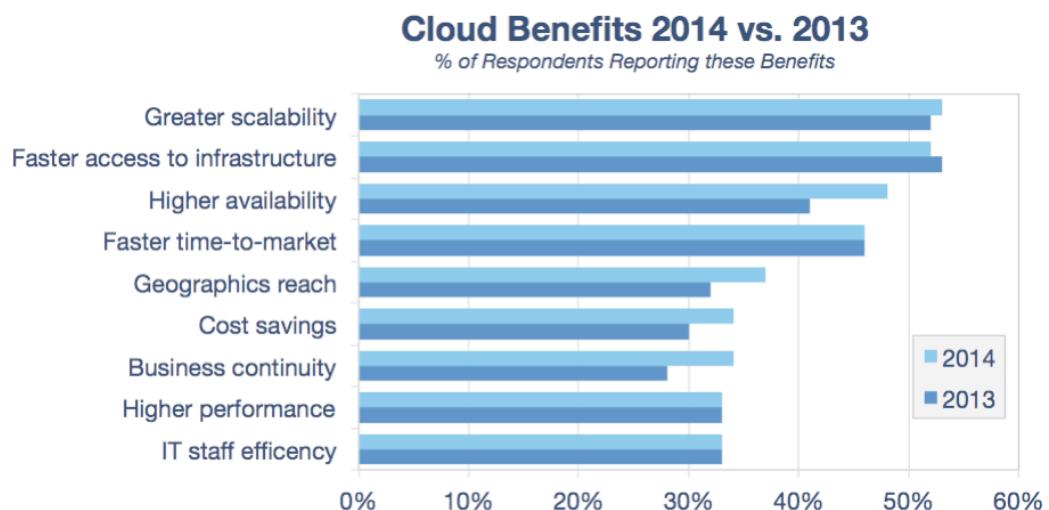


Figure 1. Growth of cloud benefits in 2014 (Rightscale 2014)

## 2.2 Waterfall model to agile models

Software development methodologies are rapidly evolving due to constantly changing technologies and new demands of users.

Historically the waterfall model has been associated with a large amount of project documentation. However, when the waterfall model is executed thoroughly, a very good set of documentation is produced and it can be a clear advantage. There are detailed lists of requirements, product specifications and software designs. All the deliverables of one phase have to be reviewed and approved before going to the next phase (see Figure 2). There are, however, some dangers using the waterfall model; usually in projects the real problems come out late, as in the testing phase. It means that in the waterfall method, if the team does not succeed in the testing phase they have to go back in to the previous phase or if the clients change their mind about some requirements, the team would have to return to the requirements and design phase and add new requirements and in a worst case scenario change the whole code afterwards. The waterfall model assumes that every requirement can be identified before any design or coding occurs, thus they leave no space for ideas and innovative thinking. (All About the Waterfall Model 2014.)

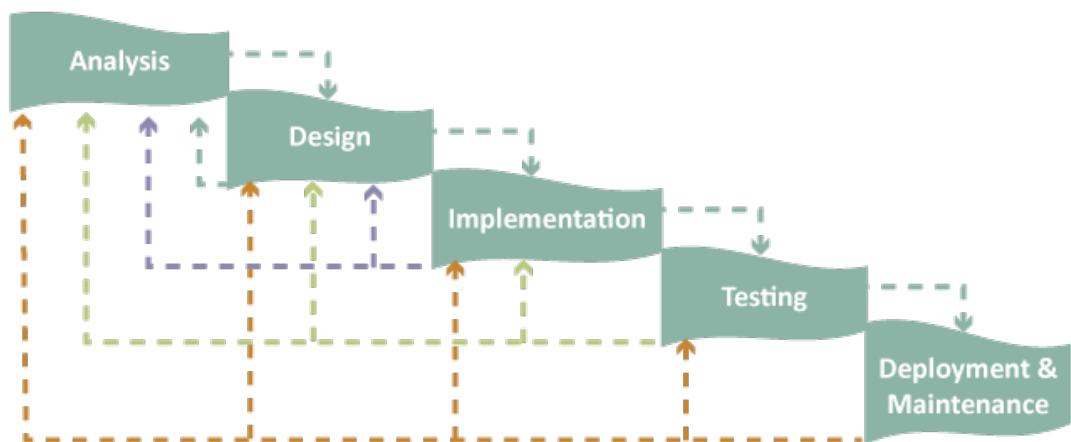


Figure 2. Waterfall model (Gladkova 2012, modified.)

Several different methodologies mixed with old and new ideas began to stand out in the late 1990s. Methodologies underlined the idea of close collaboration between the programmer team and business experts. In contrast to waterfall model face-to-

face communication was emphasized more than written documentation. (What is Agile 2014.)

One of the major buzzwords of IT development is agile. Agile development is another different way for IT projects and development teams. In the year 2001 a small group of IT development experts got together to discuss failures of traditional approaches managing software development projects. The group of people came up with the agile manifesto, which describes values that are still relevant today. (Waters 2007.)

Values written in agile manifesto:

*We are uncovering better ways of developing software by doing it and helping others do it.*

*Through this work we have come to value:  
Individuals and interactions over processes and tools.  
Working software over comprehensive documentation.  
Customer collaboration over contract negotiation.  
Responding to change over following a plan.  
(Manifesto for Agile Software Development 2001.)*

There is not just one set of agile best practices that fits for all development companies. There are various methodologies that are known agile, like DSDM, XP (Extreme Programming), Scrum and many others. Many companies have reached to more successful results by starting to make things more agile. Agile way is not just doing things using methods, it is finding agile ways of working and adjusting those methods in a way that fits best for the project. (What is Agile 2014.)

Scrum is the most commonly used agile software development process. In Scrum, the project moves forward through series of iterations (see Figure 3). These iterations are called sprints and typically each sprint is two to four weeks long.

**Product owner** is the key stakeholder of the project and represents users and customers. **Scrum team** works together towards a common goal and to complete the set of work they have committed to complete within a sprint. **ScrumMaster** is a person who is responsible for making sure that the team is working as productively as possible. ScrumMaster helps the team to use the Scrum process by removing barriers standing in the way of progress. The product owner creates a product

backlog. **Product backlog** contains all of the product features listed and prioritized. The product owner presents the top items from the product backlog to the team in a sprint planning meeting. **The sprint planning meeting** is held at the start of each sprint. Scrum team selects the work they are determined to complete during the sprint. The work selected from the product backlog is moved to **sprint backlog**.  
(Topics in Scrum 2014.)

Daily Scrum is a brief (10—15 minutes) meeting held every day during the sprint. Daily Scrum meeting is also known as Daily Standup because usually team members are standing up during daily meetings and when standing up, the meeting will stay short. The meeting held daily helps the team stay on track what is happening and what others have done, are doing currently, what are they going to do next and is there any impediments needing attention. (Yip 2011.)

Sprint review meeting and sprint retrospective are held at the end of every sprint. In the sprint review meeting the team demonstrates what they have accomplished during the sprint. Sprint is success when all the work from sprint backlog is accomplished. Sprint retrospective is a meeting where the team including ScrumMaster and product owner reflect on how well Scrum is working and how it could work even better. (Scrum Methodology: Scrum Meetings 2011.)

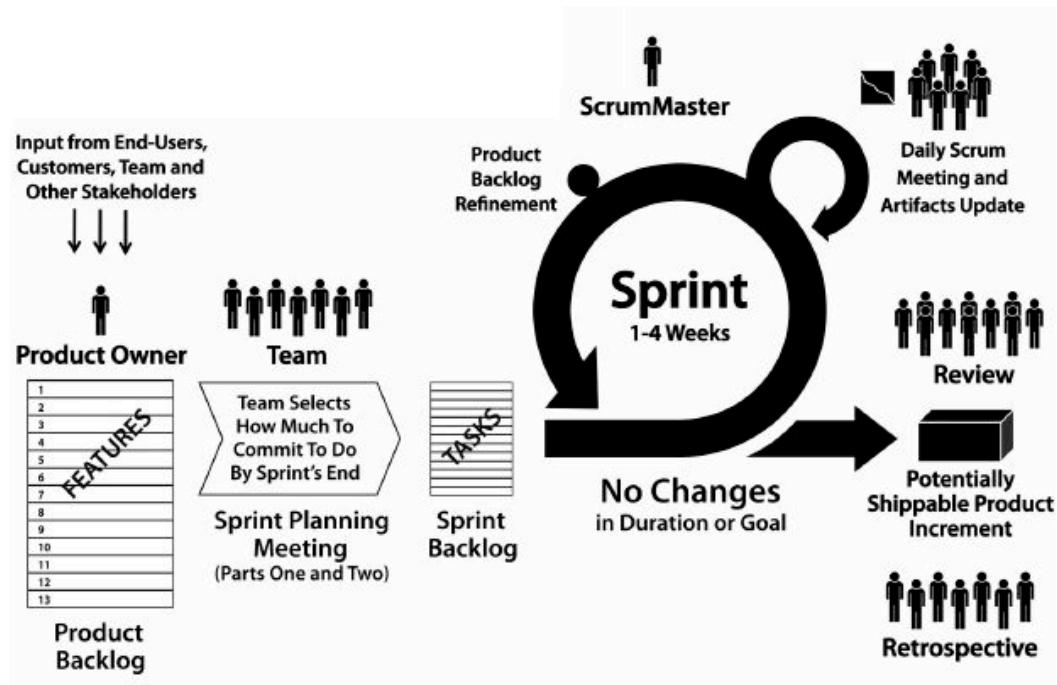


Figure 3. Elements of using Scrum for agile software development (GoodWorkLabs)

### 2.3 LEAN startup methodology

Launching a new business has always been a risky proposition: with the old, writing a business plan, selling it to investors, gathering a team, introducing a product and starting to sell as hard as possible. There are many gaps left between these points where the companies fail. Lean startup methodology makes launching new business less risky. Methodology prefers experimentation over thorough planning, customer feedback over hunch, and iterative design over traditional long design phase before development. Its concepts such as minimum viable product and pivoting have increased their popularity in business world. (Blank 2013)

#### 2.3.1 Continuous integration to continuous delivery

Continuous integration (CI) and continuous delivery (CD) act as accelerators when automating and improving the process of software delivery. Continuous approach makes it easier to find and fix problems. One of the huge benefits is the increased visibility that enables greater communication. Principle of continuous delivery is the ability to deploy software rapidly in production by shortening the cycle time between an idea and feedback. Rapid deployment requires many continuous delivery

techniques to be achieved. These techniques include automating as many stages as possible in the production cycle. (Parsons 2012)

The reference production environment of N4S@JAMK team discloses how continuous integration and continuous delivery are solved in production of the reference product (see Figure 4). The reference production environment is discussed more in detail in chapter 2.4.3 Continuously improving reference production environment COROLLA, and for a bigger picture of reference production environment, see Appendix 1.

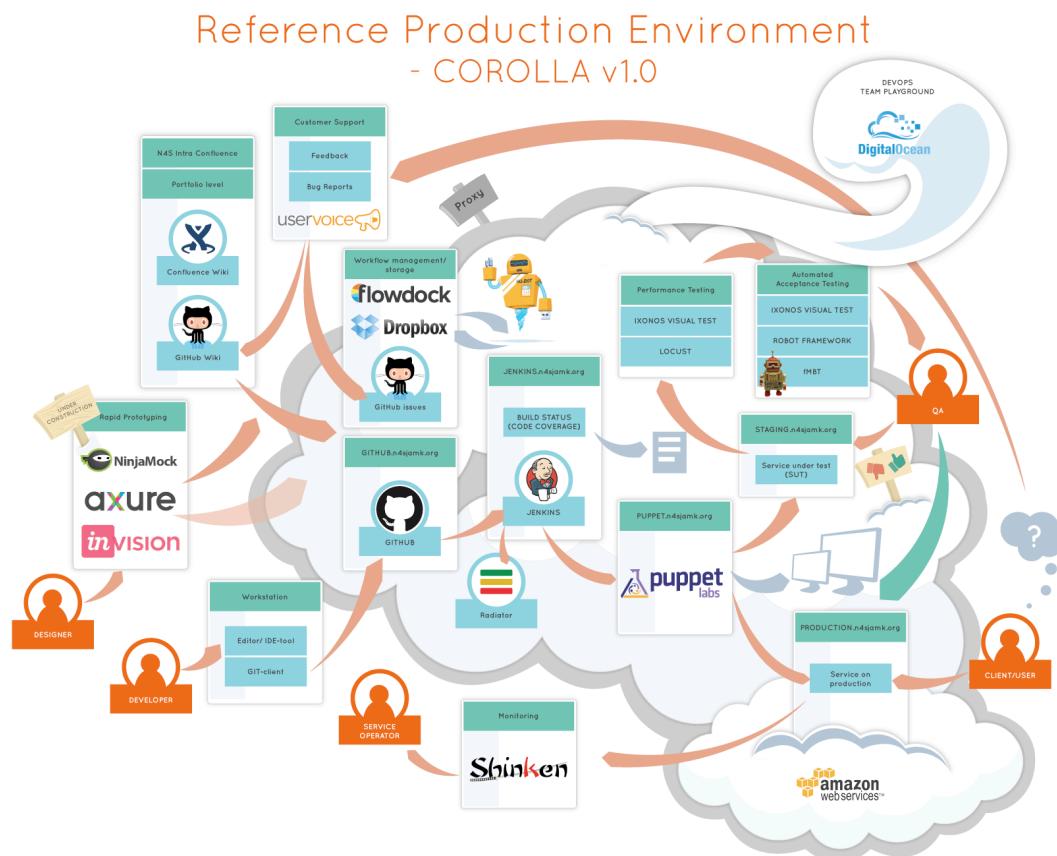


Figure 4. Right side of the environment shows tools used for continuous integration and delivery

### 2.3.2 MVP

MVP (Minimum Viable Product) is known as one of the most important techniques in lean startup. MVP is defined as a version of a new product that allows a team to collect the maximum amount of validated learning about customers with the least

effort. MVP key characteristics are that a product has got a market niche, early users (aka early adopters) benefit of the product and do not stop using it and product provides a feedback loop. (Ries 2009)

Before building MVP it is crucial to think about what it is needed to learn when building, what data is needed to collect and what happens after the experiment, what determines if experiment is a success or a failure. MVP is not about creating minimal product, it is a strategy targeted to avoid building a product that is a waste of time and money. MVP must be accomplished by Minimum Viable Design (MVD). When starting to build a product, it is important to get feedback from early adopters as soon as possible. Early feedback will guide future development and it will show if the product has any value on the market. Collecting ideas and opinions shows to user that collaboration is appreciated. That is a great motivator for increasing usage of product and keeping early adopters using the product. (Ries 2009)

### **2.3.3 Pivot**

When building a product everything does not always go the way it was planned and wanted. One of the hardest facts for a company building a product is to learn to tell the difference between progressing and waste of effort. Developing know how, when it is time to change the paradigm and when to persevere, saves time and money in the end. The practice for the jumping from one vision to the next is called pivot. Adding pivoting to build-measure-learn feedback loop (see Figure 5) maximizes the speed of product building process. (Ries 2009)

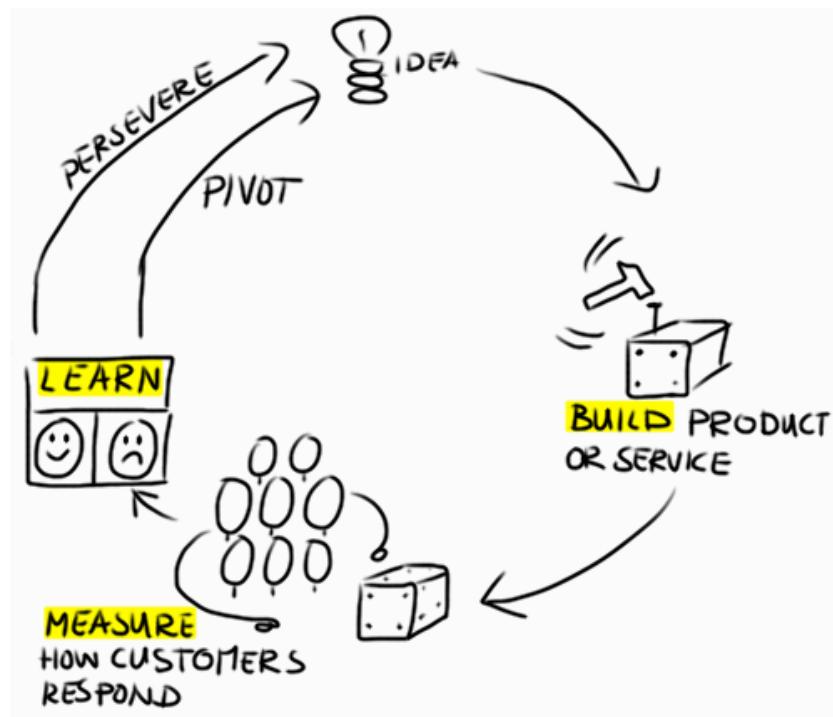


Figure 5. Pivoting when using the build-measure-learn feedback loop (Delgado Garcia 2014)

## 2.4 DevOps as a trend

Definition of DevOps (development and operations) is that it is a software development method including communication, collaboration and integration between software developers and IT operation professionals.

When thinking of rapid development and staying on top of the continuous IT evolution, inevitably comes up terms continuous development and DevOps. Term “DevOps” is popularized by Patrick Debois in a year 2009 at the first DevOpsDays-happening in Belgium. (Hüttermann. 2012, 5.) In traditional organizations there are separate departments for Dev, IT operations and QA (Quality Assurance). Communication and collaboration between previous departments is neither easy nor fast. People are working in silos and silos smother communication. DevOps advocates a set of processes and methods for rapid communication and collaboration through culture, automation, measurements and sharing (see Figure 6).

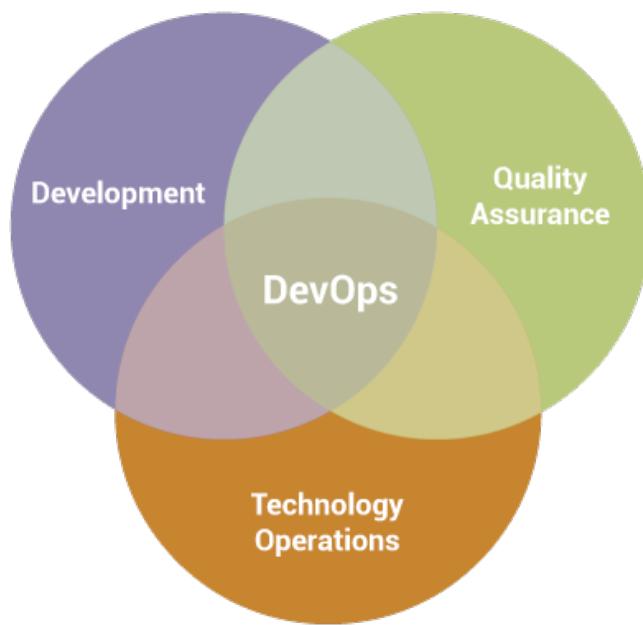


Figure 6. Venn Diagram illustration showing the collaboration enabling agility (Pant 2009, modified.)

