

An Evaluation of Non-Code Mobile Application Prototyping Tools in a Lean UX Process.

Ingar Røsok Kvalheim
Department of Product Design
Norwegian University of Science and Technology

ABSTRACT

Lean UX is the newest design process regarding software development. In a lean UX process a minimum viable product(MVP) is a key element in order to find out if the assumptions made are right or wrong, and thereby will determine if the development process should continue in the same direction or not. How these MVPs are made depends on where in the development process the project find itself, but most of them are made by non-code prototyping tools. This article studies how suited existing non-code mobile application prototyping tools are for lean UX by reviewing eight such tools. For all the tools to be reviewed on equal terms evaluation requirements are made. These will be based on an argumentation for what constitutes a good prototyping tool within the lean UX context. After the review there is a discussion where trends, flexibility, coding and experiences from using some of the mobile prototyping tools in a mobile application development project with Skanska are being discussed. These discussions explore possible improvements and developments of prototyping tools and lean UX. The results conclude that there is no prototyping tool being optimal for Lean UX and that UXPin is the best alternative amongst those evaluated. Implementation of features making interaction between users possible and features making it possible to use factors like time, position and history of use are suggested as improvements. It is also a suggested that quality code should be a possible output for the prototyping tools to be optimal for lean UX.

KEYWORDS: Lean UX, non-code mobile application prototyping tool, user experience, minimum viable product, prototyping, mobile application.

1. INTRODUCTION

After the introduction of smartphones mobile applications have become a central part of our daily life. During 2014 the number of mobile users surpassed the number of desktop users in the world [1]. The same year US adults started spending more time on digital media through mobile than through desktop/laptop [1]. In fact they spend an average of 10.1% of their time on mobile devices per day and the phrase “there is an app for that” has become well known in our globalized world [2]. A phrase that gets supported by the 1000 new apps submitted to

the App Store everyday [3]. Today a good app idea alone isn’t enough. You also need to offer a great user experience(UX) for your application to stand a chance in the market. It’s a market that is rapidly changing with constantly introductions of new trends and technologies. This gives an enormous pressure on app development teams. They need to develop applications fast in order for the application to not be outdated when it hits the market or to not be outdone by similar apps reaching the market first. They also need to develop fast because of the big demand from clients, the world is getting digitalized and more

and more companies want their products or services to be available in digital versions on desktop and smartphones. The usage of mobile web accounted for 51% of Google's traffic at the end of 2014 [4]. Today people expect websites to be mobile-friendly, looking good and fully functional on a mobile device, and Google has also announced that these sites would be prioritized in their search results on the mobile web [5]. As a result of the increasing usage of mobile web the "mobile first" approach emerged. It means designing an online experience for mobile before designing it for the desktop [6]. Also this giving enormous pressure on development teams having a lot of clients wanting to stay relevant. In order to cope with this new development processes have emerged. For UX design, lean UX is the newest user-centered design process regarding software development. The process has been discussed through online publications and two books [7] [8], but most information is found in blog posts, discussions in forums online and video recordings from conferences etc.

This article will study the use of non-code mobile application prototyping tools in regard of being used in a lean UX process. Today prototyping is crucial to communicate the idea to stakeholders, but it is often a time- and cost consuming process and when the end product is finished the prototype loses its value [9]. There is also a problem with designers building prototypes looking perfect to impress clients. But in the end the developers are not able to implement the design, resulting in designers doing a lot of work that goes to waste and clients that are disappointed, not getting what they had expected [9]. These are problems that the lean UX process is intended to solve, but how well suited are the existing non-code mobile application prototyping tools for lean UX? This is the central question this article will answer by conducting a review of eight such tools. For a better understanding of the review some central terms regarding the lean UX process and its relation to prototyping needs to be explained first.

1.1 UX and UX design

User experience(UX) is a person's perceptions and responses that result from the use or anticipated use of a product, system or service [10]. Hence UX design is the process of enhancing user satisfaction by improving the usability, accessibility, and pleasure provided in the interaction between the user and the product [11]. Utility is also a key element, if the target group don't find the product useful there is no need for the product to exist. In the lean UX process prototypes are made early with the purpose of testing out if the users see the utility of the product. The user experience would not be good if the application runs slow either. This is the responsibility of the developer, but could come as a consequence of bad design. That's why it's important with close collaboration between these two roles throughout the process, which is another key element in the lean UX process.

1.2 Design thinking

Lean UX uses the design thinking approach to understand what to develop. Design thinking was made famous by Tim Brown and he describes it as follows: "as a style of thinking, it is generally considered the ability to combine empathy for the context of a problem, creativity in the generation of insights and solutions, and rationality to analyze and fit solutions to the context." [12]. In other words it's first about understanding what problem that should be solved for the user. This is by many called the discovery phase. Then start generating a lot of ideas to solve the problem in something called the ideation phase. Lastly it's time to analyze which of the ideas really solves the problems, this is done by experimentation and testing. Before the final concept is chosen, improved and launched.

1.2 Agile development

Lean UX is used in combination with central parts of agile development processes. Agile development is an iterative process for software development. It came as a result of trying to improve the traditional software development

process, as for example waterfall where things are put together in sequential order by different departments or teams and was all dependent on having a defined product idea before starting the development [13]. With agile development this is not necessary. The process is divided into sprints of two to four weeks where a cross functional team works on a delivery. After the sprint it's time to evaluate the direction of the project, making it possible to steer in another direction [14]. The mindset became popular after some developers released what they called "Manifesto for Agile Software Development" in 2001. Here quoting their key thoughts:

"we 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

That is, while there is value in the items on the right, we value the items on the left more." [15]. The process did not function that well co-working with designers. So Lean UX has adopted a lot of the agile process, but made it more efficient in regard to the collaboration between designers and developers.

1.3 Lean startup

Lean UX is based on the principles from the term lean startup, a methodology made famous by Eric Ries. It's about reducing time, cost and risk associated with starting up a company. The main thoughts originates from lean manufacturing made famous by Toyota. They removed everything in their development process that didn't give any value to the end user.

