

Scrum in Mobile Media Design

Candidate no. 525174

MEVIT4640 – Mobile Media Design
Autumn 2011

*#Department of media and communication, University of Oslo
Norway*

Abstract— This paper is a documentation and reflection over a group project conducted in the course Mobile Media Design (MEVIT4640) at the University of Oslo. It is looking at the design process of an mobile web application called Småprat that are designed in order to answer the research question. “Can an application be used to enhance a conversation and is it socially acceptable to use?” With several iterations of designing construction and user testing the group created a prototype that could be tested in a social setting to answer the research question. In this project the designing was included in Scrum iterations, something it usually is not. The paper discusses the pros and cons of this and concludes one should focus on design throughout the whole process to get the best result. The paper also concludes that Donald Schöns concept of reflection-in-action can be compared with scrum iterations.

Keywords— Agile, Scrum, Mobile Design, Reflection-in-Action,

I. INTRODUCTION

This paper present the work a group project from the course Mobile Media Design, MEVIT4640 at University of Oslo. It will describe the act of doing design in general, mobile design in particular and the learning outcome from the work process. In addition to this it will discuss how Donald Schön's concept of reflection-in-action and the Scrum framework was used in the process and how our work was formed by our research question. The paper will first look at the design process, then discuss the learning outcome, design in scrum and reflection-in-action and ends with a conclusion.

II. DESIGNING SMÅPRAT

The purpose of the course was to come up with an idea and develop a prototype for a mobile web application. Through this work the students should learn about designing and developing in groups by using Scrum and acquire knowledge about mobile web standards like HTML, CSS and JavaScript and how to use these.

Mobile media is becoming more and more popular, and smart phones are becoming more prominent in daily lives. Some people even use their phone to get away from socially awkward situations. An American study[7] showed that 30% of 18-29 year old mobile users have used their phone to **avoid** interacting with the people around them. The group decided to develop a small talk application to investigate whether or not an application can be used to **enhance** a conversation rather than just to get away, and whether this is socially acceptable[8].

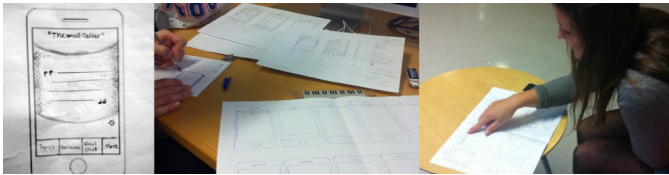


Fig. 1: Småprat-logo, group 1 MEVIT4640

The application was created in Norwegian and was called Småprat. It is an mobile web application containing questions that can be used as conversation starters, the questions are sorted into categories and activities. Users can add and remove favourites from the available questions, and last you can get phrases that can help you escape from an unpleasant situation. The research question for this project read as followed “How can an app be used to enhance a conversation, and is it socially acceptable?”

Testing was essential in this project. Not just in order to create an application that was intuitive and easy to use, but also to be able to answer the research question. This was something the group were aware of from the beginning of the project. Throughout the process 14 students from the department of informatics, divided on four iterations. We tested out low-fidelity wireframes, Photoshop sketches, did usability testing of the prototype on an iPhone, and tested it in a real-life setting at the department of informatics. In addition to this some

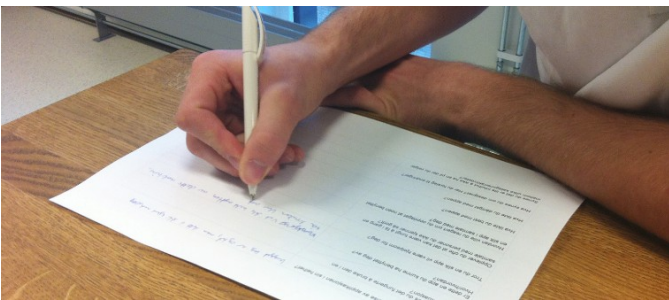
people were asked to answer a questionnaire and some participated in short semi-structured interviews.



Testing of wireframes, group 1, MEVIT4640



Usability testing, group 1, MEVIT4640



Questionnaire, group 1, MEVIT4640

A. Scrum

The Scrum framework that the project were asked to follow is a process framework used to manage complex product development Scrum rapidly surfaces dysfunction, and enables large teams and organisations to continuously improve their effectiveness[5]. It allowed its users to employ various processes and techniques, but with the rules of Scrum controlling that the events, roles, and artefacts are bound together and ensuring the relationship and interaction between them[1].