#### **2.4.1 Teamwork**

In a book “Building a DevOps Culture” written by Mandi Walls, a Technical Practice Manager of Opscode Inc, is disclosed that DevOps is not just the set of DevOps tools that enables teams to work more rapidly and efficiently. It is more making collaborating environment and changing organizational culture towards more open, professional culture sharing a common vocabulary. Open communication is the key word when team is reaching to DevOps, another important aspect is the respect. All team members should respect other team members and recognize the distribution of everyone. The focus is on the product and team approaches it through its lifecycle, discussing requirements, features, schedules and resources. (Walls 2013.)

#### **2.4.2 Fast releases, fast responses, fast fails**

Speed and stability must be pushed at the same time in order to succeed in the IT field. DevOps advocates behalf of rapid releases. Attempt is to improve the speed and quality of each release and also help team to optimize the whole process. In DevOps cycle (see Figure 7), the fast releasing of product or service offers better opportunities to take risks. Team members cannot become masters without failing sometimes, therefore it is important to build a culture where the continuous

experimentation means taking risks and learning from failure. Also understanding that it is not about who fails least or most, it is about how fast you rise after failing. (Riley 2014.)

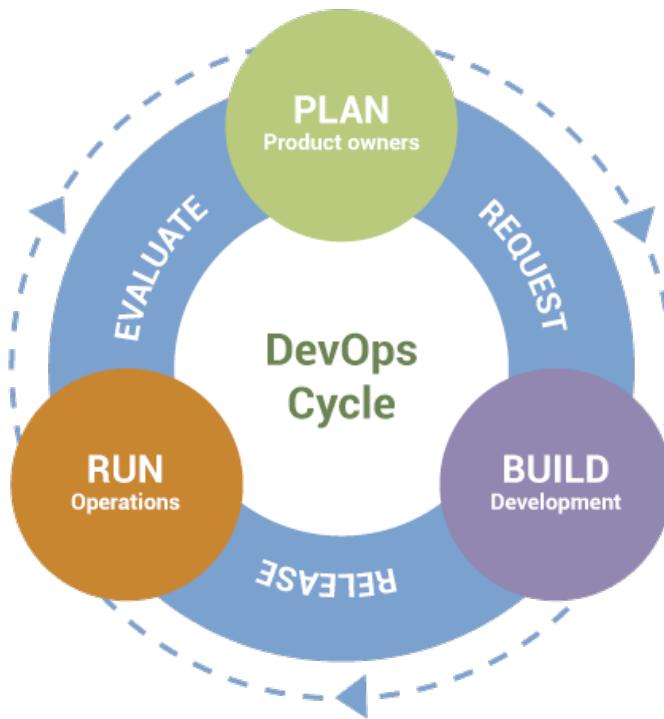


Figure 7. Visual image of the DevOps Cycle (DeVore 2012, modified.)

Driving evolution of DevOps organizations need to promote encouragement of diversity, collaboration and strategic flexibility, inside the organization and outside it.

#### **2.4.3 Continuously improving reference production environment COROLLA**

N4S@JAMK team has developed reference production environment (see Appendix 1). Model changes its appearance while it is continuously improved. Reference production environment shows the tools used in production of Contriboard.

The main idea for reference production environment is to point out how production environment could be built from DevOps point of view. Model is in use in production of Reference product Contriboard. At the same time inside N4S-Program, model acts as a reference showing how continuous deployment and continuous delivery could be implemented in projects that look for speeding up the development. Thesis helps

to grow COROLLA from design point of view. When using right kind of design methods and tools, the design process runs on its full speed. After tools are chosen for designing Contriboard, tools can be attached to reference production environment and again, model is continuously improved.

## 2.5 Service design

Service design is a part of the development process and about improving quality of the service and the interaction between service provider and customers. The purpose of service design methodologies is to design according to the needs of customers or participants, so that the service is user-friendly, competitive and relevant to the customers. It is creative work done with customers and organization and includes concrete user testing. At the beginning of service development process, proficiently made service design has got a positive and accelerating impact in the end. Service design is designing with people rather than designing for them. (About Service Design)

### 2.5.1 Deep customer insight

When developing a new product it is crucial to understand the customer. The product has to be user centered to be desirable, cost-effective to be economical and technically implementational. Getting testing data from users has to be explicit and observable. There are different kinds of ways to get the data. Traditional ways are the interviews, inquiries and research. Design ways are hatching and observing by making Service safaris, service probes or context maps. There are many kinds of service design tools, which can be used during service design process.

### 2.5.2 Service design thinking

Marc Stickdorn, a trainer and consultant for service design lays out principles for Service design thinking. Service design thinking has to be **user centered**. Being user centered requires going way beyond static data. It means that the service should be experienced through the eyes of customer. Stickdorn defines what could happen if only the static data were to be examined:

*Think of 2 customers. Both were born in 1948, male, raised in England, married, successful and wealthy. Furthermore, both of them have at least 2 children, like dogs and love the Alps. One of them is Prince Charles and the other is Ozzie Osbourne. (Stickdorn 2010, 36.)*

Service design thinking needs to be **co-creative**. It means exploring needs and new processes in collaboration with all stakeholders and not forgetting the most important stakeholder there is, the customer. Co-creation is working with the customer, creating better and more customer centric processes. **Sequencing** focuses on the service flow and how it impacts customers' mood. When thinking of user interfaces, if some common feature is too hard to find it makes user feel frustrated. Or if matters happen in low-speed, user gets bored or mad. **Evidencing** means making customer aware of intangible or sometimes "behind the scenes" services. It is like chocolates on the pillow in a hotel telling customer that cleaner has been there. **Holistic** thinking focuses on keeping customers' mood and feelings in mind through the whole customer journey. Everything matters in every touchpoint even the way how customers' senses react to the physical environment. (Stickdorn 2010, 34-44.)

### **2.5.3 Service design tool: Customer journey mapping**

Customer journey covers the entire service customer experiences from beginning to end and it is also called **end to end experiences**. The journey consists of a sequence of **touchpoints**. It includes pre-service period (expectations), service period (experiences), and post-service period (satisfaction). Touchpoints form a loop, from first contact to repeat purchase. Each touchpoint present an opportunity to strengthen the relationship with customer. Contriboard has two different user groups, power users and guest users. Contribboards user journey image below (see Figure 8) illustrates how user groups differentiate from each other, and what the main touchpoints are that user experiences on the service period.

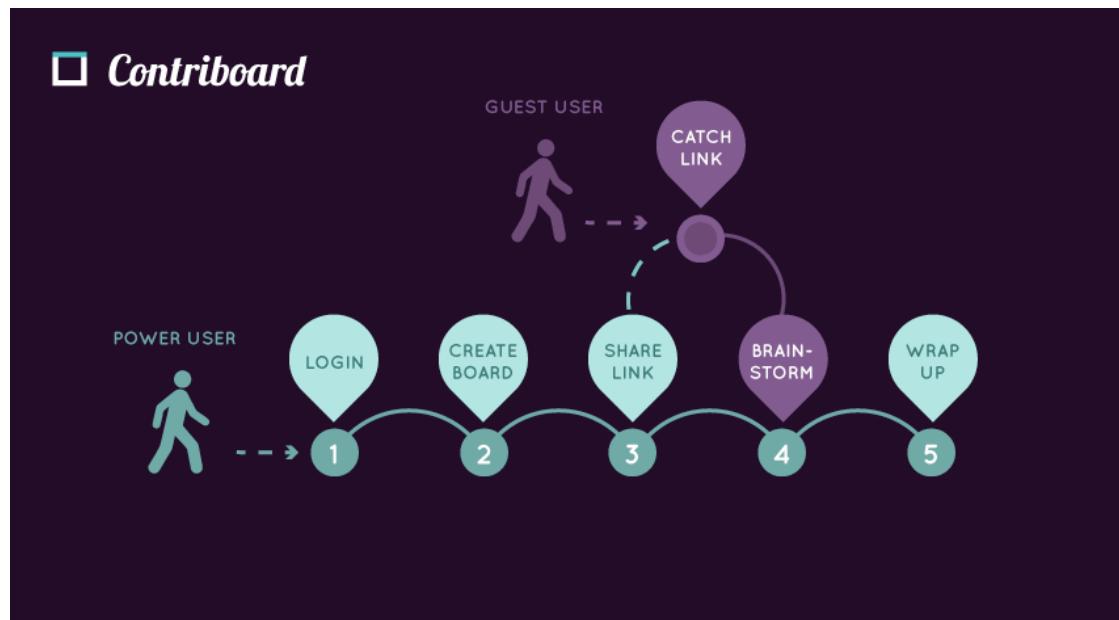


Figure 8. Main touchpoints of Contriboard

To show customer service there needs to be a customer. Just thinking experiences from for example “guest user” point of view is not enough. A persona needs to be created or thought. Good persona created is a stereotype of a customer segment. It is also crucial to understand different personas. Every persona perceives touchpoints differently. Marc Stickdorn and Jakob Schneider have made a tool called the Customer Journey Canvas (see Appendix 2). Customer Journey Canvas is a helpful tool when improving end to end experiences.

#### 2.5.4 Customer experience and social media marketing

Social media drives companies to shift marketing from advertisements to experiences. This is where service design is needed. If a company has done service design, it is most possible that customer feels great after using service and probably tells about it in social media. Sometimes positive experiences get publicity and usually negative experiences most definitely spread all over the world.

A one story of positive social media experience is the story of Joshie the giraffe in hotel: The family spent few days in a hotel in Florida. Upon returning they discovered that their son’s beloved stuffed giraffe Joshie had been forgotten to the hotel. The child was so upset of the idea of going to sleep without Joshie that the father told a

story. He told that Joshie is just taking longer vacation at the hotel. Next day father called the hotel and Joshie was found. The father asked if the hotel personnel could take a photo of Joshie by the pool so that his story could be substantiated. After a couple of days family received a package from the hotel. The package included Joshie and some Hotel branded goodies. It also contained a binder which included documentation of Joshie's extended stay at the hotel: Joshie wearing shades by the pool, at the spa, Joshie making friends, driving a golf cart on the beach etc. The hotel handled the customer experience so well that the family told it in social media and the hotel got massive visibility. This proves that it all comes down to customer satisfaction. (Hurn 2012.)

### **2.5.5 Changing of the service ecosystem**

The early focus in marketing was on products. Goods-dominant logic has value in exchange. It means producing a product and selling it, the customer service period is short. Later the line between goods and services started to fade. It should be asked from consumers if it even matters whether it is about services or products. Probably the answer would be no. If we use it, we pay for it. It is called service-dominant logic, the value is in use. The concept of new dominant logic for marketing was brought to life in 2004 by two marketing professors, Vargo and Lusch. New dominant logic has a major impact for a business model and it means value co-creation, customer produces value with the provider. (Lusch & Vargo 2004.)

Thinking of mobile phones; when there were no smartphones, people were buying phones that needed to last years and the batteries did not need to be charged every day. Previously the mobile phones were for calling and texting purposes. Today when smartphone usage is growing fast, people are changing the paradigm why to buy a certain kind of smartphone. People are doing more than just calling and messaging with their smartphones. The whole service ecosystem has changed. Designing value in use, everything starts with understanding the ecosystem. The value in use designing needs to be based on the customer journey, multidisciplinary team and on a guiding vision. (Roscam Abbing 2010.)

Even the Nokia's former CEO Stephen Elop tried to explain why Nokia was in such a trouble. This is what Elop wrote about ecosystems and Nokia:

*The battle of devices has now become a war of ecosystems, where ecosystems include not only the hardware and software of the device, but developers, applications, e-commerce, advertising, search, social applications, location-based services, unified communications and many other things. Our competitors aren't taking our market share with devices; they are taking our market share with an entire ecosystem. This means we're going to have to decide how we either build, catalyse or join an ecosystem. (Engadget 2011.)*

## 2.6 Wireframes, prototypes and mockups

Wireframes, prototypes and mockups are forms of representations of the graphical user interface (GUI). Internet is full of documents telling what these representations mean and what differences there are. The confusing part is that many documents are stating different things. Other assumption is that after wireframing comes mockuping and then prototyping. It may be in that order sometimes but as learning how to design MVP as LEAN ways as possible means using MVD. LEAN designing turns mockups and prototypes around. Medium to high-fidelity mockups should be done once feasibility is tested. It is waste of time to make high-fidelity mockups if all testing happens afterwards and the design changes. The following chapters are simplifying how these wireframe, prototype and mockup can be understood in this thesis work.

### 2.6.1 Wireframe

Wireframes are simply the backbone of the design. Wireframe is a low-fidelity representation. It shows the main content, structure of information and simple description and basic visualization of the UI interaction (see Figure 9). Wireframing is a great support in ideation process. Most used tools for wireframing are pen and paper; however, nowadays everything is more useful in digital form. This is why usually the best practice is to first sketch fast and then use a UI design tool to simple wireframing. Some of the UI design tools simplify the sketching to wireframing phase by making it easy to use a sketch as a part of the wireframes. Many of the UI design

tools make it almost too easy to wireframe so that the sketching phase can be left undone.

It may seem that wireframes look slightly unimpressive for the customer. Still there are many pros as well. Customer is not distracted with colors and it is easy and fast to make changes when wireframes are just simple drawings or made fast using UI design tools.

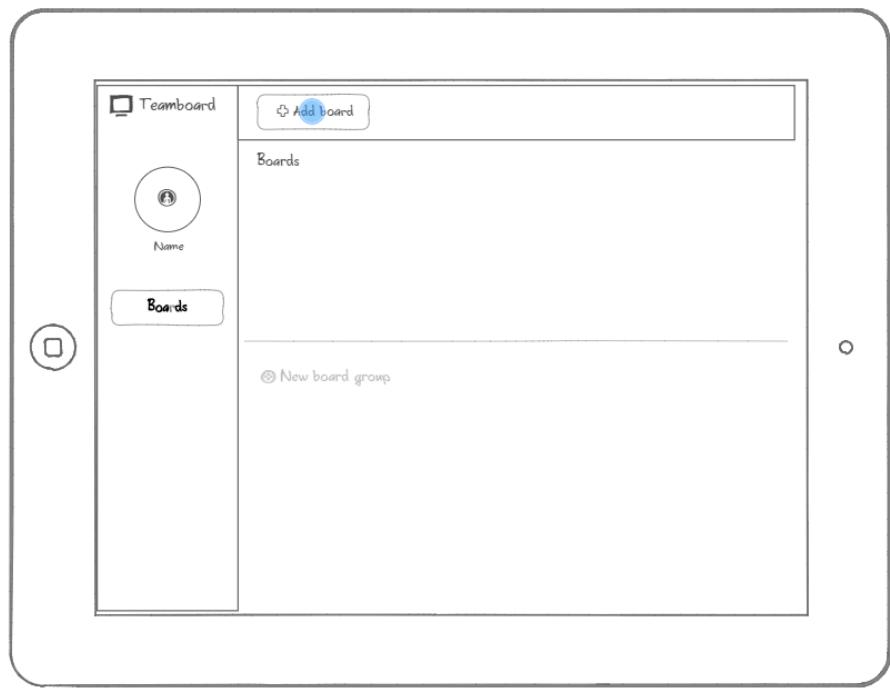


Figure 9. Wireframe made using NinjaMock UI design tool

### **2.6.2 Prototype**

Low-fidelity prototype has specific data and it looks like wireframe but with high-fidelity interactions. High-fidelity prototype looks and feels like the final UI with all interactions. Click through prototypes are good for testing purposes. Stakeholders can easily become distracted to think that the high-fidelity prototype is the final product. High-fidelity prototype is great for testing purposes, especially UX testing.

### **2.6.3 Mockup**

Mockup is a realistic representation of what the product will look like. For example, UI design tools as they are usually called mockup tools can be used for making

wireframes and Adobe Illustrator or Photoshop can be used for making mockups.

Mockup is a specific graphic design of UI. Low-fidelity mockup is like wireframe with some colors, gradients, icons and specific data. High-fidelity mockup looks like a final product print. The great feature in high-fidelity mockup is that it looks realistic and developer can use it for picking colors and icons etc. Some of the UI design tools support CSS and HTML export, which means less work for developer. A huge disadvantage is that mockup requires many hours of work, and making changes is difficult and slow. High-fidelity mockups are great for marketing purposes. UI mockup can be submerged inside a photograph of a platform and used in print media for example (see Figure 10).



Figure 10. Mockup merged inside photo of a platform

### **3 Process and methods supporting N4S-Program**

#### **3.1 Process cycle**

Process used in developing product defines the area to focus on. The thesis focuses on the design phase. The first design phase takes place at the beginning of product

development, before any code is implemented. Innovation process image (see Figure 11) shows what occurs in the design phase. Learning point is important in every process as earlier stated when defining LEAN startup methodology and build-measure-learn feedback loop.

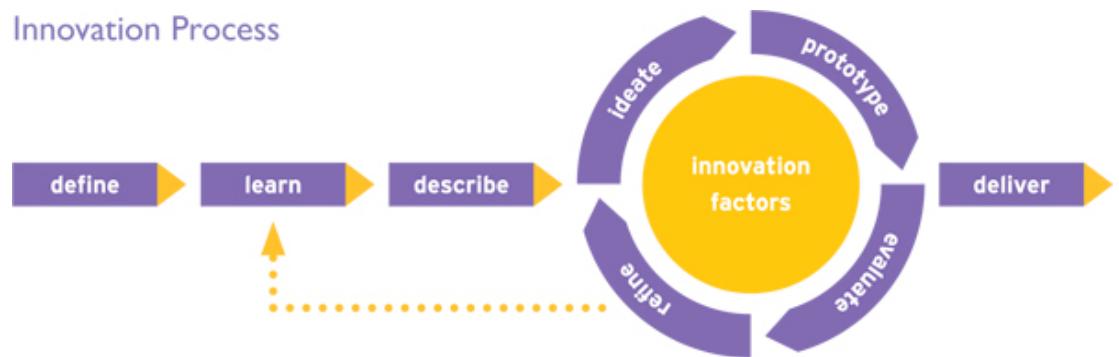


Figure 11.Innovation process (Shubert 2010)

By compounding build-measure-learn feedback loop, innovation process, LEAN Startup process-diagram, software development cycle and rapid design process the result is fast and DevOps-like production cycle (see Figure 12). The figure below shows how old silo-way of working changes in to a more collaborative process. The thesis emphasizes mostly rapid design on the left, without forgetting the issues on the right.