With lean startup the key element is to make hypotheses about your product or service and find the fastest way to validate or invalidate these hypotheses. This way you will not be using a lot of time and money on something that in the end won't satisfy the customers' needs. It's about testing instead of only assuming, and thereby learning as fast as possible and make

improvements based on your learnings. By doing this constantly during the development process it's possible to all the time evaluate that what is built is right. And what is built to test this is something called a minimum viable product(MVP), a version only containing the core elements needed to test the hypotheses [16] [17] [18].

1.4 Lean UX

Summarized Lean UXs purpose is to make good user experiences. Lean UX uses the mindset from design thinking in combination with central methods from agile development and it's based on principles from lean startup. The term was first introduced to the design community by Jeff Gothelf through the article "Lean UX: Getting out of the deliverable business" published on smashingmagazine.com in 2011 [19]. Gothelf and his team had been trying to do design in an agile development process which had resulted in a lot of failing and learning. They began to write about it and this became the foundation for the article and later the book "Lean UX: Applying Lean Principles to Improve User Experience" written by Gothelf and Josh Seiden and published in 2013 [8] [20]. Gothelf describes lean UX this way: "It's the practice of bringing the true nature of a product to light faster, in a collaborative, cross functional way with less emphasis on deliverables and greater focus on a shared understanding of the actual experience being designed." [21]. A shared understanding of what is being built and why it is built this way will according to Gothelf result in a more efficient development process. It's a process with a minimum of documentation where long detailed design cycles are eschewed in favor of very short, iterative, low-fidelity cycles, with feedback coming from all members of the implementation team early and often. Collaboration with the entire team becomes critical to the success of the product [20]. As a result of this Gothelf meant that some mindset needed to change. The designer has to be very open about showing unfinished sketches and prototypes, and the designer needs to become more of a design facilitator letting non-designers participate in the

design practice [8]. It's also important that everybody involved in a lean UX process accept the position of humility. Meaning that they accept the fact of not knowing what the result would be like, and that assumptions have to be tested in reality before treating it as the truth [20].

In January 2014, the Internet community produced a Lean UX manifesto [22] describing the Lean UX way of working in contrast to the old "standard" model:

- Early customer validation vs. releasing products with unknown end-user value
- Collaborative cross-functional design vs. lonely hero design
- Solving user problems vs. adding cool features
- Measuring key performance indicators vs. undefined success metrics
- Applying appropriate tools flexibly vs. following a rigid methodology
- Nimble design vs. heavy wireframes or specifications

1.5 Software prototyping and tools

A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from [23]. It exists both code-based prototyping tools and non-code prototyping tools. The ones reviewed in this article are all non-code tools. These tools are intended for UX designers because most UX designers can't code. Within this category of prototyping tools there exists two variants, page-based and layer-based. Page-based is easy to use and best for showing the flow between the screens. They are put together by linking different spots on a page to other pages and do not offer fine-tuned animations or complex features. Layer-based on the other hand can have multiple layers on a single page, dynamic states within a single layer, conditionals for any element, and detailed control for time-based animations. But the higher level of complexity also leads to a more time consuming prototyping process [24].

In lean UX the prototype is, as for lean startup, a MVP. The critical question when making this is to ask "what's the smallest thing we can make or do to test our assumptions?" By making the right MVP it's possible to test the value of the idea and the value of different features. The sooner this is done, the sooner it's possible to focus the resources on the best solution. Rapid prototyping can significantly reduce costs and risks. For example building a whole mobile app from scratch and then finding out that the main feature of the app isn't what the users wanted is costly [25]. A MVP is also valuable when it comes to making a shared understanding of what is being built. It's not always easy to understand ideas when only getting them told.

When prototyping in the early stages of a development process it's normal to make low fidelity prototypes that do not look polished but are easy to make. The key element is to not spend a lot of time making it look "perfect". This is because you will probably need to change a lot of it based on the feedback you get from testing it. If you've used a lot of time making it look good it's harder for you to acknowledge that changes needs to be made, and in worst case you end up not changing it as much as you should [26]. Some also say that it's harder for test-users to criticize something that looks finished. Both because they feel that it's too late for their comments to matter and that they will not hurt the people that have used a lot of time making the prototype, but this theory is not backed up by science [27] [28]. For this type of low fidelity prototyping normal tools are digital wireframing tools, or pen and paper to make something called a paper prototype. As the product starts to take shape prototypes of higher fidelity can be created. Things like gestures, animations, transitions and visual elements are often added. A lot of tools are available for these tasks offering a lot of features making the prototype feel very realistic. New tools are also constantly emerging, offering new workflows and new features, making it hard for UX designers to be up-to-date on what's available in the market.

2. METHOD

This article is based on a literature review on the topics lean UX and non-code mobile application prototyping tools. The sources are journal articles, books, blog posts, information from websites and discussions in forums online and video recordings from conferences etc. The reason why a lot of the sources are not peer reviewed is because lean UX and the prototyping tools are constantly evolving and a lot of last year's sources are already outdated. This leading to a question of validity of some sources. These are typical best practice reports from the business world limited to anecdotal references and company websites promoting their products. This article is written with this in mind, trying to select sources with an objective view.

As explained the main purpose of this article is to give an overview over how well suited the existing non-code mobile application prototyping tools are for lean UX. To do this a review of eight such tools will be conducted. For all the tools to be reviewed on equal terms evaluation requirements are made. These will be based on an argumentation for what constitutes a good prototyping tool within the lean UX context. After the review there is a discussion where trends, flexibility, coding and experiences from using some of the mobile prototyping tools in a mobile application development project with Skanska are being discussed. These discussions explore possible improvements and developments of prototyping tools and lean UX. Finally there will be a conclusion to round up the study which has been carried out.

2.1 Evaluation requirements

This article has chosen requirements based on the Lean UX manifesto mentioned in section 1.4, since this manifesto is relatively new and based on the feedback of many UX designers working with lean UX. To every key point in the manifesto follows an explanation to put it in context with prototyping tools. Based on the explanations different requirements have been suggested and these will be formed like questions to be used in the review.