Fig. 2 visualises the Scrum process. First a vision is established, then all the requirements that should be fulfilled in the complete product is identified in a product backlog[2]. The process is divided into so-called sprints. The definition of what features that should be built in the sprint is what we call the sprint backlog. The long term plans were more fluid while the short term plans made for single sprints were more stable and detailed[3]. Each sprint can be considered as a smaller project, normally on 30 days where the team each day come together for a Daily Scrum where the activities for the next 24 hours are planned. The sprint should result in a potentially shippable product increment[1].

The Scrum is one of several agile methodologies, something we can recognise by the fact that it is both adaptive and people-oriented. The different roles that exists in scrum is the product owner, Scrum Master and The Scrum Team. As this was a rather small project and no-one had much experience with scrum, we did not separate between these roles within the group, and rather worked together as a self-driven scrum team. Working on this project, which was only 10 credits we had to adapt the process so that it fitted our schedule. The project had four sprints, each on two weeks, and the group worked together about two days a week instead of everyday. Before every work session started the group had a daily scrum meeting where everyone were updated on the status of the project and the next step were clarified. After each sprint we reviewed our work and had a supervision session with the lecturer of the course where the progress were discussed and the group got guidance on further development.

III. LEARNING OUTCOME

In this project we have learned about the scrum framework and how to conduct a group project in scrum. We have worked together as a team and seen the importance of communication and internal meetings. Group members got responsibility for the different features and task based on interests and skills, some did more developing while others worked more in the design and user testing. However we mostly did the work while we were in the same room, which made it possible to keep each other updated with regular meetings and consultations, and in that way we learned from each other and everyone was included in the whole process.

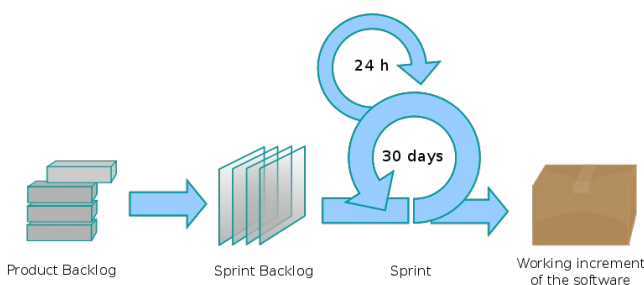


Fig. 2: The Scrum Process (Available: <http://yomo.no/tjenester/utvikling/> loaded 09.11.11)

A. Design

It was interesting to see how the user testing affected our final product. Early in the project the group made a lot of assumptions about the use of the application and its features which was the base for the first low-fidelity wireframes. Fowler[3] mentions that what elements are valuable and which once aren't is often not obvious before the product have been tested. But we quickly learned that you also can get valuable feedback from low-fidelity prototypes on an early stage. The first tests provided feedback on the concept and the placing and labelling of different elements. And the feedback we got from the user tests guided us to our next move, the design were no longer only built on assumptions, but on actually user behaviour and needs. As the prototype evolved the type of feedback also changed. Simple low-fidelity sketches provided good general feedback on the concept, structure and navigation of the application while high-fidelity prototypes with a higher level of details resulted in more feedback on details in the graphical interface and the actually use of the application.

The testing proved that some of our assumptions were correct and some of them were not. In the first sketches it was an feature where the users could contribute and add questions to the application, however it it was removed when it turned out that the users didn't see the point of this feature. The notes feature that would enable users to plan phrases and questions in advance was not rated as very important, and when we needed to make a cut in the backlog this was one of the features that were removed. During the usability testing a user stated that "I would probably just use my other notes app for that sort of things". This made the decision about cutting the feature easier.

B. Mobile Media Design

The difference between desktop applications and mobile applications can be large. First you have the most obvious difference like screen size and performance, but the use of a mobile in general also differ a lot from a computer. The small screen size forced us to prioritize what really mattered to the users. While location and time act as constraints on the mobile design process and forced us to think differently about how people would use the product[6]. When the group first started coming up with ideas we focused a lot on the medium we were designing for and how we could take advantage of it in an application. The group looked at situations where people have their mobile phone available, found a problem, and tried to create a mobile

application that would be able to solve it. We did not limit our user group to any specific age group, but all our user testing were performed on students from the department of informatics for practical reasons.

The application in this project were designed for touch screens, due to this the testing on low-fidelity prototypes worked very well. The size of the wireframes were about the same as an iPhone, when they were tested the users could point at and touch the drawings as if it was on a real phone. We could then get an impression of how large buttons should be and where to place different elements in the interface. This is more natural to do on a sketch of a phone application then on a sketch for a desktop application where one normally would use a mouse arrow for navigation.