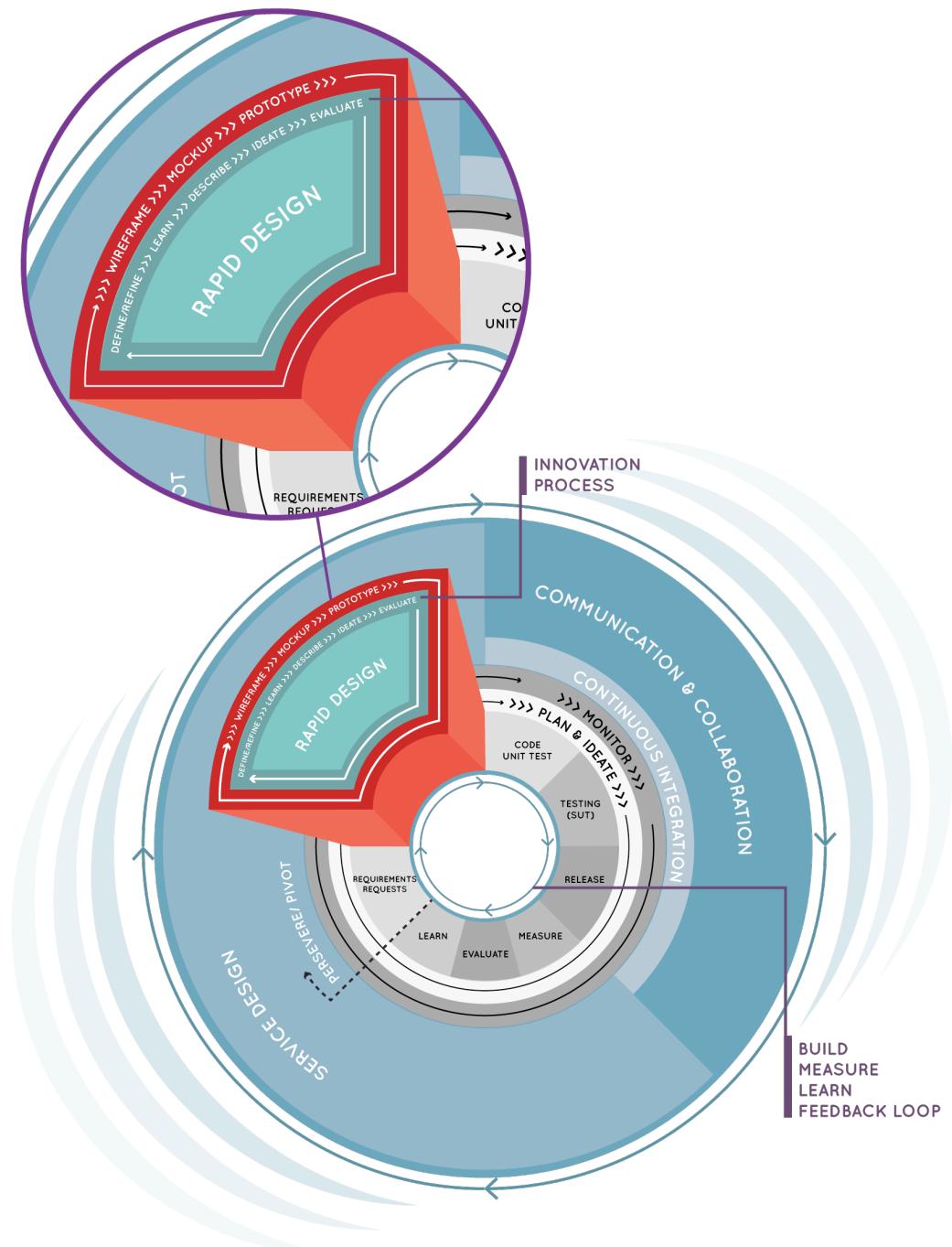


Figure 12. Compounded Agile DevOps-like process cycle

### 3.2 Using UI design tools as a part of design process

The design cycle of product or a service starts with a simple idea. After the idea comes the identifying of customer needs. UI designing starts with rough sketches and after that come wireframes, prototyping and mockups.

There are many issues to take into account when showing designs to stakeholders. Depending on the stakeholder being and IT-professional or completely unaware of

the qualifications of a good user interface designs, it is important to focus on user experience in the early stage of the designing process. Where the stylish visualization image might be enough to one stakeholder the other might demand interactivity to see what happens when pushing the buttons. The worst case scenario is that the nice looking design will get through production process. When user tries the product and usability is bad, user does not want to use it anymore. Bad experiences spread to other users and the product becomes a flop. It is not the best practice to show only the visualizations of designs to stakeholders; the designer needs to show what happens in design. Using UI design tools could be the answer to diminish the gap between designers and developers as well as with the whole company. When all the stakeholders are working closely together, the team makes sure that the product will be successful and satisfies everyone and redundant waste is not made.

A problem faced when starting to build product is that designers cannot evaluate the design properly until it is built and building a product takes too much time. Designer needs feedback and ideas of designs from developing team regarding usability and functionality of a certain feature of a product before it is built. Usually designer has few ideas how a feature could be implemented and needs more perspectives. With the right kind of UI design tool, designer can make clickable wireframes fast, share examples and collect real-time feedback from team and at the same time support the rapid development of a product and LEAN UX. Using the right kind of UI design tools could be the key to simulate the designs. Simulating designs could bring out the usability problems in early stage and that would rapidly lower costs and save time in the end. Delivering results faster without forgetting user experience serves the purpose of WP1 in N4S-Program.

## **4 UI design tool testing requirements and implementation**

### **4.1 Requirements for evaluation**

The variety of UI design tools is wide. Online tools are inevitably more portable than standalone tools. This is why testing concentrates on web based UI design tools and one of the criteria needed is machine independence. There are still many great

standalone UI design tools need to be mentioned by name like AxureRP and Balsamiq. A high amount of tools obliges to test the ones with a free trial period. Most tools offer 30 days trial and some of the tools have a free pricing plan with limited features.

## **4.2 Comparison criteria**

After understanding the right kind of design process, the best working habits serving the DevOps make UI design tools selecting and testing simpler. When knowing what to design and how, it is easier to decide the main functionalities what to look for the mockup tool.

### **Workability, designing, usability and formability**

- usability, fast to learn/ fast to use
- work online/ machine independent
- different layout alternatives (desktop, tablet, mobile)
- hot spots/ linking
- library of components (buttons, icons etc.)
- file upload
- export options (PDF)

### **Testability and collaboration**

- collaboration possibility (adding users for testing, collecting feedback)
- Link share (fast testing)

### **“Nice to have” features**

- button interactions
- reasonable price
- huge community

## **4.3 Cost structure**

The prices for the tools are categorized in four groups (A, B, C, D) by their pricing methods. The groups are introduced below.

The groups for UI design tool prices:

- Price category A, license quantity based pricing.
- Price category B, license and project quantity based pricing.
- Price category C, reviewer and storage quantity based pricing.
- Price category D, strongly on features based pricing.

#### **4.4 Test implementation**

To get solid answers to the research questions and more extensive test-results, the thesis includes a tripartite testing phase. The testing phase includes:

1. Status quo and survey.
2. Deeper testing of features and functionalities.
3. Testing UI design tool in design process of Contriboard.

Mapping out status quo of web-based UI design tools is executed by finding tools from the internet by using search engines. To get a stable basis for testing, some blogs and forums were found with comparisons of UI design tools. The testing phase was implemented in collaboration with three N4S@JAMK interns in the summer of 2014 to give more perspective and volume to the testing of UI design tools. Each UI design tool was tested once, except for one tool. The tool InVision App was tested twice, while noticed before getting to next testing phase that the tool was significantly developed and had many new features. The information on testing was collected in a form of a web-survey. The survey was executed with Google Forms by the author of the thesis and it consisted of questions (see Appendix 3) supporting comparison criteria listed earlier in the thesis. Test results lead to choosing tools with the most comparison criteria required. The most suitable tools found continued to a deeper testing phase.

The deeper testing phase is for testing features required in rapid designing and understanding the functionalities of the tools. Deeper understanding helps to select the most suitable tools for the design process of Contriboard.

The third phase was about using the most suitable tools for design process of Contriboard. Using UI design tools in real development process, gave more valid results answering research questions.

## 5 Research and evaluation of UI design tools

### 5.1 The status quo and survey

Mapping out the status quo of web based UI design tools is a process driven by intuition, word of mouth, and people using UI design tools and spreading information on those on the web. Blog writings found are a year or two years older, therefore it must be remembered that the results found in comparisons can be expired. Most help found regarding to blogs and posts was from Smashing Magazine articles when using words: Prototyping, Web Design, Wireframing, interaction design, process and productivity. The tools that were found to be most interesting to test were Concept.ly, InVision App, FluidUI, NinjaMock, Proto.io, ProtoShare, UXPin, Easel, Justinmind, and Moqups. After choosing the list of UI design tools to be tested, the list was given to interns. The interns tested the tools and answered to the survey questions. The survey answers (see Appendix 4.) helped to map out which tools are most suitable for closer testing.

Based on the survey grading (see Table 1) the most versatile tool is Proto.io. The tool is felt to be very easy to use and many tutorials as well as help can be found easily if any problems appear. Library elements can be modified widely; therefore detailed work is possible. The tool presents to be great for high-fidelity prototyping purposes. InVision App was tested twice for two different purposes. The grading points of InVision differ from each other. The reason for differing points can be in what purpose the tool has been tested for. InVision App is more for using ready-made mockups and making them clickable. The timespan of InVision App tests being almost five months can explain the increase of support points. The tutorial amount increases while the community increases.

Table 1. Summary of UI design tool grading points from inquiry

UI design tool	Ease of use	Support	Formability of elements	Grade	Total
Concept.ly	3	3	1	2	9
Easel	2	5	5	4	16
FluidUI	4	2	3	3	12
InVision App	5	3	1	2	11
InVision App	5	5	1	5	16
Justinmind	3	5	5	3	16
Moqups	5	5	3	3	16
NinjaMock	5	4	4	4	17
Proto.io	5	5	5	5	20
ProtoShare	3	5	5	4	17
UXPin	2	5	5	5	17

Ease of use: 1= Very hard, 5= Very easy

Support: 1= No help found, 5= Lot of tutorials and help

Formability of elements: 1= No modifications, 5= Modifications can be made widely

Grade: 1= Bad. Would not use it anymore, 5= Great! Would recommend to others

Closer testing covers seven UI design tools of status quo. These tools are Concept.ly, FluidUI, InVision App, NinjaMock, Proto.io, ProtoShare, and UXPin. Concept.ly received weakest points, still it is needed to be tested more clearly to have comparing pair for InVision that have same kind of functionalities. The tools left out of evaluation were Easel, Justinmind Prototyper, and Moqups. Easel and Justinmind Prototyper felt to be hard and slow to learn to use. Later on the developers of Easel decided to shut it down. The main reason to dropping Moqups out of the scope of evaluation was the comment feature that could not be tested.

## 5.2 Deeper testing of features and functionalities

Evaluation of tools demand clarifying some of the words used. The projects inside the tools are groups of wireframes, prototypes or mockups. Reviewers mean users participating only in the discussions by adding comments into projects. Reviewers cannot create or edit projects. Web storage contains wireframes, prototypes, assets and discussions, everything made in projects.

The following chapters cover a variety of UI design tools of status quo, focusing on the features they have to support the needs of designers.

### 5.2.1 Concept.ly

#### Workability, designing, usability, formability

Concept.ly is an online tool for making mockups clickable. There are no ready-made components. Workflow is made simple by deciding layout alternative (desktop, tablet, mobile), uploading static pictures (mockups), linking mockups by making hotspots, and sharing to others. Dropbox integration makes workflow rolling faster. Once mockups are brought to Concept.ly from Dropbox, linking is easier and faster. When modified mockup is reloaded to dropbox, the change can be seen at Concept.ly. Project view is showing a minimap of the project's first mockup. After a new mockup is reloaded to Dropbox, old mockup stays in the minimap. When the view is clicked it shows the new mockup, a small usability problem there. The project has PDF-export including feedback.

The main view is somewhat confusing at first (see Figure 13) because of the test project showing all the test screens. Something that might help user to get less confused could be to make another view for administering projects. User could visually see all the projects at one time. Deleting views from the project is time consuming. The delete button is hidden and it took time to find it.

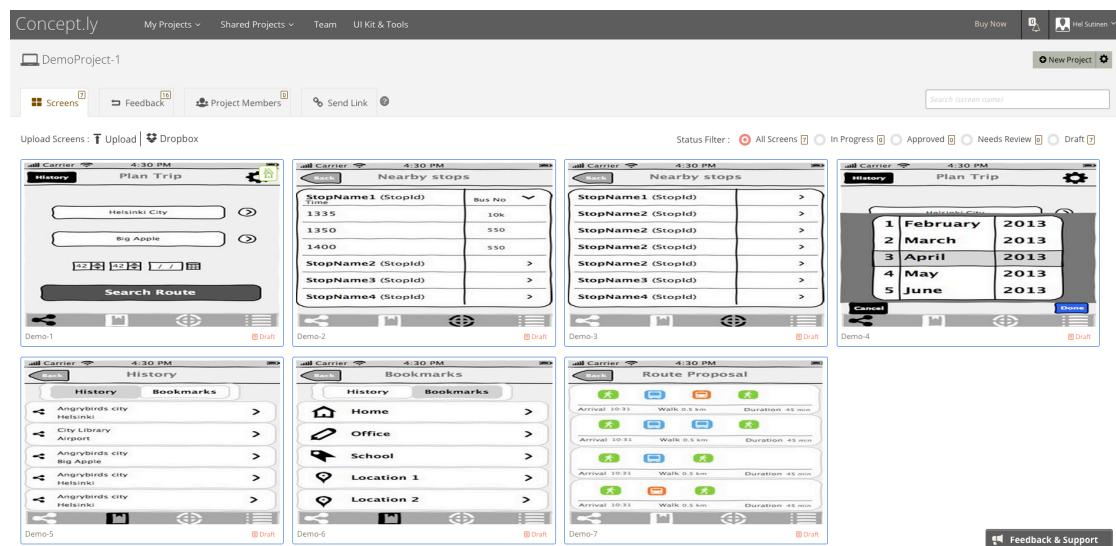


Figure 13. Main view of Concept.ly

### **Testability and collaboration**

Collaboration is made precise, easy and fast. Clickable mockup can be shared by URL and through social media. There are multiple types of feedback. Feedback can be pointed anywhere on the screen by marking the area and providing comments. Feedback can be marked as a comment, suggestion, problem or idea. Feedback activity can be monitored easily and the tool tells when feedback is provided. Instant email notifications are provided.

### **“Nice to have” features**

Concept.ly is not made for piling clickable wireframes from the scratch so a library of components and button interactions is missing entirely and is not even relevant. Concept.ly pricing goes to price category B. Concept.ly pricing is simple and reasonable if this kind of UI tool is needed (see Table 2). Concept.ly is providing support and response time is promised to be 2-4 business hours. UserVoice feedback tool is used as a feedback channel and for collecting ideas from community. New releases are collected to Concept.ly Wordpress blog and websites include multiple videos of features.

Table 2. Plans and pricing for Concept.ly

Plan name	Price \$/ mo.	Licences	Projects	Reviewers	Other
Free	0	1	2	Unlimited	
Premium	9.99	Unlimited	Unlimited	Unlimited	

### **5.2.2 Fluid UI**

#### **Workability, designing, usability, formability**

Fluid UI is a fast to learn and fast to use online tool. Simplicity of the UI makes the workflow intuitive. There are different layout alternatives and in the paid version, alternatives can be specified to different platforms. The workflow is smooth. It starts with deciding on what kind of platform to design and how the interactions take place (slide left, slide right etc.), building mockups of library icons and buttons etc. (see

Figure 14), linking pages together and trying out. Paid version offers wider arrange of library components. Components are categorized in different devices, when making mockups for Android for example, Android icons and other components are in use.

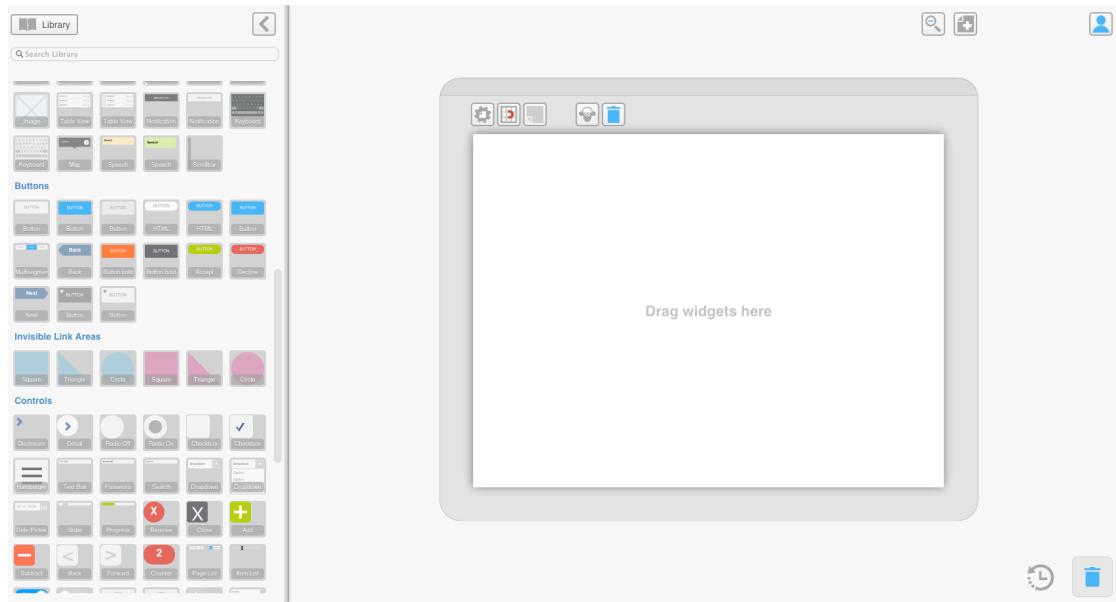


Figure 14. Simple user interface, library icons on left

The formability of elements is experienced to be working nicely. Buttons can be fixed to any sizes and colors. Testers felt that Fluid UI lacks social buttons. Testing shows that users do not feel possibility to make high-fidelity prototypes. Tool is experienced to lack features of modifying possibilities. Free version of Fluid UI does not offer possibility to bring pictures in mockups. Paid version offers multiple exports. Screens, interactive mockup (HTML) and screenflow diagram can be exported. Screenflow diagram shows screens and linking between screens. Linking is not shown precisely, as like where to go when pushing some button, just between screens (see Figure 15). If there is plenty of linking, screenflow diagram can be hard to read and it can be misunderstood. In some of the testing cases Fluid UI stops working. It freezes and user must sign out and in again to get it work. Sometimes mockup must be run few times before changes made are showed in URL and sometimes Fluid UI lost few images.