Manifesto: Early customer validation vs. releasing products with unknown end-user value.

Explanation: It's important to get customer validation or fail early to not waste time and money. To do this the customer needs to have the opportunity to try out and test the MVP.
Requirements: The MVP should be shareable. The MVP should be realistic enough for the test to give useful feedback.

Manifesto: Collaborative cross-functional design vs. lonely hero design.

Explanation: To create a feeling of ownership to the project it's important that everybody in the team have the possibility to contribute in the design phase as well, even though they are not designers. New ideas and point of views will often result in a better design.
Requirements: It should be possible to collaborate on the creation of the MVP. It should be possible to comment on the MVP. It should be possible to use built-in design elements so that people without design skills still can visualize their thoughts.

Manifesto: Solving user problems vs. adding cool features.

Explanation: In almost every case a customer buy a product because of the problem it solves and not because of the additional cool features it provides. To not waste resources only the core features that solves the problem should be developed.
Requirements: It should be possible to add annotations to different parts of the design, so that it's clear which problem it's suppose to solve.

Manifesto: Measuring key performance indicators vs. undefined success metrics.

Explanation: In lean UX success is based on the outcome of the product. The different MVPs are built in hope of getting specific outcomes. To measure the outcome running a test and collecting feedback are very helpful.
Requirements: It should be possible to do user testing inside the prototyping tool. In regard to

this it should be possible to track the actions of the user and it should be possible for the user to give feedback.

Manifesto - Applying appropriate tools flexibly vs. following a rigid methodology.

Explanation - It's often more efficient to only use one or a few prototyping tools throughout the developing process. This is because people develop efficient workflows and habits using the same tool over and over again.

Requirements: It should be possible to both make low-fidelity and high-fidelity MVPs. It should be possible to output the MVP in usable code for further development.

Manifesto - Nimble design vs. heavy wireframes or specifications.

Explanation - Using a lot of time making specialized design and complex MVPs should be avoided when it's not the MVP that the user will buy in the end.

Requirements: The prototyping tools should have built-in design elements. It should be possible to reuse design.

To better evaluate the prototyping tools the requirements have been formed into questions. Similar requirements have been merged together. The questions are as follows:

Q1. How realistic can the MVP be?

Q2. How easy is it to collaborate on the creation of the MVP?

Q3. How easy is it to comment and make annotations on the MVP?

Q4. How easy is it to do user testing with the MVP?

Q5. How flexible is the prototyping tool?

Q6. How easy is it to get usable code out of the prototyping tool?

3. PROTOTYPING TOOLS REVIEW

The non-code mobile application prototyping tools selected to be reviewed in this article are the ones mentioned most times in recent articles and blog posts found on the internet in conjunction with the writing of this article [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39]

[40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50]. When searching through the articles and blog posts over 60 different prototyping tools was mentioned. The tools chosen are as follows: Proto.io, Marvel, InVision, Justinmind, UXPin, Pixate, Balsamiq and Axure RP. The first six tools are relatively new and have had their breakthroughs during the recent three years. They have managed to get special traction and publicity in a situation with lot of competitors coming through at the same time. The last two tools mentioned are veterans in the industry respectively started in 2008 and 2002. They have managed to stay relevant and it seems to be unavoidable to mention them when listing available prototyping tools in articles. The review of the tools will be written as answers to the questions formulated in section 2.2. At the end of the answer there will be given a rating on the scale from 1 to 5, where 5 is the best rating. Lastly there will be an analysis of all the results.

3.1 Proto.io

Proto.io is a layer-based prototyping tool that runs through a web browser.

Q1 - Realistic: Proto offers a big variety of interactions and animations that can be controlled into detail. There are built-in native elements for both iOS, android and windows. Own content can also be imported. It's possible to get the MVP on a mobile device [51]. Responds to different kind of inputs from users [52]. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *4*

Q2 - Collaboration: It's possible for the owner to give other people access to editing and creating projects [53]. People can also only get access to review MVPs with the possibility to make comments and annotations while seeing the creation of the MVP in real-time [54]. Built-in elements, as explained, makes it easier for non-designers to design. *5*

Q3 - Comments: It's possible to add comments to screens, elements or anywhere in the work area when inside comment mode. Possible to select persons to notify when making a comment and to mark comments as active or completed. Overview of comments in a left pane [54]. *5*

Q4 - User testing: It's possible to publish a MVP and make it public. By sharing the link to the MVP people can use it both on desktop and mobile. People can also make comments [55]. *3*

Q5 - Flexibility: It's possible to both use built-in elements and import own elements. Not possible to create elements inside the tool. Possible to duplicate screens and containers [56]. Templates exists for a head start [57]. Syncs with Dropbox and has plugins for Sketch and Photoshop [58]. Let you import both image, audio and video [59]. Proto is only available when connected to the internet. *4*

Q6 - Code: It's possible to export the MVP to HTML. The MVP pages are rendered by the Proto.io rendering engine which means that modification of the code may break the presentation and functionality of the MVP [60]. Seems to be of little value. *1*

3.2 Marvel

Marvel is a page-based prototyping tool that runs through a web browser or a mobile app [61].

Q1 - Realistic: Marvel offers some standard transitions and interactions. Possible with layers. No built-in elements are available, all the design are imported into the tool and the quality depends on its creator. It's possible to get the MVP on a mobile device. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *3*

Q2 - Collaboration: It's possible to collaborate on the creation of the MVP by making teams [62]. No built-in elements for those with limited design skills. *4*

Q3 - Comments: It's possible for both the team and others that have a link to the MVP to make comments to the MVP in general and add annotations to specific places [63]. *5*

Q4 - User testing: It's possible to do user testing with the MVP, but it's done with the help of another program called lookback and only available for iOS projects. Possible to record the test users screen, taps and gestures and also possible to see their reactions using the camera and microphone on the device [64]. *4*

Q5 - Flexibility: Only possible to make MVPs by importing content. Possible to use the smartphone to draw screens or snap a photo inside the marvel app. Possible to duplicate screens and links. Syncs with Dropbox and Google Drive and has support for Sketch and Photoshop. Let you import images and embed audio and video. Marvel is only available when connected to the internet. *3*

Q6 - Code: Not possible to get any valuable code out of a page-based MVP. *0*

3.3 InVision

InVision is a page-based prototyping tool that runs through a web browser.