The group members were both Android users and iPhone users, and were used to different conventions from the different platforms. Early in the process it was decided that the group were creating a web application¹, we did not have any important features that required it to be a native application² and we wanted to reach out to as many users as possible. This affected the design, both Android and Apples iOS have different design conventions they follow. First we placed the menu on the top which is the standard on an Android application, but after having tested the wireframes on two iPhone users it was decided to move it to the bottom, as they were used to from their iPhone. If the same wireframes had been tested on Android users we might have gotten a different result.

From this example one can see that the results from the testing would have been more usable if we as designers stuck to the conventions for the platform we were designing for, and made sure that the test persons were users of the same platform. When the group performed usability testing of the prototype later in the process all the tests were conducted on an iPhone, also here we should have considered the conventions and made sure that the test people used iPhone in order to get more valid results. The usability testing somewhat proved that we managed to stay focused. The test people did not get any time to get to know the application, but because of its few features they learned quickly and found it easy to use. This in spite that the application was tested on an unknown platform for some users.

1 http://webtrends.about.com/od/webapplications/a/web_application.htm

2 http://www.pcmag.com/encyclopedia_term/0,2542,t=native+application&i=47651,00.asp

During the real-life testing the students were asked to use the application on their own phones. This could have been a problem to go through with if very few students had a touch phone, but it turned out not to be an issue. We had only tried the application on the native browsers on Android and iPhone, and some of the users that were testing it experienced problems as they were using a different browser that did not support WebKit³. In order to make sure the users were served content adapted to their phones conventions the application could have implemented so-called browser sniffing that could determined the web browser that was used and adapt from that.

Some users tried to tap the question card or shake the phone to get a new question. It is getting more and more common to interact with touch interfaces, but as we can see we can still not take for granted that people know how to use gestures without guidance. Småprat relies on people using sliding gestures in order to display new questions, we were continuously reminded of the importance of action cues to increase the affordance.

IV. DISCUSSION

C. Work process

As mentioned scrum provided a framework to follow in the work process, but allows various processes and techniques within this framework. As the group did not have much experience with developing mobile applications time estimation that was repeated in front of every sprint, turned out to be one of the biggest challenges. The techniques that were used was planning poker⁴. Here the team used a stack of cards with different points to estimate the time it would take to build a feature. However when this planning occurred the group did not take into account the time it would take to learn JavaScript, Sencha⁵ and so on. As a result the estimation turned out to be very inaccurate.



Planning poker, group 1 MEVIT4640

3 <http://www.webkit.org/>

4 http://en.wikipedia.org/wiki/Planning_poker

5 <http://www.sencha.com>

The burndown chart that visualised the progress of the work gave us the opportunity to discover that we were on a wrong track early in the project. In order to be able to finish the project the group needed to take some sort of action. The product backlog was organised in descending order after level of importance. To get back on track the group had to decrease the amount of features by making a cut in the backlog and remove some of the less important features.

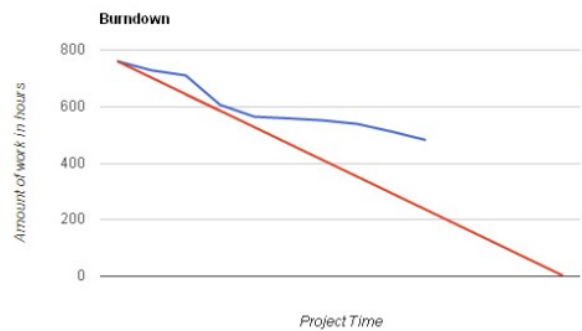


Fig. 3: Burndown chart before feature were removed, group 1, MEVIT4640

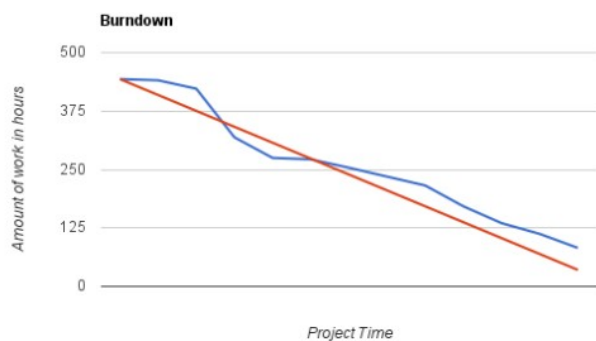


Fig. 4: Burndown chart after features were removed, at the end of the project, group 1, MEVIT4640

At the end the project was not completely finished, as there were some bugs that needed to be fixed and there were still estimated more hours then the group had time for in their schedule. However the project ended up much closer to a shippable project then it would have done if features were not removed.