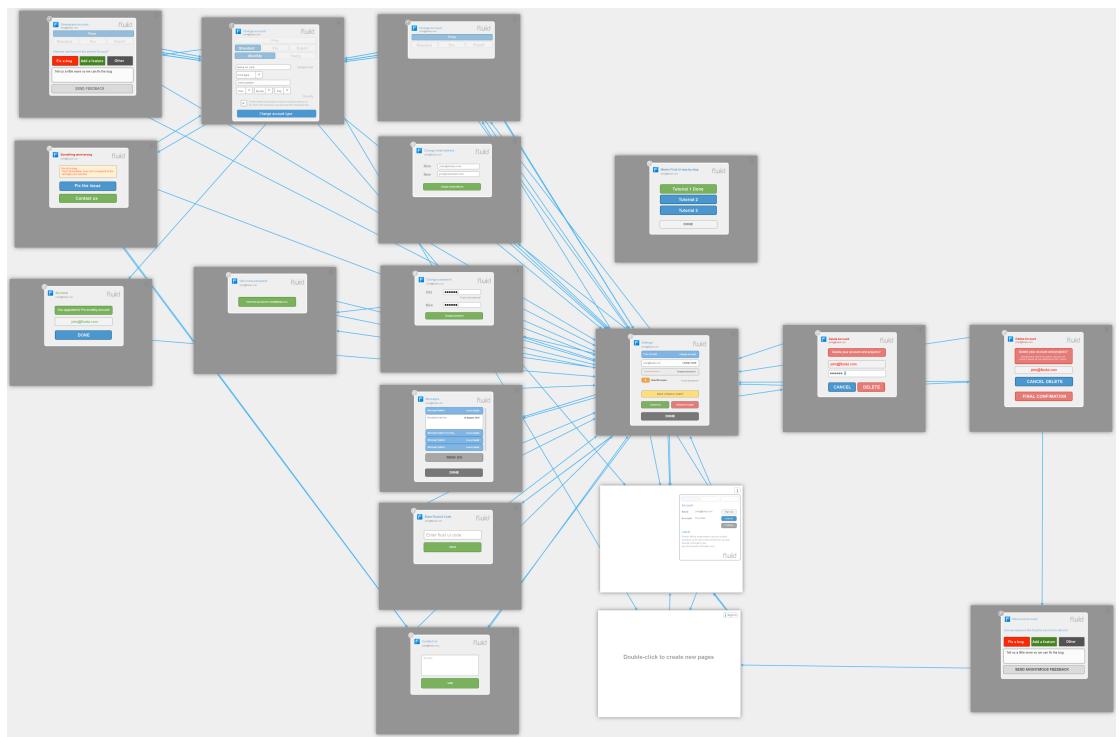


Figure 15. Screen diagram of Fluid UI demo (Fluid UI)

### Testability and collaboration

Link or QR-code to clickable mockup can be shared to collaborators and it can be tested on devices. QR-code is a fast way to test mockup on device. The testing mockup behind link gives a possibility to show notes; still there is no button to make notes. Giving feedback on mockup feels impossible.

### “Nice to have” features

Fluid UI has the ability for linking interactions that can be used for objects. Buttons do not change color while hovering; however, touch gestures (tap, long press, swipes etc.) and screen transitions (slide, fade etc.) can be added. Community is growing fast and Fluid UI website is offering help and tips for users. Fluid UI claims to be in top 3 places for range and depth of functionality offered and as for the terms of pricing the 3rd cheapest. Still it must be remembered that if examining through project perspective, with 12 dollars per month user can only have one active project. There are 3 plans posted on Fluid UI websites (see Table 3). Fluid UI promises tailored plans upon request and offers a tool in use with -50% student license. University teachers teaching UX are able to get heavily discounted offers.

Table 3. Plans and pricing for Fluid UI

Plan name	Price \$/ mo.	Licences	Projects	Reviewers	Other
Lite	12	1	1	Unlimited	
Standard	29	1	10	Unlimited	
Pro	49	1	30	Unlimited	

### 5.2.3 InVision App

#### Workability, designing, usability, formability

InVision App is marketing itself as an online prototype, collaboration and workflow platform. UI looks well designed and gives a good feeling. Unlike Concept.ly, InVision App has a project view showing all projects user has (see Figure 16). InVision App workflow is almost similar to Concept.ly, including Dropbox sync. The only difference is the projects view. InVision offers different layout alternatives like desktop and Apple or Android tablet and mobile. User is able to make interactions to mockup to make it more of a prototype. Mockup pictures can be transformed into active prototypes adding clicks, hovers, gestures for transitions, and animations for interactivity. A prototype can be viewed in browser or on mobile. Exporting to PDF is nicely implemented. All the comments are included with commenters. UI is usable and works well without any errors.

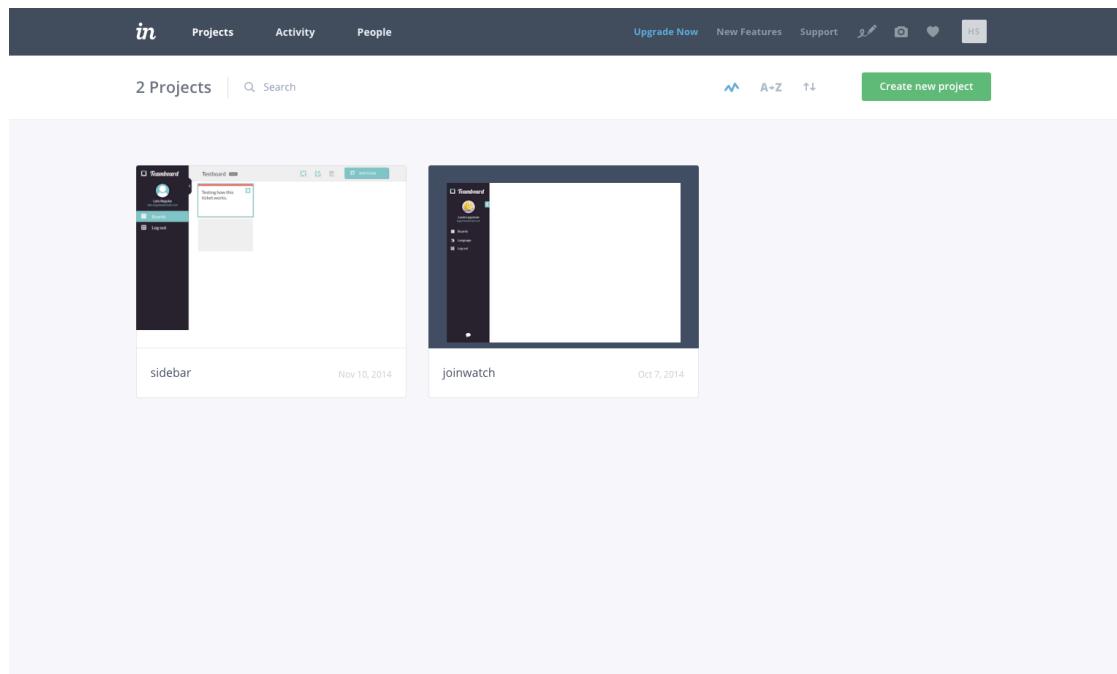


Figure 16. Projects view

### Testability and collaboration

Collaboration is made easy through a shared link. Giving feedback is easy after collaborator understands to click comment mode on. Comments can be clicked anywhere needed and another great feature is that collaborator can include drawing inside comment (see Figure 17). InVision offers comment history and version history.

InVision has live share mode and live share includes a link to a whiteboard with possibilities to draw and write. Live share can be a nice addition to have when including a brainstorming session to the design process.

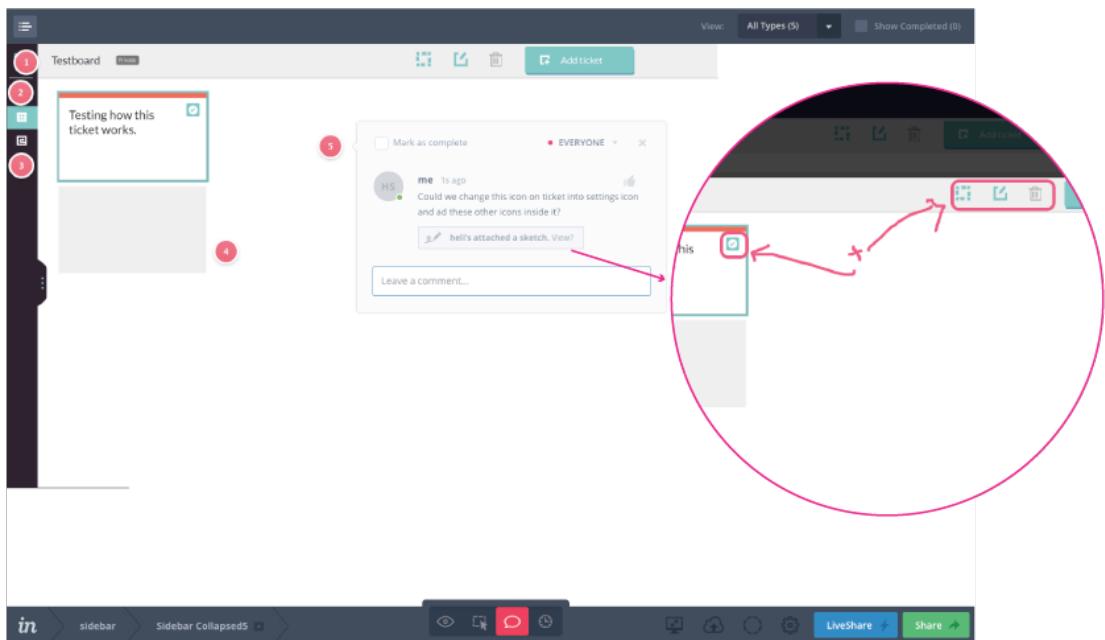


Figure 17. Comment and drawing inside comment is added

#### **“Nice to have” features**

In marketing InVision emphasizes customers who are using the product. InVision has got a FAQ site including tutorials and a contact form. InVision App fits to price category B. There are five different plans for pricing. Getting the most effort of this UI tool, the plan should be Professional (see Table 4), which is slightly too expensive price for this project.

Table 4. Plans and pricing for InVision App

Plan name	Price \$/ mo.	Licences	Projects	Reviewers	Other
Free	0	1	1	-	
Starter	15	1	3	-	
Professional	25	1	Unlimited	-	
Team	100	5	Unlimited	-	
Enterprise	-	-	Unlimited	-	Advanced features

### 5.2.4 NinjaMock

#### Workability, designing, usability, formability

Starting to use NinjaMock “My projects” view is slightly confusing. If user has many projects, this kind of view would need to have a preview image to show what is inside the project. User gets confused what is the difference between projects and files when it looks like projects created are not inside files (see Figure 18.). Project management lacks good usability.

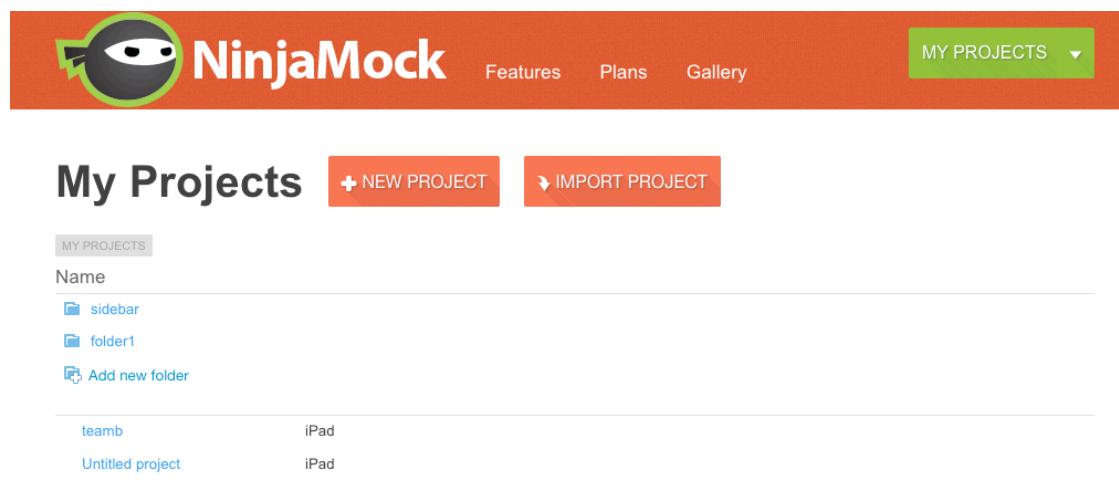


Figure 18. Project management in NinjaMock

NinjaMock is great for making fast wireframes with hot spots. Learning to use NinjaMock does not take too much time. All the library components look hand drawn and are made for wireframe purposes. NinjaMock has one specific feature that lacks from other UI design tools, vector editor. NinjaMock has a pen tool and a path edit tool. It is easy to draw any form of elements. Export is possible to PDF, and it can include wireframes, comments and links. UI looks slightly unprofessional, it might be because of the hand drawn outlooks of elements (see Figure 19).

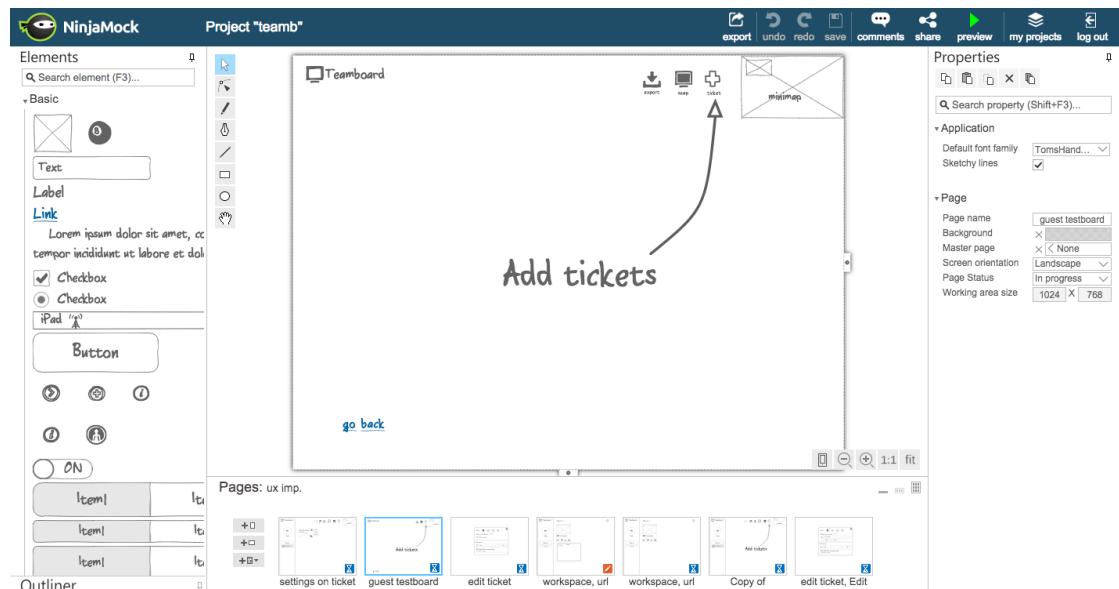


Figure 19. Outlook of NinjaMock UI in wireframing phase

## Testability and collaboration

Collaboration takes place in preview mode through a shared link. Before sharing a link, user can refine which view is a starting view in the link. When changing the starting view, sometimes it does not change in link. Commenting commences by first writing a comment, after writing and clicking comment-button a speech bubble appears on screen and can be dragged wherever desired on wireframe. Comments use color coding: unresolved comment is blue and resolved grey (see Figure 20).

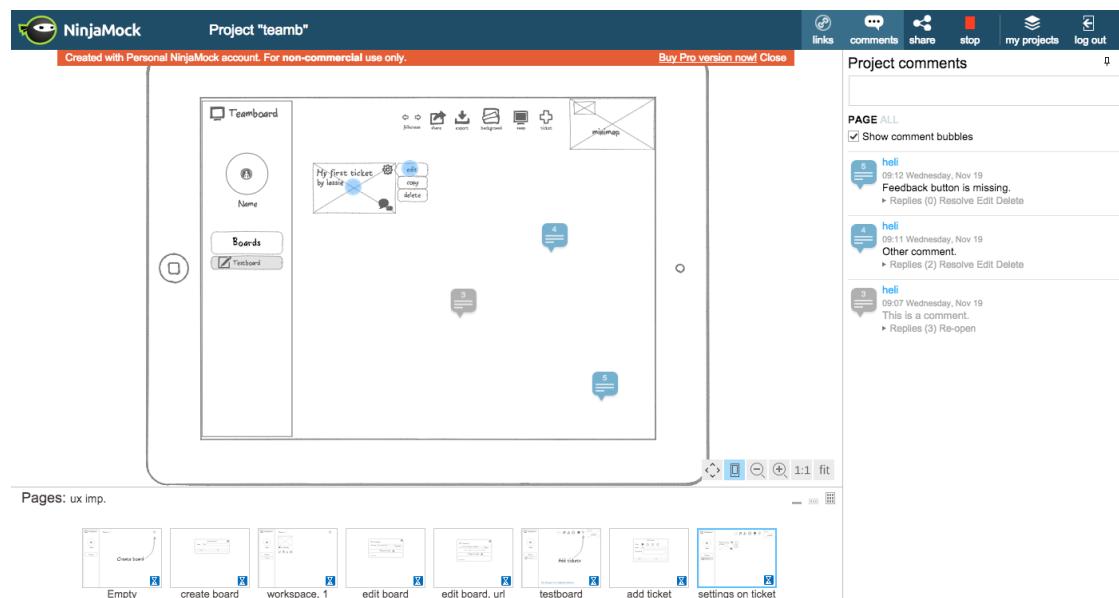


Figure 20. Project preview and comments

### **“Nice to have” features**

Button interactions are not possible to make. In this tool when it is all about wireframes, interactions are not even necessary. The community is growing all the time. Some of YouTube videos can be found and all of the features are precisely introduced on web pages. NinjaMock collects Feedback and ideas through UserVoice feedback channel. NinjaMock is in price category D. It has different plans for non-commercial and for commercial use (see Table 5).