Q1 - Realistic: InVision offers a variety of transitions, interactions and animations [65]. No built-in elements are available, all the design are imported into the tool and the quality depends on its creator. Possible to put together a screen consisting of different imported content [66]. It's possible to get the MVP on a mobile device. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *3*

Q2 - Collaboration: It's possible to collaborate on the creation of a MVP. With the feature LiveShare it's also possible to have in-browser meetings where you see everyone's pointer with their name attached while you can hear their voices. Possibility to draw and comment when

inside a meeting [67]. No built-in elements for those with limited design skills. *4*

Q3 - Comments: It's possible for people with access to the MVP to make comments [65]. They can be placed anywhere on the screen. Possible to tag people in the comments. An overview of all comments are found when clicking the comments tab [68]. *5*

Q4 - User testing: It's possible to do user testing with the MVP. But, as for Marvel, it's done with the help of another program called lookback and only available for iOS projects. Possible to record the test users screen, taps and gestures and also possible to see their reactions using the camera and microphone on the device [65]. *4*

Q5 - Flexibility: Only possible to make MVPs by importing content. Possible to design own screens by using different imported content. Integrates with Dropbox, Google Drive, Slack, Photoshop, Sketch and more [65]. Let you only import images. InVision is only available when connected to the internet. *2*

Q6 - Code: It exists a beta version where it's possible to create stylesheets and generate real code for any design element [65]. But it's not yet for public use. *0*

3.4 Justinmind

Justinmind is a layer-based prototyping tool that runs through a software. Available for windows and iOS [69].

Q1 - Realistic: Justinmind offers a large amount of interactions, transitions and animations. There are built-in native elements for both iOS, android and windows. Own content can also be imported. It's possible to get the MVP on a mobile device [70]. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *4*

Q2 - Collaboration: It's possible for the entire team to edit and make changes simultaneously

on the same MVP as long as they are online. Version history is available and it's possible to work on a shared MVP in offline mode [70]. Built-in elements are available. *5*

Q3 - Comments: It's possible for everybody with access to the MVP to make comments. These can also be organized in categories. Comments made offline will be synced when online [70]. *5*

Q4 - User testing: It's possible to publish a MVP and make it public. By sharing the link to the MVP people can use it both on desktop and mobile. People can also make comments [70]. *3*

Q5 - Flexibility: It's possible to both use built-in elements and import own elements. Not possible to create elements inside the tool. Possible to reuse style customization and to use masters. Templates exists for a head start. Integrates with Photoshop and Illustrator. Let you only import images [70]. Available both online and offline. *4*

Q6 - Code: It's possible to generate an interactive HTML+JavaScript MVP [70]. Since it's a layer-based prototyping tool it could be of little value. *1*

3.5 UXPin

UXPin is a layer-based prototyping tool that runs through a web browser.

Q1 - Realistic: UXPin offers interactions, transitions and animations that can be controlled in detail. Possible to use built-in elements and also possible to import own content. It's possible to get the MVP on a mobile device. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *4*

Q2 - Collaboration: It's possible to work with teammates on the same projects and design in real-time. Also possible to present projects in real-time with screen sharing, cursor tracking,

and voice calling [71]. Built-in elements available. *5*

Q3 - Comments: It's possible for everybody having access to the MVP to comment and make tasks with a built-in checkbox [71]. *5*

Q4 - User testing: It's possible to create user testing sessions where you share the MVP with user groups, and record their experience. The recordings include insights from screen, webcam recording and voice calling [71]. Not on smartphone. *4*

Q5 - Flexibility: It's possible to both use built-in elements and import own elements. Not possible to create elements inside the tool. Elements and their interactions can be grouped into smart elements and easily be reused. Integrates with Photoshop and Sketch. Let you only import image-file types. Only available when connected to the internet. *4*

Q6 - It's possible to export the MVP to an HTML version. Seems to be of little value. *1*

3.6 Pixate

Pixate is a layer-based prototyping tool that runs through a software. Available for windows and iOS.

Q1 - Realistic: It's possible to use and make complex animations, transitions and interactions. Only possible to import own content. It's possible to get the MVP on a mobile device [72]. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *3*

Q2 - Collaboration: It's possible to add users with the role "designer" to the prototyping project. Users with this role can create, delete and edit the MVP [73]. *5*

Q3 - Comments: It's possible for "designers" to make comments. It's also possible to add users with the role "collaborators" that are only allowed to watch and comment [73] [74]. *5*

Q4 - User testing: It's possible to share a link so others can visit to preview your MVP on their device. The Pixate mobile app is required to preview MVPs [72]. No further features for user testing are provided. *2*

Q5 - Flexibility: Only possible to make MVPs by importing content. Possible to design own screens by using different imported content. Let you only import image file types. Not dependent on internet connection. *2*

Q6 - Code: Not possible. *0*

3.7 Balsamiq

Balsamiq is a page-based prototyping tool that can run both through a web browser and a desktop app [75].

Q1 - Realistic: Balsamiq do not offer animations or transitions. Only mouse-click interaction. Not possible to use the MVP on a smartphone. Possible to use built-in elements. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *2*

Q2 - Collaboration: When running the online version of balsamiq it's possible to add project members that have the ability to comment and edit the MVP being made. It's also possible to change the access to the MVP so that everybody with the link can comment and edit [76]. Built-in elements are available for those without design skills. *5*

Q3 - Comments: As explained comments can be made by both project members or anonymous viewers based on what kind of access the project has [76]. *5*

Q4 - User testing: Not possible for users to test the MVP on a smartphone. When using the online version of Balsamiq it's possible to share the MVP through a URL and to get feedback in form of comments. *3*

Q5 - Flexibility: Only possible to use built-in elements or elements added from the community site. Templates, masters, and component libraries can be created and reused [77]. Can be used both offline and online. *2*

Q6 - Code: Not possible. *0*

3.8 Axure RP

Axure is a layer-based prototyping tool that runs through a software. Available for windows and iOS [78].

Q1 - Realistic: Axure offers the possibility of using a lot of interactions, animations and transitions that can be controlled in detail. There are built-in native elements. Own content can also be imported. Responds to different kind of inputs from users. Not possible to interact with other users when using the MVP. Not possible to make use of factors like real time, position and history of use. *4*

Q2 - Collaboration: It's possible to use "Team Projects" to work with other team members on a project. It uses a shared network drive or an SVN server for remote access [79]. It's also possible to use AxShare, an online service, to share MVPs with others [80]. *5*

Q3 - Comments: It's possible to make notes both on elements and to pages [81] [82]. Comments can also be made by users having access to the MVP in AxShare [82]. *5*

Q4 - User testing: As explained it's possible to share the MVP through AxShare, which can be opened both on mobile and web. Comments to the MVP can be made [82]. *3*

Q5 - Flexibility: It's possible to both use built-in elements and import own elements. Not possible to create elements inside the tool. Possible to use masters and templates. Not dependent on internet connection. *3*

Q6 - Code: It's possible to export the MVP to an HTML and JavaScript version [79]. Seems to be of little value. *1*

3.9 Results

Final score:

Proto.io: 22	Marvel: 19	InVision: 18
Justinmind: 22	UXPin: 23	Pixate: 17
Balsamiq: 17	Axure RP: 21	

As we see from the score UXPin is the mobile prototyping tool best suited for being used in a lean UX process. In this review all the questions have equally importance, but it could be that some of the questions are more important than others. This is something this article don't look into. It could also be that not all the possibilities of the mobile prototyping tools are explained through the literature found, probably there exist features in some tools that have an impact on the scoreboard. It is also hard to tell the ease of use of some of the features by just reading about them, this is something only firsthand experience can tell. When it comes to the requirement of reusable code this is extra hard. There are a lot of different ways to code something, resulting in different quality of coding. It's also hard to tell if the code is detailed enough for each element for it to be usable in further development.

4. DISCUSSION

4.1 Trends

One of the biggest trends in the world of UX is anticipatory design, also called context awareness design. It's about understanding the full story around a human experience, in order to bring users exactly what they want, with minimal interaction [83]. To anticipate what the user wants the product needs to be smart enough to understand the user context, the environmental context and the world context. Take the music streaming Spotify for example. By analyzing history of use Spotify gives you a list of new music every week that you most probably would like. Google is another company with a lot of anticipatory design in their applications [84] [85],

and a large amount of companies are following in their footsteps.

Another trend that has been around for a while now, so long that it in the digital world might no longer count as a trend, is applications being “social”. This means there is some kind of possibility of interaction between users in the app. Facebook, Snapchat and Twitter are the biggest and clearest examples, totally dependent on shared content between users. There are an enormous amount of apps with social features on the market. The latest trend in social apps are those based on the trend of shared economy, like Uber and Airbnb, which easily explained is based on sharing of resources. More of these apps are emerging every day just making the ability of testing social features even more important.

Today no non-code mobile application prototyping tools gives the opportunity to make MVPs with anticipatory design or social interactions. These are factors that are becoming more and more crucial for applications, and some apps are totally based on the use of these trends. Without these features the prototyping tools won't in a lot of cases be able to create realistic MVPs. I think making a prototyping tool with the ability to implement social features and anticipatory design would be an important asset when it comes to validate the value of an idea in a lean UX process.

4.2 Flexibility

None of the reviewed prototyping tools got a maximum score on flexibility. There is no tool that is perfect for both easy page-based prototyping and more complex layer-based prototyping. None of the tools supported creating own elements inside the program, just importing own design created in another program or using built-in elements. And only some tools had features for running and collecting feedback from user testing. In other words there is no “to do it all” tool. Using a lot of different programs during the lean UX process is challenging in regard of collaboration. For team

members not originally doing design it's a lot to keep track of. Remembering how each program worked when only using it occasionally. But it seems that this kind of prototyping tool could be coming already in 2016 with the launch of Adobe's new software Project Comet. Referred to as “the first all-in-one solution for UX designers” by Adobe themselves [86].

4.3 Coding

There were no prototyping tools in the review that promoted native code as one of their output features. Some of the tools offered the possibility of exporting the MVP to a HTML, CSS and JavaScript version, but this seemed to be a highly specialized code with no possibility for adjustments. For a page-based MVP this would only be code containing transitions from page to page with no detailing, so either way the code would be worthless. HTML, CSS and JavaScript works for putting together an app with phonegap, working on multiple platforms [87]. But a lot of apps are developed with native android or iOS code and no of the tools offers this kind of output. As explained lean UX process tries to be as efficient and resource reducing as possible by only focusing on the things giving value to the end user. A MVP gives a lot of value to the end user in regard to finding the core features and best usability, but it's not efficient to build the end product from scratch when a highly realistic MVP exists. As long as the prototyping tools don't outputs valuable code there will be a point where start coding is more efficient than using time on a non-code MVP. I think a further research on this topic would be interesting. While the optimal solution I think is for non-code mobile application prototyping tools to able to generate valuable code. Another interesting question that emerges in relation to this topic is; should designers using a lean UX process be able to code? If it's most efficient to start coding early in a prototyping process it would sound great if designers could take some of the load off the programmers' shoulders. But as Jesse Weaver states in an article on medium “Let's get real here. Design and development (both front end and back end)

are highly specialized professions. Each takes years and countless hours to master. To expect that someone is going to become an expert in more than one is foolhardy.” [88] In other words learning both design and development could be too much to ask for and lead to people not being particularly good in any of the fields. “I believe we need designers that have enough knowledge about coding to understand and respect the challenges and the mindset.” - Trond Are Øritsland [89]. This could be a better answer. With designers who understands a bit of coding the chances for making perfect prototypes impossible for programmers to develop would be reduced. And most probably it would in general ease the collaboration between programmers and designers.