Table 5. Plans and pricing for NinjaMock

Plan name	Price \$/ licence/ mo.	Licences	Projects	Reviewers	Other
Personal	0	1	3	Unlimited	Non-commercial use
Pro	10	1	Unlimited	Unlimited	
ProPlus	25	1	Unlimited	Unlimited	Shared workspace for multiple users

### **5.2.5 Proto.io**

#### **Workability, designing, usability, formability**

Proto.io is online UI tool for making wireframes and high-fidelity prototypes. Formability of elements is experienced to be good in testing. Tool lets user to do so much that it is felt to be almost like Adobe Illustrator. Files can be uploaded and export is enabled to PDF, PNG, HTML and ZIP. Hotspots and linking is possible and Proto.io has great animation possibilities, everything can be animated. Specific loops and animation sequences can be done. UI is simple enough (see Figure 21), only some of the features have to be searched to find. Projects view is easy to understand and a new project is easy to make.

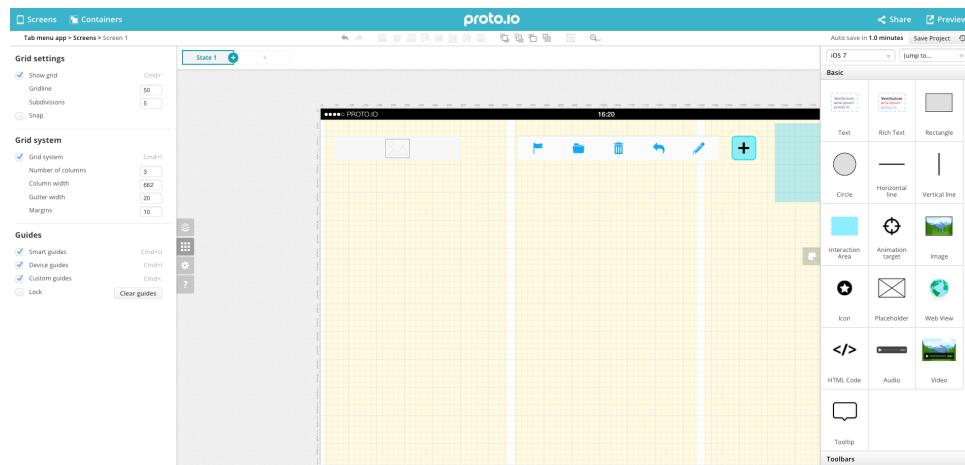


Figure 21. Proto.io has simple UI

### Testability and collaboration

Prototypes can be tested on an actual device, in browser or in a native app made for Proto.io called Proto.io Player app. When testing in a browser, tester can decide whether to test through white device or dark (see Figure 22). Proto.io webpages indicate that simple collaboration is enabled with instant feedback, free plan does not have the feature.

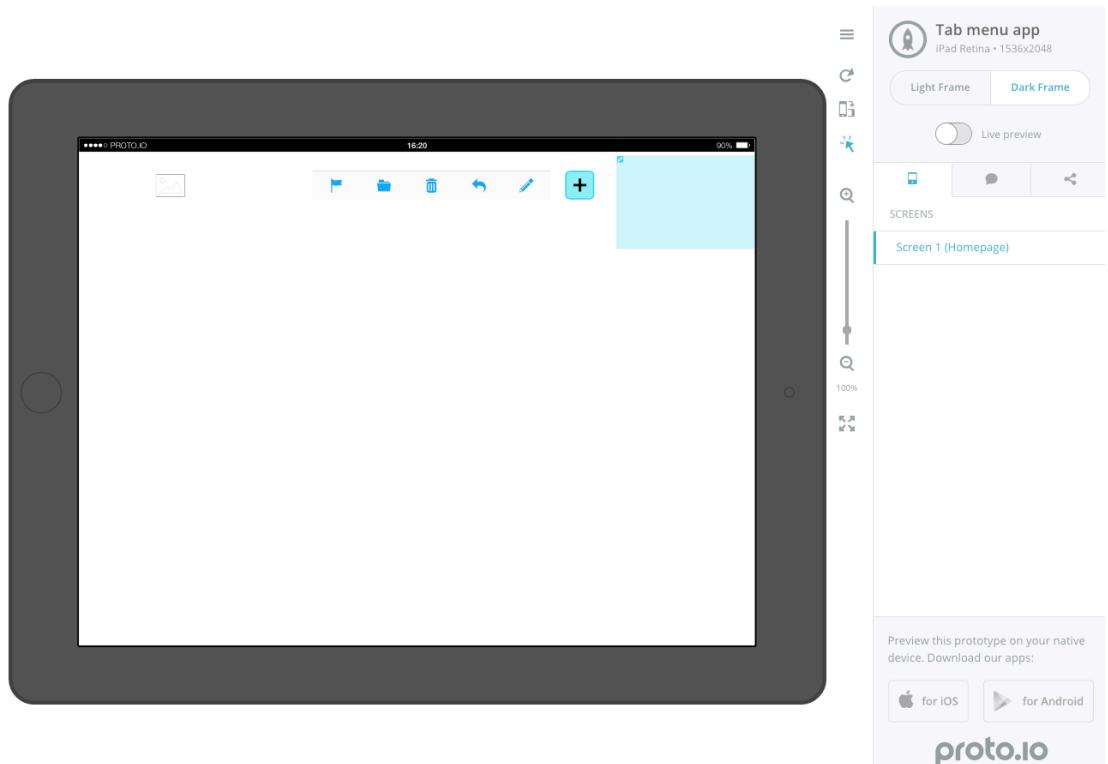


Figure 22. Browser testing with dark device

### **"Nice to have" features**

There are many Proto.io customers using the tool. Their webpage offers quick tips and step-by-step tutorials. There is a wide Learn & Support-page that includes plenty of video tutorials, documentation, forum and support. Proto.io goes to price category B and leans to price category D as well (see Table 6). It provides special offers for academic and non-profit organizations for giving -50% discounts.

Table 6. Plans and pricing for Proto.io

Plan name	Price \$/ licence / mo.	Price \$/ mo. prepaid annually	Licences	Projects	Reviewers	Other
Free plan	0	-	-	1	0	5 screens, no export or share
Freelancer	29	24	1	5	Unlimited	
Startup	49	40	2	10	Unlimited	
Agency	99	80	3-5	15	Unlimited	
Corporate	199	160	6-10	30	Unlimited	
Enterprise	Special features: more users and projects, enterprise-grade security, analytics, webinars, priority support					
Park	5	-	-	-	-	For keeping projects and data hosted

### **5.2.6 ProtoShare**

#### **Workability, designing, usability, formability**

ProtoShare is online UI tool. Formability of elements is possible. There are some difficulties using the tool when there are so many different features inside UI. Some of the features are hidden and the workflow suffers from hidden boxes that jump in

front of the project a user is working on (see Figure 23). Proto.io is saving itself every minute, user has to remember to click save button when leaving project.

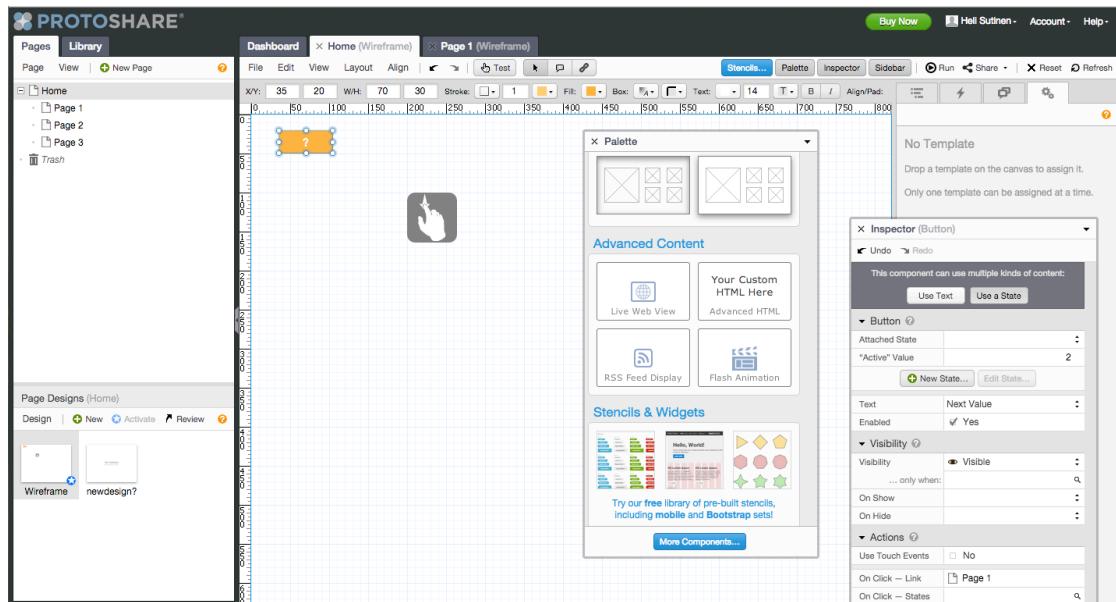


Figure 23. Project view of ProtoShare

It feels like some of the features which would not need to be showing all the time are showing and some of the main features required are hidden. For example, the library elements are behind stencils-button. Workflow for finding library elements is behind five steps. The figure below shows the steps with numbers from the first step to the fifth step marked with red lines (see Figure 24), which gives a bad experience of the usability.

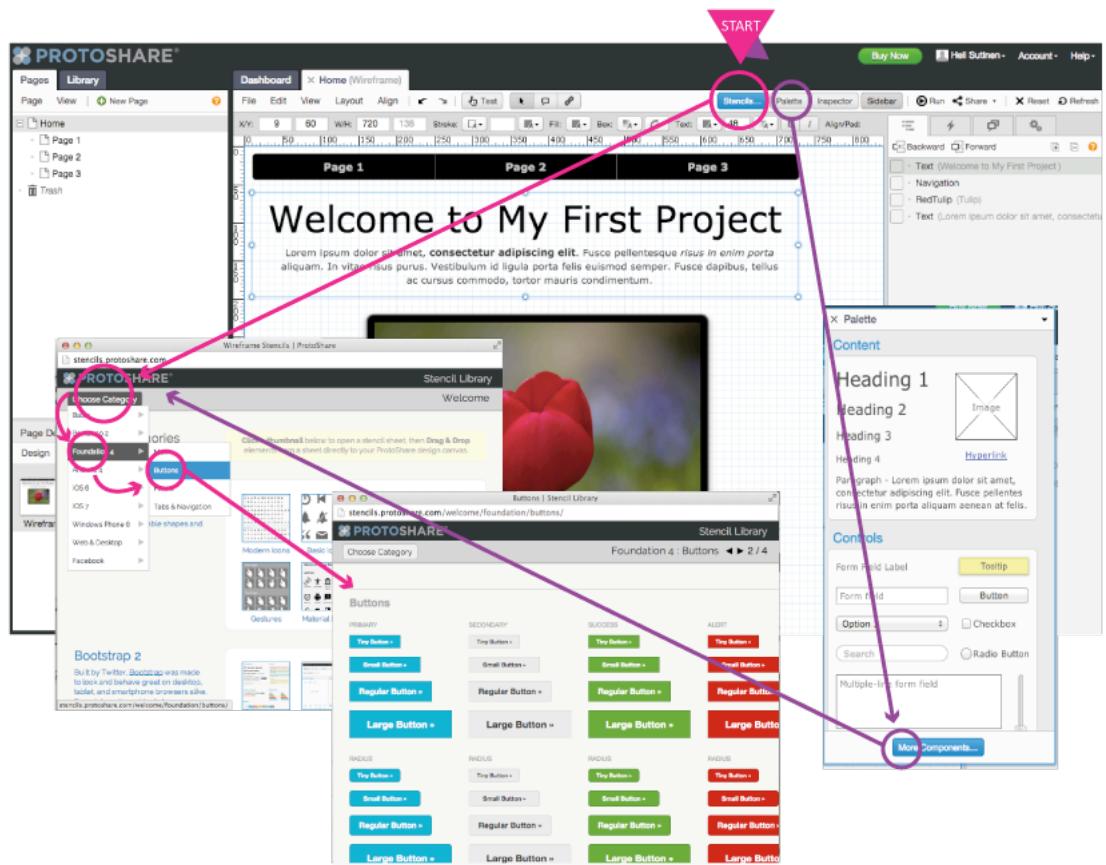


Figure 24. ProtoShare UI workflows for finding library elements

ProtoShare feels more of developer's design tool because programmer vocabulary is used. There are many interactions that can be added for elements and designer might not understand them all at first. It takes plenty of time to understand how ProtoShare works and to understand the programmer language. For example, the button for interactions can be found behind inspector-button, it might be something considered self-evident for some users but not for everyone. Library of components is wide, almost anything can be found from Bootstrap responsive grids and Foundation tabs and navigations to iOS 7 layouts (see Figure 25). User can add Google fonts into prototype. Pictures can be uploaded to library. Export feature is enabled to detailed Word spec document, PNG images and HTML archive or a hosted URL.

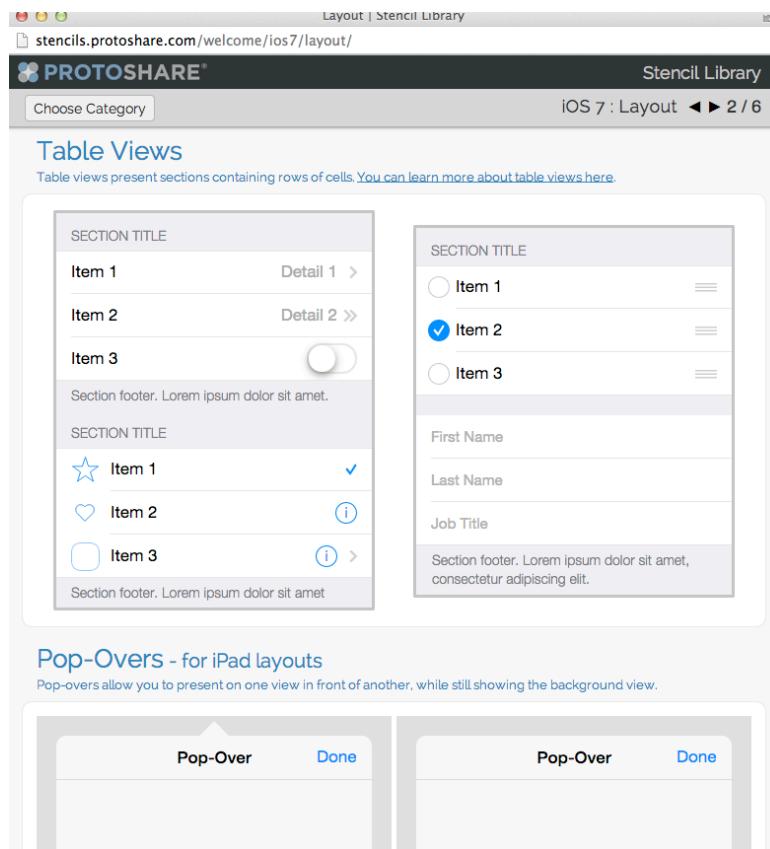


Figure 25. Some of the library elements of ProtoShare

### Testability and collaboration

Collaboration is made possible with “Review” interface. The link can be shared to reviewers. Reviewers can interact with prototypes by creating discussion topics and making comments. Discussion topics are shown as red pushpins on prototype. All the feedback is stored in a single location. ProtoShare includes a feature to add mockups and prototypes into Confluence and to Jira issues. The feature could be beneficial for N4S-Program that uses Confluence and Jira. It could bring mockups and prototypes of Contriboard and other products that are developed to everyone’s sight inside N4S-Program. This would enhance collaboration and enable more feedback.

### “Nice to have” features

As earlier revealed, button interactions are possible to make. This feature is one of the features that enable making high-fidelity prototypes with ProtoShare. There are webinars, video tutorials, a user guide and forum for helping users to use ProtoShare. The community seems huge. There are many customer endorsements

on the web page. ProtoShare prices are in category C (see Table 7) and Business-plan goes to price category A and D when buying annually and buying for more than 5 licenses. Annual pricing discount is -10% for Business licenses. By choosing the annual billing with Business license, business only features are enabled, as white labeling, Always-on SSL self-hosting and free one hour training session with 5 or more editors.

Table 7. Plans and pricing for ProtoShare

Plan name	Price \$/ licence/ mo.	Licences	Projects	Reviewers/ project	Storage/ editor
Standard	29	1	Unlimited	0	1GB
Professional	49	1	Unlimited	10	2 GB
Business	59	1	Unlimited	Unlimited	5 GB

### 5.2.7 UXPin

#### Workability, designing, usability, formability

UXPin has many features. UI has so many buttons (see Figure 26) that it takes time to understand what everything does. Still it does not take much time to learn how to use the tool. There are many libraries having plenty of components. The workflow differentiates from other same kind of UI design tools with the idea of different layout alternatives being just components in libraries. Usually user can choose to make for example a tablet prototype and the project opens with tablet background. UXPin has tablet, mobile, web etc. backgrounds as components in library; user needs to drag backgrounds from library. It is a different way of making the same thing. Linking views to each other is possible through interactions. Since having so many features UXPin feels slightly slow when designing wireframes or prototypes. Exporting projects is possible to PDF, PNG and HTML.