4.4 Experience from the Skanska project

I have in parallel to this article worked on a mobile application development project with the building company Skanska. Through this project I have tried out all the prototyping tools reviewed. Even though this was not a lean UX process I have gotten some valuable experiences in relation to the topic. In the end it was decided that the application should offer the possibility to view and interact with a 3D model. This was not possible to do in any mobile prototyping tool, and a simplified presentation of this feature was made in proto.io which was the chosen prototyping tool for this last high-fidelity MVP. With virtual reality(VR) and augmented reality(AR), based on 3D technology, being on the verge of a breakthrough 3D features could become necessary for prototyping tools in order to stay relevant in the future [90]. Another interesting thing I noticed was that the people I worked with in Skanska wanted a MVP that looked polished before they were comfortable to show it and let other people test it. So for lean UX to be really successful it's not only the designer who needs to be open about showing unfinished sketches and MVPs, but it's at least equally important that the rest of the team and stakeholders are comfortable about it so that valuable feedback can be gathered.

5. CONCLUSION

This article has studied the use of non-code mobile application prototyping tools in regard of being used in a lean UX process. A review of eight such tools has been completed with the purpose of giving overview over how well suited they are for lean UX. The results show that there is no prototyping tool that is optimal for Lean UX and that UXPin is the best alternative amongst those evaluated. The review revealed that there are a lack of features in regard to making highly realistic MVPs. To achieve this more social features and features that made it possible to use factors like real time, real position and history of use is proposed. From the parallel project with Skanska I also found a lack of ability to implement 3D models into the mobile prototyping tools. During the project I also discovered the importance of openness for sharing unfinished MVPs.

In the introduction two problems were highlighted. The first problem was that the prototype loses its value after the building of the end product starts. In this article it has been argued that the prototype gives a lot of value to the end user before it is no longer needed, and it has been suggested that for the prototype to be even more valuable it should be able to output quality code so that the building of the end product won't start from scratch. The second problem was that designers use a lot of time making prototypes with features and design that developers aren't able to implement. Heavy collaboration between programmers and designers is lean UXs answer to this problem, for this collaboration to work it has been suggested in this article that designers should have a basic knowledge about coding.

Further work on this article would be research on the importance of each element in the lean UX process, this way an even better review could have been conducted. And also research on when it's most efficient to stop non-code prototyping and start with real coding instead. These are topics I would suggest for further studies.

REFERENCES

- [1] Bosomworth, D. (2015). *Mobile Marketing Statistics 2015*. Available from <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [2] Ramos, R. (2013). *Content Marketing: Insider's Secret to Online Sales & Lead Generation*, New York: One Night Expert Publishing.
- [3] Number of apps available in leading app stores 2015. (2015). Available from <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores>
- [4] Why Mobile Web Still Matters in 2015. (2015). Available from <http://www.clickz.com/clickz/column/2388915/why-mobile-web-still-matters-in-2015>
- [5] Takaki Makino, C. J., Doantam Phan. (2015). *Finding more mobile-friendly search results*. Available from <http://googlewebmastercentral.blogspot.no/2015/02/finding-more-mobile-friendly-search.html>
- [6] Graham, R. (2012). *Mobile First: What Does It Mean?* Available from <http://www.uxmatters.com/mt/archives/2012/03/mobile-first-what-does-it-mean.php#sthash.7UpwweNP.dpuf>
- [7] Klein, L. (2013). *UX for Lean Startups* (1st ed.). Sebastopol: O'Reilly Media, Inc.
- [8] Gothelf, J. (2013). *Lean UX: Applying Lean Principles to Improve User Experience* (1st. ed ed.). Sebastopol: O'Reilly Media, Inc.
- [9] Lassen, A. (2015). *How App Design Can Hurt App Development*. Available from <http://techcrunch.com/2015/07/26/how-app-design-can-hurt-app-development/>
- [10] International Organization for Standardization (2009). *Ergonomics of human system interaction - Part 210: Human-centered design for interactive systems*. Geneva, Switzerland: ISO/IEC.
- [11] Kujala, S., & Roto, V., & Väänänen-Vainio-Mattila, K., & Karapanos, E., & Sinnelää, A. (2011). UX Curve: A method for evaluating long-term user experience. *Interacting With Computers* 23 (5), 473–483.
- [12] Jez Humble, J. M., Barry O'Reilly. (2015). *Lean Enterprise: How High Performance Organizations Innovate at Scale* (1st ed.). Sebastopol: O'Reilly Media, Inc.
- [13] Bowes, J. (2014). *Agile vs Waterfall: Comparing project management methods*. Available from <http://manifesto.co.uk/agile-vs-waterfall-comparing-project-management-methodologies/>
- [14] Jongerius, P. (2012). *GET AGILE! Scrum for ux, design & development*. Amsterdam: B/S Publishers.
- [15] Manifesto for Agile Software Development. (2001). Available from <http://agilemanifesto.org/>
- [16] Blank, S. (2013). Why the lean start-up changes everything, *Harvard Business Review*, 91(5), 63-72.
- [17] Ries, E. (2011). *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses* (1st ed.). United States: Crown Business.
- [18] Lean. (2015). Retrieved September 09, 2015. Available from <http://no.wikipedia.org/wiki/Lean>
- [19] Gothelf, J. (2011). *Lean UX: Getting Out Of The Deliverables Business*. Available from <http://www.smashingmagazine.com/2011/03/lean-ux-getting-out-of-the-deliverables-business/>
- [20] Lean UX in the Enterprise: Agility through cross-functional collaboration with Jeff Gothelf. (2015). Available from <https://www.youtube.com/watch?v=5aDalsVKuIQ>
- [21] Jeff Gothelf - Workshop Lean UX au NUMA le 26/01/2015 (2015). Available from <https://www.youtube.com/watch?v=bnPISc2aQFI>
- [22] Viviano, A. (2014). *The Lean UX Manifesto: Principle-Driven Design*. Available from <http://www.smashingmagazine.com/2014/01/08/lean-ux-manifesto-principle-driven-design/>
- [23] Blackwell, A. H., & Manar, E., (2015). *Prototype. UXL Encyclopedia of Science* (3rd ed.). Published by: U·X·L.
- [24] Kim, B. (2014). *Comparing Popular Layer-Based and Code-Based Prototyping Tools*. Available from <http://uxmag.com/articles/comparing-popular-layer-based-and-code-based-prototyping-tools>
- [25] Odysseos, A. (2015). *10 Mobile App Prototyping Best Practices that Result in a Stellar Finished Product*. Available from <http://blog.proto.io/10-mobile-app-prototyping-best-practices-that-result-in-a-stellar-finished-product/>
- [26] Busche, L. (2014). *The Skeptic's Guide To Low-Fidelity Prototyping*. Available from <http://www.smashingmagazine.com/2014/10/t>

- he-skeptics-guide-to-low-fidelity-prototyping/
- [27] Low-fidelity vs high-fidelity: Do they elicit different types of reactions? (2013). Retrieved September 20, 2015. Available from <http://ux.stackexchange.com/questions/39693/low-fidelity-vs-high-fidelity-do-they-elicite-different-types-of-reactions>
- [28] Bailey, B. (2005). *Paper Prototypes Work as Well as Software Prototypes*. Available from <http://www.usability.gov/get-involved/blog/2005/06/paper-prototypes-and-software-prototypes.html>
- [29] Heaton, S. (2015). *5 of the Best Prototyping Tools to Test Out Your Web and Mobile Designs* Available from <https://www.shopify.com/partners/blog/62745923-5-of-the-best-prototyping-tools-to-test-out-your-web-and-mobile-designs>
- [30] Schwartzman, E. *Designer's Toolkit: Prototyping Tools.*, Retrieved October 15, 2015. Available from <http://www.cooper.com/prototyping-tools>
- [31] Best App Prototyping Tool - Proto.io vs. InVision vs. Marvel. (2015). Available from <https://www.enolalabs.com/blog/archives/best-app-prototyping-tool-proto.io-vs.-invision-vs.-marvel>
- [32] Crow, D. (2014). *Comparing 9 Mobile Prototyping Tools Every Designer Should Consider*. Available from <http://moneysummit.mx.com/the-summit/2014/12/10/9-mobile-prototyping-tools-every-designer-should-consider>
- [33] Gosha, G. (2015). *3 Prototyping Tools Compared: Invision, Marvel & Justinmind*. Available from <http://www.sitepoint.com/invision-marvel-justinmind-prototyping-tools-compared/>
- [34] Mat, T. (2015). *Five app prototyping tools compared Proto.io, Pixate, Origami, Framer & Form*. Available from <https://medium.com/sketch-app-sources/five-app-prototyping-tools-compared-form-framer-origami-pixate-proto-io-c2acc9062c61>
- [35] Mobile App Prototyping – 13 Solid Tools for Designing Your App. (2015). Available from <http://techbloke.com/mobile-app-prototyping-tools/>
- [36] Best Mobile App Prototyping Tools. (2015). Available from <http://digitalagencynetwork.com/best-mobile-app-prototyping-tools/>
- [37] App Prototyping Tools., Retrieved October 01, 2015. Available from <https://gist.github.com/stonehippo/2fbf5a6b3beb0ea2e3f8>
- [38] Weller, N. (2015). *The Top Rapid Prototyping Programs And Techniques For Designers*. Available from <https://www.elegantthemes.com/blog/resources/the-top-rapid-prototyping-programs-and-techniques-for-designers>
- [39] 19 Easy to Learn Mobile App Prototyping Tools. (2015). Available from <http://blog.venturepact.com/19-easy-to-learn-mobile-app-prototyping-tools/>
- [40] Prikaznov, A. (2015). *Best Resources For Creating and Prototyping Mobile Apps*. Available from <https://dzone.com/articles/best-resources-creating-and>
- [41] Wilcox, M. (2015). *Are prototyping tools becoming essential?* Available from <http://www.developereconomics.com/prototyping-tools-becoming-essential/>
- [42] Goff, J. (2015). *UX Prototyping: New Tools for Designers*. Available from <https://medium.com/@argodesign/ux-prototyping-new-tools-for-designers-6c863919e573#g8qyvpu9q>
- [43] Smith, G. (2013). *20 Excellent Wireframing Tools for Mobile*. Available from <http://mashable.com/2013/04/02/wireframing-tools-mobile/#Wsjk2Y5vk5qq>
- [44] What are the best tools for prototyping mobile interactions/animations?, Retrieved October 15, 2015. Available from www.slant.co/topics/1640/compare/~framer-studio_vs_pixate_vs_invision
- [45] Pedzai, C. (2012). *Axure vs Balsamiq prototyping*. Available from <http://uxdesignmastery.com/2012/07/09/axure-vs-balsamiq-prototyping/>
- [46] The Best Wireframe Tools & Software., Retrieved October 15, 2015. Available from <https://www.comparakeet.com/best-wireframing-tools/>
- [47] Benazzi, L. (2014). *Best prototyping tools out there? A review of Axure, Justinmind, and other UX tools*. Available from <http://www.humaneinterface.net/best-prototyping-tools/>
- [48] Daggi, K. *Prototyping Tools and Practices That Yield Maximum Benefits*. Available from <http://www.hongkiat.com/blog/prototyping-tools-practices/>
- [49] Ovsyannykov, I. (2015). *5 Excellent Prototyping Tools and Best Practices*. Available from [An Evaluation of Non-Code Mobile Application Prototyping Tools in a Lean UX Process.](http://inspirationfeed.com/articles/design-</p>
</div>
<div data-bbox=)