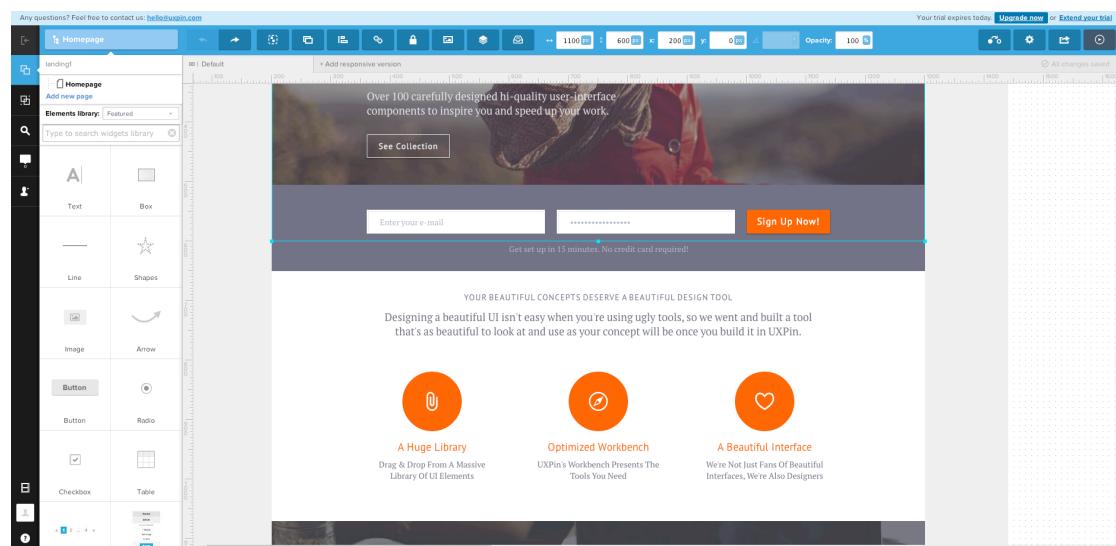


Figure 26. UXPin user interface has many buttons

### Testability and collaboration

Testing is possible through link only on a browser. The more complex prototypes user is making, the more time it takes to load the link. Slow loading makes testers feel frustrated. When editing a project, user can make observations on design by adding notes. Collaboration feature is well done. New comments are shown red, open comment is blue and resolved comments are green (see Figure 27). Comments can be voted by giving a heart when comment is good.

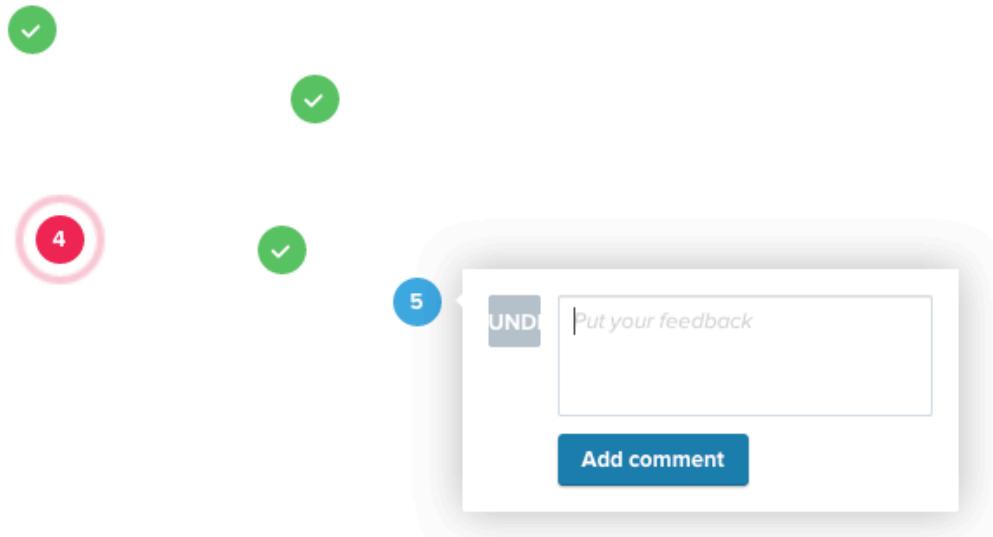


Figure 27. Comments are shown as colored circles

### **"Nice to have" features**

Many interactions can be done. Every element can have interactions. Multiple elements can be changed to be as one by making a smart element. The smart element feature fastens the workflow when user does not need to make the same interactions to every element. With pricing UXPin has changed from one simple plan \$14.99 to more feature based pricing (price category D) with three different plans (see Table 8). Basic and Pro plans have so limited features that the price gets too expensive for getting everything needed.

Table 8. Plan content and pricing for UXPin

Plan name	Price \$/ licence/mo.	Licences	Projects	Other
Basic	15	1	Unlimited	No UI libraries or live share
Pro	25	1	Unlimited	No live share
Pro+	40	1	Unlimited	Usability testing coming soon

### **5.2.8 Comparison results of the second phase**

Comparing all tested tools in general, the tools have some same and different features. All the tools tested have some kind of comment-feature. A comment-feature is important when involving the whole development team in the design process. Collaboration is the key to LEAN UX.

To clarify tool comparison, tools can be categorized in three groups by their usage:

- Making mockups clickable: Concept.ly, InVision App
- Making wireframes, mockups clickable: FluidUI , NinjaMock
- Making wireframes, mockups clickable and prototypes: Proto.io, ProtoShare, UXPin

Concept.ly and InVision App work in the same way; they import images and make hot spots. The comment feature works on both tools nicely. While following the development process of these tools, InVision App has grown more. The increasing

community and major growth of features indicate that InVision App has stabilized its place in UI design tool markets. Usability of InVision App is more explicit and the tool feels more professional. If utilizing InVision App in design process, there needs to be another tool for making wireframes.

Making wireframes needs to be as fast as using pen and paper. When using UI design tools instead of pen and paper there needs to be a fast linking and sharing possibility. NinjaMock and FluidUI are similar tools for making wireframes. FluidUI offers more stylized library elements than NinjaMock, which is not necessarily a good feature. Less stylized and more sketched looking wireframe mixes less on the final design. Sketched look gives more space for opinions. It lets commenters to pay more attention on positioning of elements and usability than outlooks and styles. FluidUI developers are still working on the comment feature. NinjaMock offers better features for easing and speeding design process. ProtoShare has huge library of elements, still bad usability took its share of using it. Designing is hard when most of the design view is hidden behind open boxes and other views. Proto.io and UXPin prices are too high for the project. UI design tools have plenty of different pricing methods. Most of the tools are still focusing on pricing based on the amount of projects or limiting features (see Table 9).

Table 9. Summary of UI design tools price categories

Price category	A	B	C	D
Concept.ly		x		
InVision App		x		x
FluidUI		x		x
NinjaMock				x
Proto.io		x		x
ProtoShare	x		x	x
UXPin				x

Price category A, license quantity based pricing.

Price category B, license and project quantity based pricing.

Price category C, reviewer and storage quantity based pricing.

Price category D, strongly on features based pricing.

The selected tools for design process of Contriboard are InVision App, NinjaMock and UXPin. The selection of most suitable tools for design process of Contriboard is

tested more clearly in action to evaluate how the tools fit to the design process of the reference product. Seeing the testing tools in action in real production process gives answers to the research questions.

## 6 Case: Contriboard

### 6.1 History and present of Contriboard

The thesis includes designs approach to developing a reference product Contriboard for N4S-Program. The product was originally a component of the FreeNest Product Platform 1.4 developed during 2011–2013. It was called Teamboard and was built on the need to have a ticket board, which could be used as a Kanban-tool for teams. The main idea of the product is to make boards that act as brainstorming base for ideas. An idea is thrown on the board in form of a ticket that user can move. It could be referred to as a substitute for writing thoughts on post-its. N4S@JAMK team is building Contriboard as a product that does not include FreeNest Product Platform. Through the transformation the product changes its appearance for more simpler and fresher look and turns responsive for tablets (see Figure 28).

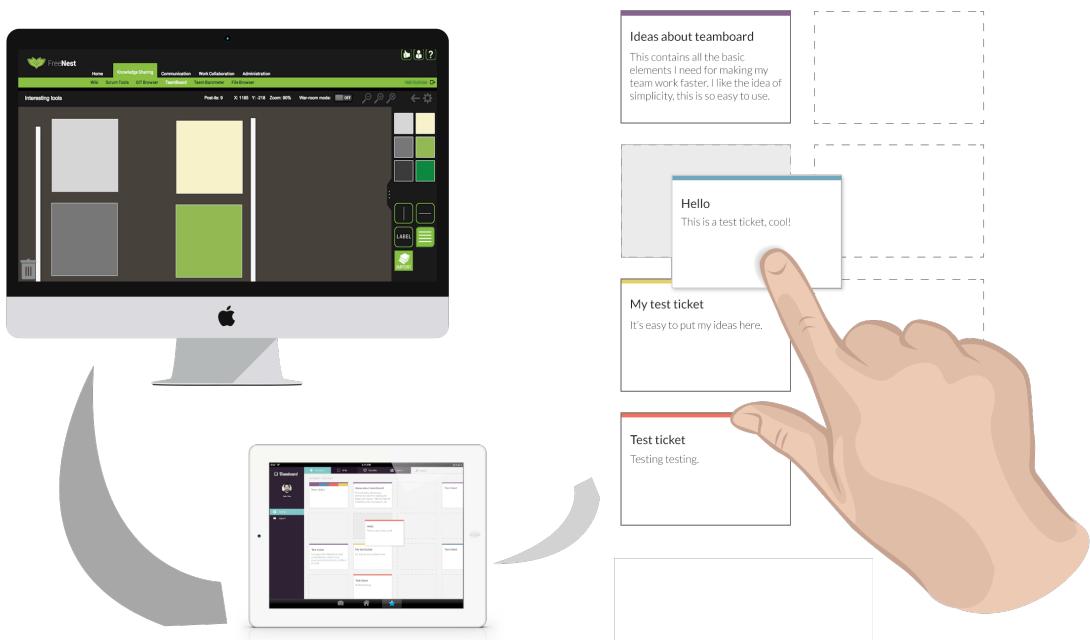


Figure 28. Illustration showing the transformation of Contriboard

## 6.2 Service Design Workshop

The service design workshop of Contriboard was held with the service design professionals and team in its entirety of 18 people. The service design workshop helped to perceive a common vision and understanding in what direction the product should be developed to make it successful. The workshop was categorized into three parts, which are introduced within their own chapters below.

### 6.2.1 Defining objectives

First of all it is important to define providers' objectives and users' needs. It is important to think what kind of service should be provided there and what kind of a service the customer needs. If there is a good idea but no need for the service, it is unprofitable to build it. As a result of this stage the team understood the importance of continuing to develop Contriboard. There are many tools having same kind of idea and the same problem. All products are complex to use, color blindness is not taken into account and initializing is time taking.

### 6.2.2 Gathering test data

The most convenient way to collect data was to use traditional ways with some of the design ways included. Interviews brought data about good and bad experiences and observations were made by giving tasks and observing users' ways to use the existing product, asking questions containing words "what", "why" and "how" without leading the user and making notes.

### 6.2.3 Integrating observations and results

The observations were changed into needs. Some cases in observation phase the user did not find the way making some task; therefore the need was discovered for a product that is more easy to use. By integrating observations came up the solution that Contriboard should be the project management tool supporting teamwork and helping the team to communicate with each other.

Suggestions listed below for improving usability:

- Advice user how to use the product
- Support user's individual needs
- Use the same language as the user

All material gained through the Service Design Workshop for Contriboard is taken into account when developing the new Contriboard. Improvement observations made the team wiser and gave new innovative ideas making Contriboard user-friendlier.

### **6.3 Testing UI design tool in design process of Contriboard**

To get valid answers to research questions there was need to use the tools in real development process. Developing of Contriboard consists of continuous development and iterations. It has been ongoing before evaluation and after, the whole design process could not be included to research. Following chapters introduce individual situations of design process in multiple iterations. The first UI design tool testing is made using NinjaMock and includes designing and usability testing of functionality of tickets. The second testing is executed using InVision App tools free version and ready-made mockups for testing and voting different versions of sidebar designs. As stated in testing UXPin the pricing is too high, however, the versatile usability and great comment feature attracted to test tool in action. The third testing is made using UXPin free trial and testing how fast it is to make a mockup of Contriboard landing page.

#### **6.3.1 Low-fidelity mockups of UI with NinjaMock**

While planning on functionality of tickets there were problems to be solved. Designing for tablet provided more challenge at this point. User needs to have the ability to move tickets on the board and edit the content. Considering options of tap, double tap and long pressing, only the tap option was most usable. From the user point of view the double click is hard to discover and it argues with the ease of use. Long presses have been commonly used with Android and Windows tablets; however, just for some additional features have the possibility to be long pressed not any vital actions. When thinking of long press from a user's point of view there might

be a situation when the user is trying to make a swipe move on the screen and accidentally keeps the finger too long on the screen and the long press move happens.

Designing functionality of tickets there were two ideas how editing and deleting tickets could work when using tap. The team considered the idea of the edit/delete mode for the tickets to ease users' work with tickets. Pushing edit button (see Figure 29) tickets would go into the edit mode where user could edit or delete tickets but not move them. Low-fidelity mockups are modified in the figure below using Adobe Illustrator to illustrate and simplify how low-fidelity mockup changes when going to the edit mode and deleting a ticket. Pink areas with numbers show that user needs to make three taps to delete a ticket and one more to leave the edit mode.

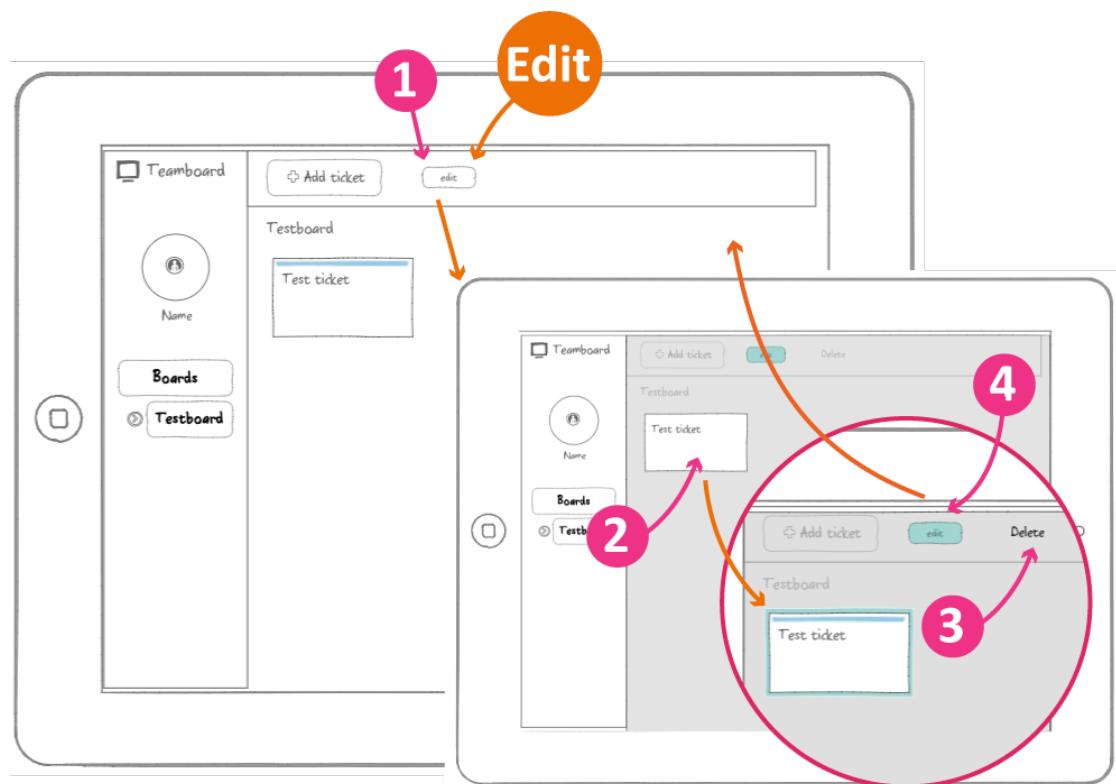


Figure 29. Edit/delete mode functionality and number of taps

Another idea the team considered was to have a select button in the corner of every ticket and after selecting a ticket the toolbar would show the edit and delete buttons. When selecting a ticket/ tickets user could delete multiple tickets at the same time and in the future development probably edit multiple tickets

simultaneously. The number of taps user has to make to delete one ticket is two (see Figure 30). Low-fidelity mockups are modified in the figure below using Adobe Illustrator to illustrate and simplify how a low-fidelity mockup changes and what the number of taps is.



Figure 30. Idea of select to delete ticket

Making low-fidelity mockups and using hot spots NinjaMock helped to bring ideas in to testing level before doing any code. Clickable mockups helped the team to understand clearly by testing two alternatives, which alternative takes more clicks from the user and which would be an easier way for the user to use it. After testing two clickable low-fidelity mockups with NinjaMock, the team changed opinions from the earlier idea about edit mode in to the selection idea. The edit mode required more clicks from the user. It was easy to demonstrate ideas by giving just a link to testing the clickable mockups.

### 6.3.2 UI mockups clickable with InVision App

Contriboard consists of two main views, the first view is for making boards and the second view is inside the board where tickets can be made. The main view has a

sidebar on the left (see Figure 31) where user can change language, give feedback or log out.

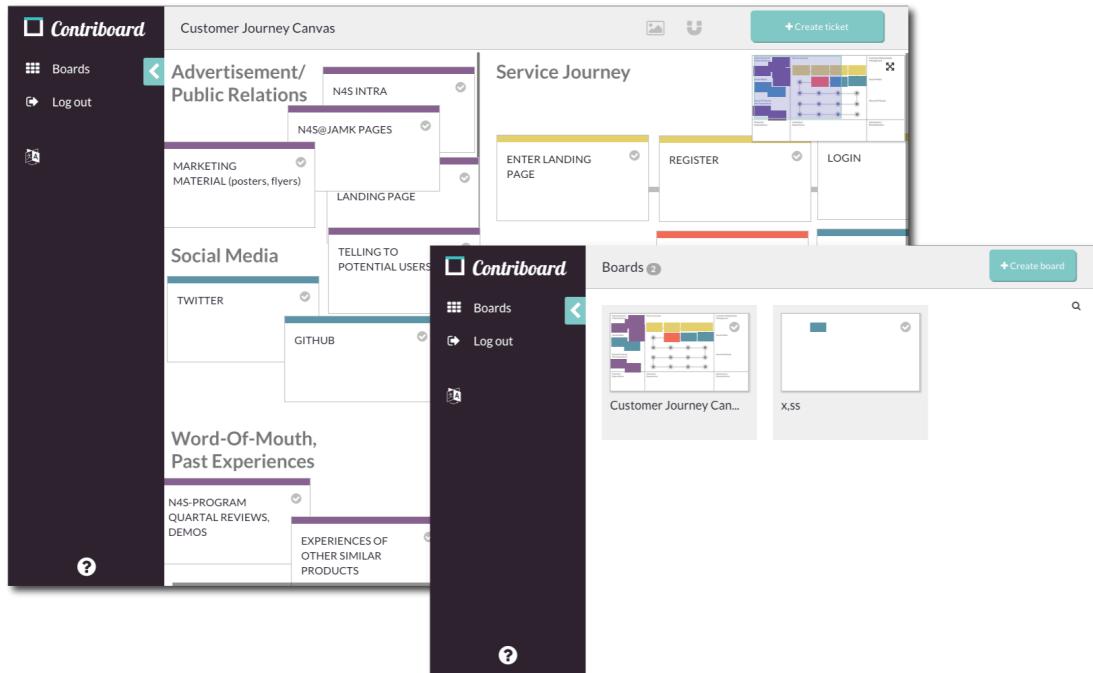


Figure 31. Screenshot of ticket view left and boards view right

The team needed quick demonstrations about a collapsing sidebar. The idea was to collapse the sidebar hidden by clicking a button in order to leave all the space for working. There were two different ideas, collapsing sidebar completely or collapsing it to be smaller. Making low-fidelity prototype demonstration needed high-fidelity mockup designs showing sidebar hidden and visible. Both mockups (see Figure 32) are made with Adobe Illustrator. InVision App supports the needed features for importing pictures, to create hot spots, and to collect ideas.