- articles/5-excellent-prototyping-tools-and-best-practices/
- [50] Complete Beginner's Guide to Interaction Design. (2015). Available from <http://www.uxbooth.com/articles/complete-beginners-guide-to-interaction-design/?ref=webdesignernews.com>
- [51] UI prototyping for the space age., Retrieved October 26, 2015. Available from <https://proto.io/en/tour/>
- [52] Introducing variables., Retrieved October 26, 2015. Available from <http://docs.proto.io/videos/introducing-variables/>
- [53] Users and roles., Retrieved October 26, 2015. Available from <http://docs.proto.io/documentation/dashboard/people-and-roles/>
- [54] Comments and annotations., Retrieved October 26, 2015. Available from <http://docs.proto.io/documentation/dashboard/people-and-roles/>
- [55] Publish your prototype., Retrieved October 26, 2015. Available from <http://docs.proto.io/documentation/publish-prototype/>
- [56] The editor., Retrieved October 26, 2015. Available from <http://docs.proto.io/documentation/editor/>
- [57] Project Templates., Retrieved October 26, 2015. Available from <http://docs.proto.io/videos/project-templates/>
- [58] Proto.io video academy., Retrieved October 26, 2015. Available from <http://docs.proto.io/tutorials-academy-beginners/>
- [59] Features galore!, Retrieved October 26, 2015. Available from <https://proto.io/en/features/>
- [60] Export prototype to HTML. Retrieved October 26, 2015. Available from <http://docs.proto.io/documentation/export-prototype-to-html/>
- [61] Marvel., Retrieved October 26, 2015. Available from <https://www.crunchbase.com/organization/marvel#/entity>
- [62] Mutlu, M. (2014). *You Can Now Collaborate On Prototypes With Your Team!* Available from <http://blog.marvelapp.com/you-can-now-collaborate-on-prototypes-with-your-team/>
- [63] Mutlu, M. (2015). *You can now add comments and annotations to your prototypes!* Available from <http://blog.marvelapp.com/you-can-now-add-comments-and-annotations-to-your-prototypes/>
- [64] Mutlu, M. (2015). *Introducing Marvel User Testing Tools for iOS!* Available from <http://blog.marvelapp.com/introducing-marvel-user-testing-tools/>
- [65] Design better. Faster. Together., Retrieved October 27, 2015. Available from <http://www.invisionapp.com/#tour>
- [66] Preview: Motion-advanced animations for web & mobile design., Retrieved October 27, 2015. Available from <http://www.invisionapp.com/new-features/45/preview-motion-advanced-animations-for-web-and-mobile-design>
- [67] LiveShare: In-Browser Design Collaboration., Retrieved October 27, 2015. Available from <http://www.invisionapp.com/new-features/18/liveshare-in-browser-design-collaboration>
- [68] The Redesigned Comments Tab., Retrieved October 27, 2015. Available from <http://www.invisionapp.com/new-features/17/the-redesigned-comments-tab>
- [69] Schwartzman, E. (2015). *Designer's Toolkit: Prototyping Tools - Justinmind*. Retrieved October 27, 2015. Available from <http://www.cooper.com/prototyping-tools/Justinmind>
- [70] Features., Retrieved October 27, 2015. Available from <http://www.justinmind.com/features#>
- [71] Wireframe, mockup & prototype. Together., Retrieved October 28, 2015. Available from <https://www.uxpin.com/prototyping.html>
- [72] Prototypes., Retrieved October 28, 2015. Available from <http://help.pixate.com/knowledgebase/articles/665659-4b-prototypes>
- [73] Managing your team., Retrieved October 28, 2015. Available from <http://help.pixate.com/knowledgebase/articles/665665-4c-managing-your-team>
- [74] Holt, A. (2015). *Communicating your designs just got better: Introducing Pixate Comments*. Retrieved October 28, 2015. Available from <http://blog.pixate.com/>
- [75] Life's too short for bad software!, Retrieved October 29, 2015. Available from <https://balsamiq.com/>
- [76] Sharing your work with outside clients in myBalsamiq., Retrieved October 29, 2015. Available from <https://docs.balsamiq.com/mybalsamiq/sharin>

- g/
- [77] Why Balsamiq Mockups for wireframing?, Retrieved October 29, 2015. Available from <https://balsamiq.com/products/mockups/>
- [78] Schwartzman, E. (2015). *Designer's Toolkit: Prototyping Tools - Axure*. Available from <http://www.cooper.com/prototyping-tools/Axure>
- [79] Introducing Axure RP., Retrieved October 29, 2015. Available from <http://www.axure.com/features>
- [80] AXSHARE BASICS - *Hosting and Sharing Projects on the Web.*, Retrieved October 29, 2015. Available from <http://www.axure.com/learn/axshare/basics>
- [81] WIDGET NOTES - Add Notes to Communicate., Retrieved October 29, 2015. Available from <https://www.axure.com/learn/documentation/widget-notes>
- [82] PAGE NOTES - Page Overviews & Custom Fields., Retrieved October 29, 2015. Available from <https://www.axure.com/learn/documentation/page-notes>
- [83] David, A. B. (2014). *Context design: How to anticipate users' needs before they're needed*. Available from <http://thenextweb.com/dd/2014/04/28/context-design-anticipate-users-needs-theyre-needed/>
- [84] Stenovec, T. (2015). *Google has gotten incredibly good at predicting traffic - here's how*. Available from <http://www.techinsider.io/how-google-maps-knows-about-traffic-2015-11>
- [85] Busche, L. (2015). *What You Need To Know About Anticipatory Design*. Available from <http://www.smashingmagazine.com/2015/09/anticipatory-design/>
- [86] A whole new experience in user experience design., Retrieved November 22, 2015. Available from <http://landing.adobe.com/en/na/products/creative-cloud/comet/229818-notifyme.html?scid=social52541916>
- [87] Traeg, P. (2014). *Four Ways To Build A Mobile Application, Part 3: PhoneGap*. Available from <http://www.smashingmagazine.com/2014/02/our-ways-to-build-a-mobile-app-part3-phonegap/>
- [88] Weaver, J. (2014). *We Don't Need More Designers Who Can Code*. Available from <https://medium.com/re-write/we-dont-need-more-designers-who-can-code-b81483d2a0e6#.ruy8vrsxp>
- [89] Øritsland T.A., personal communication (2015)
- [90] Møllevik, K. (2015). *Virtual Reality vil snart bli like vanlig som video*. Retrieved December 05, 2015. Available from <http://www.tek.no/artikler/disse-to-store-utfordringene-star-vr-bransjen-overfor/194031>