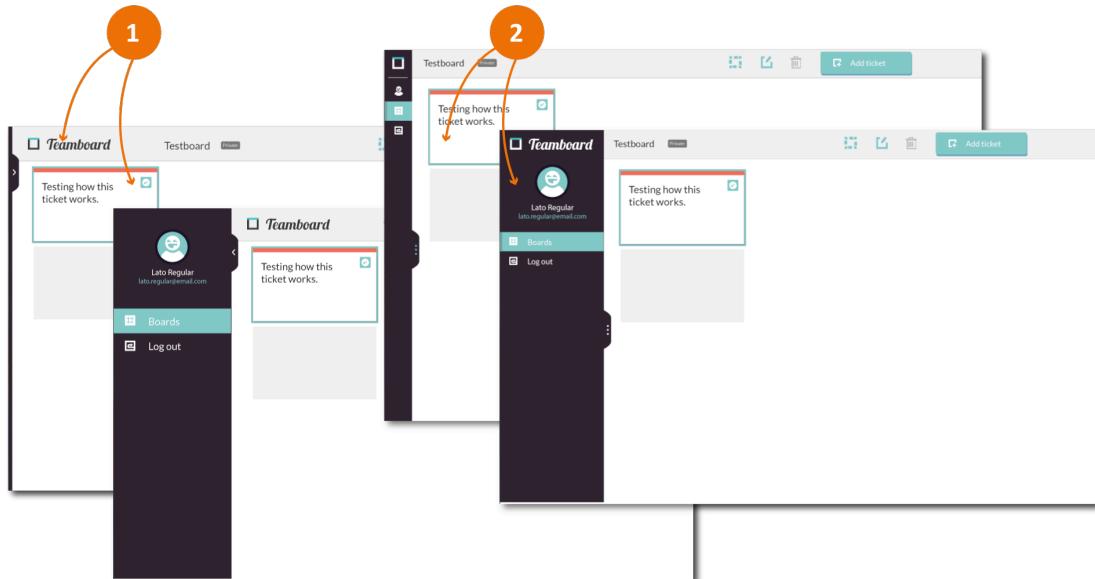


Figure 32. Mockups needed to include two sidebar ideas

The team tested both ideas through the link leaving comments. Leaving comments by clicking the spot anywhere the tester wanted was easy and did not take too much time. As being editor of the mockups, activity view in InVision App showed new comments immediately (see Figure 33).

Figure 33. Activity view of InVision App

### 6.3.3 Fast landing page mockup for Contriboard using UXPin

Designing the landing page was a great chance to test how UI design tool performs in real designing. In order to have fast progress with the tool, it needed to be thought what elements the design needs. Fast wireframes (see Figure 34) using pen and paper were made to reduce excessive relocating of elements when using UXPin.

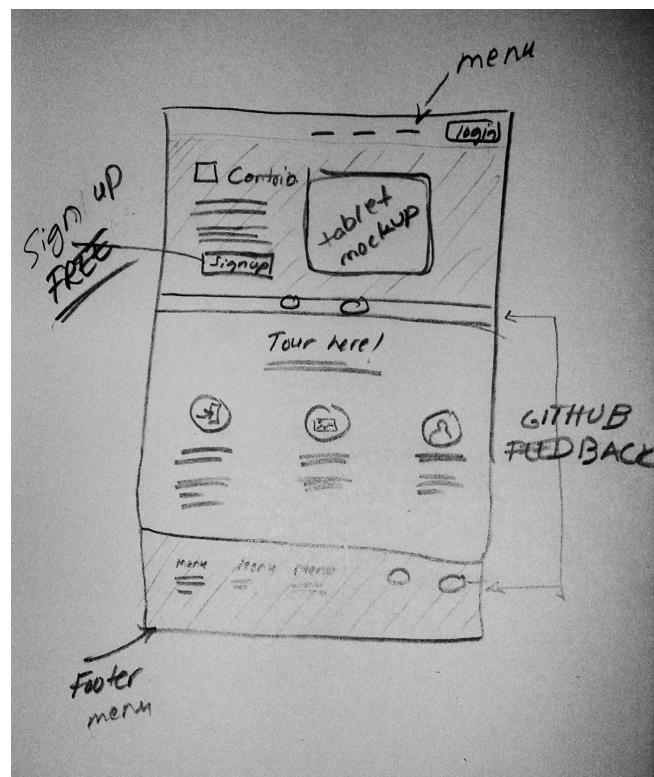


Figure 34. First wireframe of landing page of Contriboard

Contriboard uses Awesome font icons in UI and UXPin has got the icon-set in their element library. Having the same icons in use made executing high-fidelity mockup fast when the same icons were used on the landing page. All the elements used making mockup (see Figure 35) were found in the library.

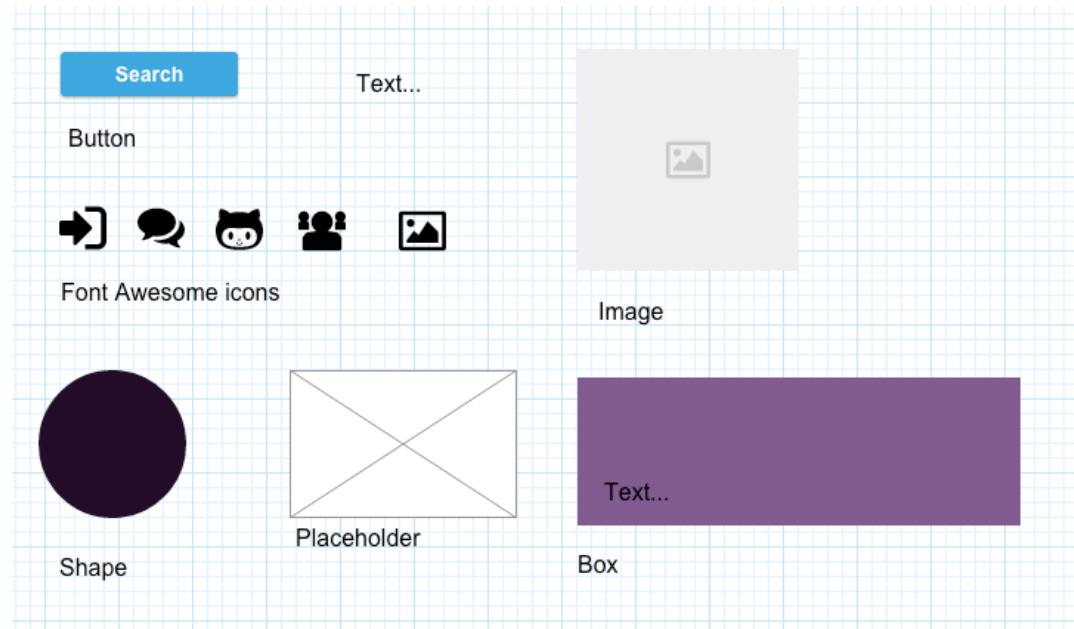


Figure 35. All the elements used in mockup landing page

Piling of landing page design was easy by dragging every element on its place. First there was no need for thinking of text content and “lorem ipsum” text was possible to make by clicking one button. Piling of elements took place within five minutes. Previewing design was easy and building of page into wanted form started. After changing colors of elements and fonts matching design guidelines and making some texts, the mockup was ready (see Figure 36). The figure shows high-fidelity mockup and final landing page together side by side.

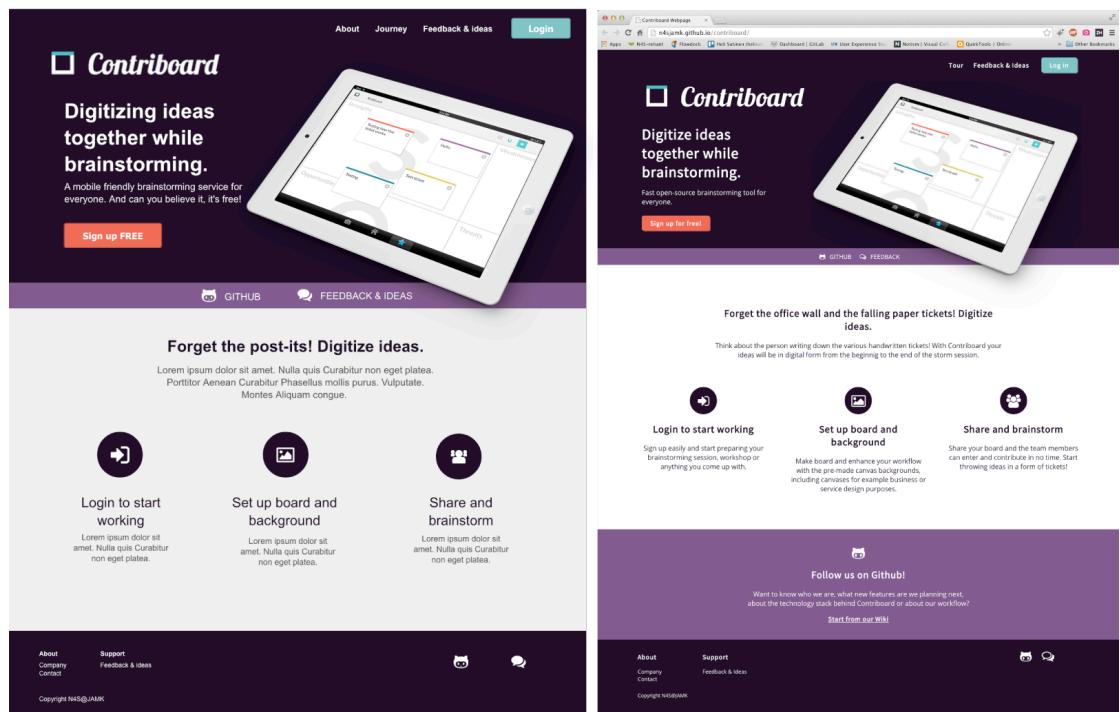


Figure 36. High-fidelity mockup and final landing page

Making mockups with UXPin is easy. It can be fast as well if some dysfunctions regarding to reliability of tool are not taken into account. While previewing the final mockup an error occurred and UXPin lost all elements from the edit mode. This can take designing back to start and no time has been spared. Mockup was still showing on the preview mode and capturing screen saved the design. If design had been incomplete everything would have needed to restart from the beginning. Reliability of the tool is most important.

## 7 Conclusions

### 7.1 Research results

The aims for the thesis were to research web based UI design tools in design process and the effect of tools being part of the software development process. The research questions were:

- Is there a right kind of UI design tool to be used in every situation?
- Does the designer provide more effective results and better collaboration by using UI design tools?

- Do UI design tools provide more effective way to increase speed of development process?

The right kind of UI design tool to be used in every situation is hard to find while changes are happening all the time and projects are changing. More important aspect could be to think if there is a right kind of tool for using in an ongoing job. This is when knowledge of available tools is required. Even when deeper testing was made, tools were being developed all the time and test results can become obsolete fast. New tools are developed all the time and a perfectly reliable answer cannot be obtained. The tests confirmed that using the right kind of UI design tools in design process, more effective results and better collaboration can be accomplished. The testing showed that using UI design tools to collect ideas and feedback, better collaboration and more communication inside the team has been accomplished. In a general level it was also noticed that communication inside the N4S@JAMK team must be improved, this is also what was reflected in the results of tests.

Using UI design tools in the design process helps to get developers and everyone else involved in the designing. UI design tools can provide a more effective way to increase the speed of the development process when some of the testing can be done using these tools before any code is made. Tools can be used as a vanguard for showing the right direction to take in the development process.

## 7.2 Research critique

The research of UI design tools was mostly successful. Great perceptions were made of the tested tools. However the status quo remained incomplete because of the large quantity of tools and all could not be included in the testing.

Second testing phase does not include testing of HTML and CSS-exports of the UI design tools, code export could have richen the testing. Observations considered useful to be tested in the future could be testing HTML and CSS-exports of UI design tools to ease and speed the front-end developers' coding work. Another interesting aspect could be to take customer in to the design process using UI design tools.

### **7.3 Reflections on the research**

During the research I gained plenty of new information of how the product development process need to be executed to gain most effort and how every phase in development process works most effective way. The thesis increased my skill level considerably.

I found the research to be truly rewarding for its results being positive. The research has shown the most effective way to rapidly produce results in designing phase of the product development process. My methods of designing have changed to more nimble and light wireframes and collaborative design.

## References

About Service Design. SDN UK. Accessed on 20 August 2014. Retrieved from [http://uk.service-design-network.org/?page\\_id=27](http://uk.service-design-network.org/?page_id=27)

All About the Waterfall Model. 2014. Accessed on 6 September 2014. Retrieved from <http://www.waterfall-model.com/>

Blank, S. 2013. Why the Lean Start-Up Changes Everything. Harvard Business Review article. Accessed on 13 September 2014. Retrieved from <http://hbr.org/2013/05/why-the-lean-start-up-changes-everything/ar/1>

Childs, M. 2011. John McCarthy: Computer scientist known as the father of AI. The INDEPENDENT article 1.12.2011. Accessed on 23 August 2014. Retrieved from <http://www.independent.co.uk/news/obituaries/john-mccarthy-computer-scientist-known-as-the-father-of-ai-6255307.html>.

Delgado Garcia, J. 2014. Conoce nuevas metodologías del Silicon Valley: Lean Startup Crear, Validar Aprender. Coffeework blog 16.7.2014. Figure. Retrieved from <http://coffeework.eu/conoce-nuevas-metodologias-del-silicon-valley-lean-startup-crear-validar-y-aprender/>

DeVore, H. 2012. BMC Application Management. Figure. Retrieved from <https://communities.bmc.com/docs/DOC-19965>

Fluid UI. Desktop Settings. Fluid UI settings management application demo. Figure. Retrieved from <https://www.fluidui.com/wp-content/demos/screenflows/DesktopSettings.png>

Gladkova, E. 2012. Agile vs Waterfall Development Model. Figure. Retrieved from <http://sysgears.com/articles/agile-vs-waterfall-development-model/>

GoodWorkLabs. Figure. Retrieved from <http://www.goodworklabs.com/process/>

Hurn, C. 2012. Stuffed Giraffe Shows What Customer Service Is All About. Article 17.5.2012. Accessed on 22 October 2014. Retrieved from [http://www.huffingtonpost.com/chris-hurn/stuffed-giraffe-shows-wha\\_b\\_1524038.html](http://www.huffingtonpost.com/chris-hurn/stuffed-giraffe-shows-wha_b_1524038.html)

Hüttermann, M. 2012. DevOps for Developers. New York: Apress.

Järvinen, J. 2014. N4S-program (Need for Speed). YouTube video 4.4.2014. Accessed on 22 August 2014. Retrieved from <https://www.youtube.com/watch?v=ZIWfEjnTu4g&list=UUPLEtMCZ51EzUEAS-r-VfLQ>

Lusch, R. Vargo, S. 2004. Evolving to a New Dominant Logic for Marketing. Journal of Marketing. Accessed on 31 October 2014. Retrieved from  
<http://dret.net/lectures/ssme-spring07/VargoReading-Recommended.pdf>

Manifesto for Agile Software Development. 2001. Accessed on 7 September 2014. Retrieved from <http://agilemanifesto.org/>

Pant, R. 2009. Organizing a digital technology department of medium size in a media company. Figure. Retrieved from  
<http://www.rajiv.com/blog/2009/03/17/technology-department/.>

Paradigm Change – Delivering Value in Real Time. Accessed on 17 June 2014. Retrieved from [www.n4s.fi/en](http://www.n4s.fi/en), Work packages, Delivering value in real time.

Parsons, R. 2012. Continuous Delivery with Continuous Design: Completing the Cycle. Accessed on 1 October 2014. Retrieved from  
<http://www.infoq.com/articles/continuous-delivery-continuous-design>

Ries, E. 2009. Minimum Viable Product: a guide. Accessed on 13 September 2014. Retrieved from <http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html>

Ries, E. 2009. Pivot, don't jump to a new vision. Accessed on 16 September 2014. Retrieved from <http://www.startuplessonslearned.com/2009/06/pivot-dont-jump-to-new-vision.html>

Rightscale Cloud Computing Trends: 2014 State of the Cloud Survey. 2014. Accessed on 25 August 2014. Retrieved from <http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2014-state-cloud-survey.>

Riley, C. 2014. Releasing your team from Release Thinking. Blog post 9.6.2014. Accessed on 17 June 2014. Retrieved from <http://devops.com/blogs/releasing-team-release-thinking/>

Roscam Abbing, E. 2010. Customer journey mapping. Slideshow 17.11.2010. Accessed on 6 November 2014. Retrieved from  
<http://www.slideshare.net/brandriveninnovation/consumer-journey-mapping>

Salo, I. 2010. Cloud computing palvelut verkossa. Jyväskylä: WSOYpro.

Salo, I. 2012. Hyötyä pilvipalveluista. Jyväskylä: Docendo.

Scrum Methodology: Scrum Meetings. 2011. Accessed on 8 August 2014. Retrieved from <http://scrummethodology.com/scrum-meetings/>

Shubert, L. 2010. Learn Product Design Innovation--the IDEO and Stanford d.school way. Innovation Process Figure. Accessed on 10 November 2014. Retrieved from  
<http://www.eventbrite.com/e/learn-product-design-innovation-the-ideo-and-stanford-dschool-way-tickets-829532153>

Stickdorn, M., Schneider, J. 2010. This is Service Design Thinking: Basics, Tools, Cases. Netherlands: BIS Publishers.

Stickdorn, M., Schneider, J. 2010. The Customer Journey Canvas. Retrieved from <http://thisisservicedesignthinking.com/>, Customer Journey Canvas.

Topics in Scrum. 2014. Accessed on 7 September 2014. Retrieved from <http://www.mountaingoatsoftware.com/agile/scrum/overview>

Turunen, A. 2014. The N4S Program is building new infrastructure for industrial software applications. Article 26.4.2014. Accessed on 22 August 2014. Retrieved from <http://www.n4s.fi/2014magazine/article1/>

Walls, M. 2013. Building a DevOps Culture. O'Reilly.

Waters, K. 2007. What is Agile? (10 Key Principles of Agile). Accessed on 5 September 2014. Retrieved from <http://www.allaboutagile.com/what-is-agile-10-key-principles/#sthash.OdqWKurR.dpuF>

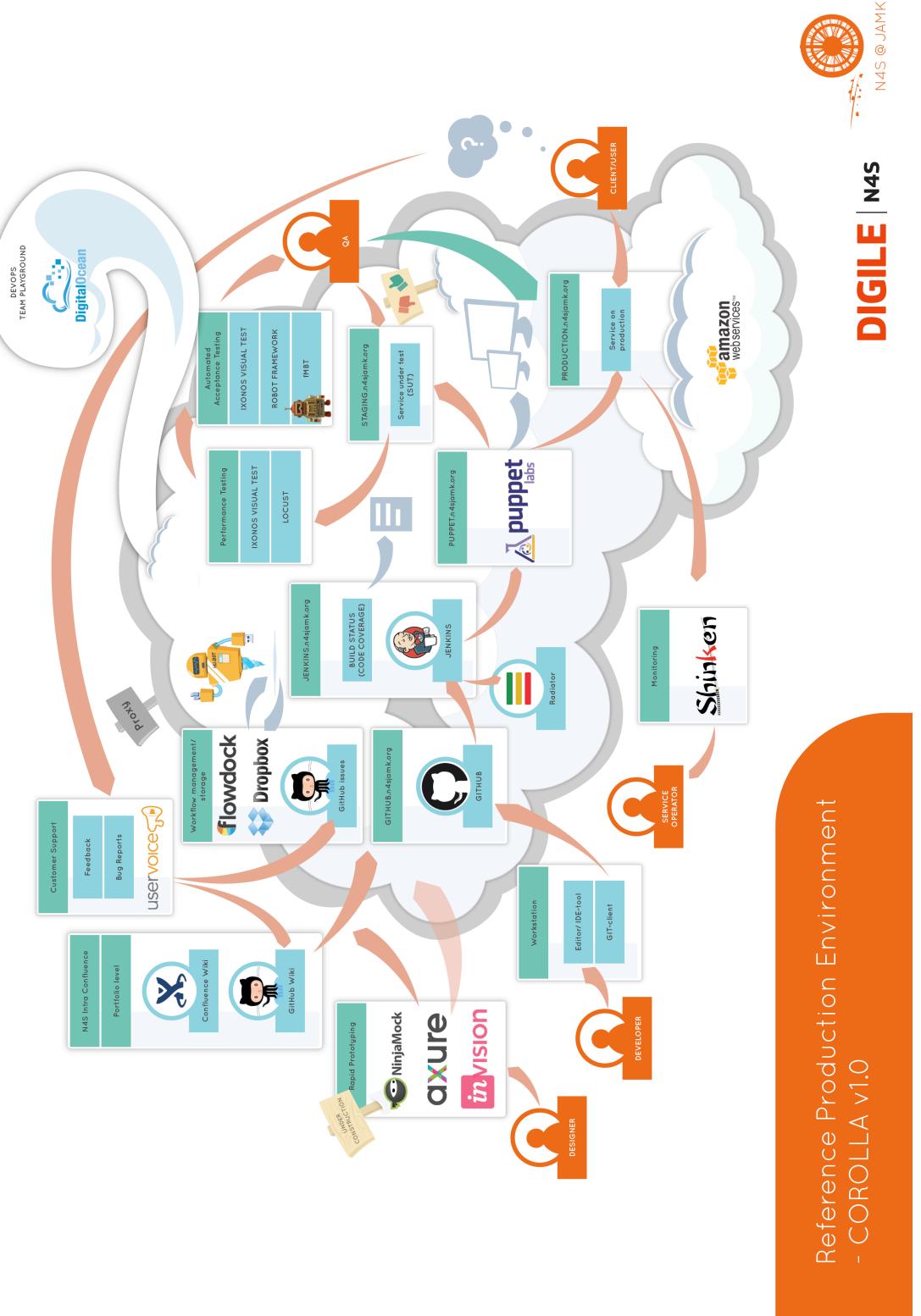
What is Agile Software Development?. 2014. Accessed on 4 September 2014. Retrieved from <http://www.agilealliance.org/the-alliance/what-is-agile/>

Yip, J. 2011. It's Not Just Standing Up: Patterns for Daily Standup Meetings. Accessed on 7 August 2014. Retrieved from <http://martinfowler.com/articles/itsNotJustStandingUp.html>

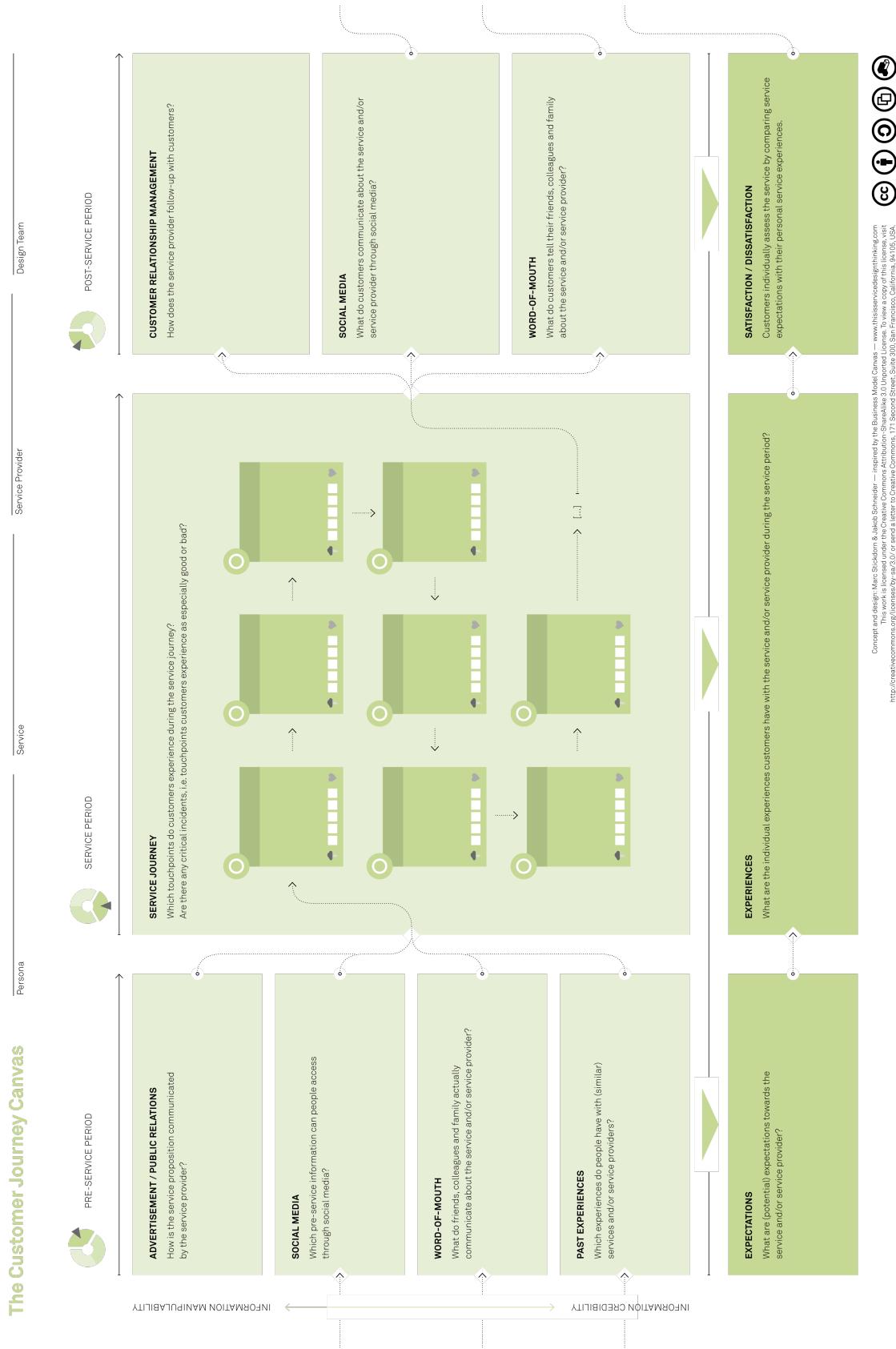
Ziegler, C. 2011. Nokia CEO Stephen Elop rallies troops in brutally honest 'burning platform' memo? (update: it's real!). Engadget article 8.2.2011. Accessed on 6 November 2014. Retrieved from <http://www.engadget.com/2011/02/08/nokia-ceo-stephen-elop-rallies-troops-in-brutally-honest-burnin/>

## Appendices

### Appendix 1. Reference production environment COROLLA



## Appendix 2. Customer Journey Canvas



### Appendix 3. UI design tool survey questions

1. In what are you using the tool?
2. What kind of features would be important for this tool to have so you could do what you are doing?
3. Did you find all the features you needed?
  
4. How easy it is to use this tool?

1	2	3	4	5
Very hard <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Very easy				

5. Why it is/is not easy to use?
6. Can you find tutorials and help?

1	2	3	4	5
No help found <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Lot of tutorials and help				

7. What kind of help you can find?
8. Formability of elements, e.g. modifying shapes and buttons.

1	2	3	4	5
No modifications possible <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Modifications can be made widely.				

9. Is it possible to make high-fidelity prototypes with this tool? If not, why?
10. Is it possible to make interactive elements (e.g. mouse over button, colour changes)?
11. What can you do with this tool?
12. How simple it is to make interactions to interactive elements?
13. Is the tool working properly (slowing down, does not work etc.)?
14. What grade would you give to this mockup tool?

1	2	3	4	5
Bad. Wouldn't use anymore. <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Great! Would recommend others to use as well.				

15. Other comments?

## **Appendix 4. UI design tool survey responses**

### **UI design tool (name, version, evaluation date):**

**Concept.ly, Beta, 9.7.2014**

1. Web page & UI (Toolbar).
2. Interactions.
3. There isn't really that much features.
4. 3
5. There isn't that much features, but it's still kind of hard to use.
6. 3
7. Video tutorials.
8. 1
9. No. Well, it depends how good pictures you've made of your "product"
10. No interactions
11. Mockup, wireframe
12. No interactions
13. working just fine.
14. 2
15. Feels a bit strange to me. Can't make any wow-effects, but it's still kind of hard to use.

### **UI design tool (name, version, evaluation date):**

**Easel, 7.7.2014**

1. Web page & UI (Toolbar).
2. I think it's all there what I need.
3. For some features I had to use search to find them.
4. 2
5. There's so many features, so it takes some time to learn it.
6. 5
7. Basic tutorial tours, Help docs.
8. 5
9. No, doesn't have that many features.
10. No interactions, only basic clicks to links.
11. Mockups, wireframes
12. No interactions
13. Part of the time it was slowing down. And yes, it takes some time to download the pages in preview mode.
14. 4
15. Easel feels the same as Adobe Photoshop or Illustrator, just a bit simpler (not that much features). You can create a lot of things with it. Easel has short cuts, so after a while it's really fast and easy to work with it. But it takes some time to learn how it works, 'cause there's so much what you can do.

**UI design tool (name, version, evaluation date):****FluidUI, 22.5.2014**

1. Making simple user interface for coffeemaker for Unity Mockup demo
2. Generating URL where the mockup is available and can be used. It was important to be able to modify the interface on the fly for the Unity Mockup demo. Being able to import own images was also important.
3. FluidUI wasn't very extensive on the features, but I found all the basic features such as what page each button opened. You couldn't make any functionalities on the page. You could only link pages to each other.
4. 4
5. FluidUI's design is very simple, there are not so many features, workflow is intuitive
6. 2
7. There are some sample demos and a guide
8. 3
9. No, there are not so many features or modifying possibilities
10. No interactions, just different kind of clicks (eg tap, press and hold, press)
11. flowcharts, slideshows perhaps
12. No interactions
13. Froze couple of times, had to run few times before changes appeared on generated URL, lost images few times
14. 3
15. –

**UI design tool (name, version, evaluation date):****InVision App, 7.7.2014**

1. Web page & UI (Toolbar).
2. Interactions.
3. There isn't really that much features.
4. 5
5. It's simple to use and it has simple features.
6. 3
7. Basic Help-tips and some YouTube-videos.
8. 1
9. No, features are too low.
10. No interactions, none.
11. mockup
12. No interactions
13. Works well, no problems.
14. 2
15. It's a lot of different than other Mockup Tools. Can't create any elements with InVision – you need to have layout in full images (JPG, PNG, GIF). Only thing that you can do, is to set the background color, hurray!

**UI design tool (name, version, evaluation date):****InVision App, 28.11.2014**

1. I have hi-fi mockups of product called Contriboard. I need to collect feedback about mockups from team.
2. Import files, linking mockups to each other, sharing and collecting feedback.
3. Yes.
4. 5
5. UI is simple and easy to use. I don't need to help my team to use shared mockups, they know how to comment mockups without help.
6. 5
7. Website has blog and FAQ-site. I can find tutorials from youtube.
8. 1
9. Yes. If high-fidelity prototypes include interactions, those can't be made, unless making mockups of every frame and connecting those together, then high-fi prototypes can be done, but it is very time taking and not agile at all.
10. No interactions. No interactive elements, but you can make interactions by making hot spot which changes the whole mockup when mouse over or when clicking.
11. Middle-fidelity prototypes, import mockups and make them clickable by making hot spots, share to others and collecting feedback.
12. No interactions
13. It works 100%.
14. 5
15. Tool was great for linking ready maid mockups together and gathering feedback. UI is simple and looks professional. It's great for these purposes, but if I would need to make wireframes, I would need to use some other tool.

**UI design tool (name, version, evaluation date):****Justinmind Prototyper 5.6.1, 10.7.2014**

1. Web page & UI (Toolbar).
2. I think it's all there what I need.
3. After looking a round - yes.
4. 3
5. There's so much going on - lot of different menus and things.
6. 5
7. Video tutorials, step-by-step tips, forum and user guide
8. 5
9. Yes
10. Lot of interactions. Actions, events, animations, drag and drop.
11. high-fidelity prototypes, mockup, wireframe.
12. Requires some time
13. No troubles.
14. 3
15. Nice software to work with. Needs time to learn how to use it. Sharing the prototype is a bit strange and slow.

**UI design tool (name, version, evaluation date):****Moqups, 13.10.2014**

1. Clickable wireframes.
2. Tool to make shapes, library of templates. Sharing and commenting.
3. There are shapes, and library templates like buttons. Sharing is possible but I can't make or have comments, because of the free version... too bad.
4. 5
5. Simple UI.
6. 5
7. Help center where I can find anything. Community and support pages.
8. 3
9. No, mockups can be imported and linked but there are no interaction possibilities.
10. No interactions. Just hot spots and linking
11. Wireframes, mockups, clickable mockups.
12. No interactions.
13. Yes it is.
14. 3
15. Library of elements was confined. This is why it would be nice that I could rotate elements to use them the way I want. For example library has horizontal slider element, I needed vertical but I couldn't rotate it. I sure can draw my own slider, but it takes more time. It is huge disadvantage that tool did not have commenting possibilities in tryout version. It would have been nice to try if it works and how. These things lowered the grade of the tool.

**UI design tool (name, version, evaluation date):****NinjaMock, 27.11.2014**

1. Designing UI wireframes for product called Contriboard
2. linking, library elements, sharing of link to wireframes and commenting.
3. Yes, all the features was there.
4. 5
5. Every tool I need is showing. I don't have to be looking for anything.
6. 4
7. youtube videos. Tutorials are not needed that much, UI is so simple.
8. 4
9. No. This tool is for wireframing and making hot spots to wireframes.
10. No interactions
11. wireframe, share wireframes and collect comments.
12. No interactions
13. Sometimes some elements don't get saved, it's frustrating to pile up elements again... When sharing link to wireframes, sometimes when I change starting wireframe, it does not change to link and wrong wireframe is at the starting wireframe in link.
14. 4

15. This is a great tool for making clickable wireframes and commenting is simple. Only those bugs lowered grade. It's frustrating that I had to make some elements again because they disappeared.

**UI design tool (name, version, evaluation date):**

**Proto.io v5, 3.7.2014**

1. Toolbar: web page and UI.
2. I think it's all there what I need.
- 3 So far I did.
4. 5
5. UI is really simple. If you don't get something you'll find it from tutorials.
6. 5
7. Video tutorials, step-by-step tips, quick tips, Learn&Support-page, documentation.
8. 5
9. Yes
10. Lot of interactions. Events, actions, screen transitions, animations. Lot of everything.
11. mockup, wireframe, high-fidelity prototypes.
12. Simple and fast
13. When having several states in one screen it's slowing down (ccould also be like that because using free-trial).
14. 5
15. You can create a lot with Proto.io. It lets you to do so much, that you can in some point forget using Adobe Photoshop or Illustrator.

**UI design tool (name, version, evaluation date):**

**ProtoShare 1.2.1 (not sure about the version), 9.7.2014**

1. Web page & UI (Toolbar).
2. I think it's all there what I need.
3. Yes, after searching them.
4. 3
5. Some features are "hidden", can't find them with just one look.
6. 5
7. Docs and videos.
8. 5
9. Yes
10. Lot of interactions. Actions, animations.
11. High-fidelity prototypes, mockup, wireframe.
12. Requires some time
13. Didn't have any troubles.
14. 4
15. I felt like ProtoShare is a bit difficult. I had to search lot of information to find out what features it has. It's definitely not simple. For some reason I think this is the tool for developers.

**UI design tool (name, version, evaluation date):**

**UXPin, 8.7.2014**

1. Web page & UI (Toolbar).
2. I think it's all there what I need.
3. After a while yes.
4. 2
5. It takes some time to learn how to use it properly.
6. 5
7. Tutorial docs with step-by-step and pictures.
8. 5
9. Yes.
10. Lot of interactions, events, actions.
11. High-fidelity prototypes, mockup, wireframe.
12. Requires some time
13. Yes. At first I had problems to upload images, but after a contact to Labranet Helpdesk everything worked fine.
14. 5
15. It feels the same as Proto.io or Easel.