What we experienced in this process was that short iterations and frequent intervals of feedback through the sprint review and burndown chart, provided a greater visibility of the state of the project[3]. The process gave us control over the unpredictability, and we learned that in this adaptive process it was necessary for us to deal with changes in required features[3]. Without the

burndown chart we might not have discovered this in time to straighten it up and not ended up with a shippable product in time of the delivery.

D. Design in Scrum

One challenge that were discovered early in the project was how to include the design in the scrum framework. Designers are usually more involved in the planning, like creating the product backlog together with the product owner, then the actually construction of the product. Martin Fowler discusses[3] that designing and construction are two very different activities, and should be separated from each other in the work process. He describes design as more difficult to predict and a process that requires expensive and creative people compared to construction which is easier to predict[3]. To be able to deal with the construction in a predictable way Fowler states that you first have to create the design before you make a plan for the construction[3]. The plan will contain the tasks that needs to be done and the relations between the tasks. A predictable schedule will make it easier to use people with lower skills for construction.

Our approach to this project did not completely follow Fowlers suggested approach for software engineering. The group started out making the product backlog describing all the requirements for the application. However we did also include things we needed to do that was not directly a part of the application itself. Among other things wireframe interface design, user testing and evaluation of sprint. After creating the product backlog we went straight to the sprint planning, if we should have followed Fowler we should have worked out the design before we started planning the construction. Preparing the scrum-board and burndown chart and creating the design became a part of the first sprints and was done simultaneously as other parts of the team started working on an UML diagram and the back-end of the application.

There are both advantages and disadvantages about including the design in the scrum framework. As Fowler [3] discusses it will be harder to estimate the time on a project when the design is not finished. To plan the construction will be difficult if it is not decided which features it should contain etc. However Fowler[3] also mentions that it is difficult to deliver a design that is capable of turning the code into a predictable construction activity, so this might be an impossible target to reach. Some errors in design are often only uncovered during coding and testing. Normally testing

and debugging is a part of the scrum iterations, but the testing mostly focuses on the functionality and bugs and not usability like we included in this process. Testing on low-fidelity prototypes could have been done before the sprints started, but without a working application, we would not have been able to do the real life testing, which needed to be done in order to answer our research question.

If we look at our Photoshop sketches and the final prototype we can see they do not look similar, some of the changes were implemented as a result of the user testing others due to constraints in the JavaScript framework, our limited knowledge about it or errors that occurred in the process.



Photoshop sketches, group 1, MEVIT4640



Prototype, group 1, MEVIT4640

The work became an iterative process with several iterations of design, construction and testing. We tried to stick to the design and focus on usability as changes

came up and our experience was that it was good to have a focus on design also in the implementation process.

E. Reflection-in-Action

To make a design is complex, with lots of variables affecting the result. Donald Schön[4] describes the design as a reflective process, a conversation between the designer and the material of a situation. Meaning that designers learn from iterations of moves, which lead them to reappraise, reinvent, and redraw. With our testing we have challenged our assumptions, and had reconsider design decisions and features. We can say that the test persons were communicating the consequences of our decisions back to the team.

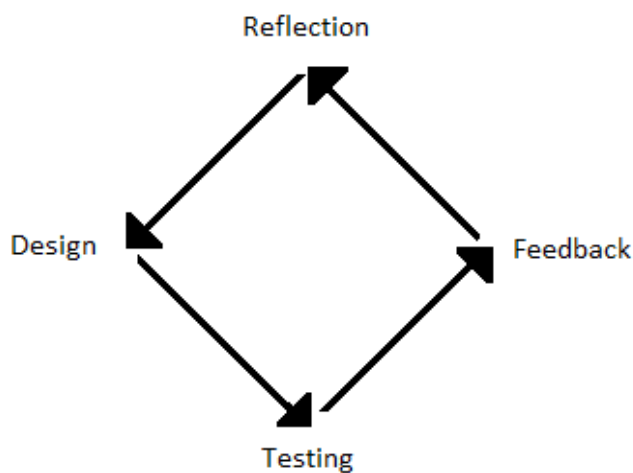


Fig. 5: Reflection-in-Action

Fig. 5 describe how I have interpreted reflection-in-action in connection to this project. Throughout the whole process we have been making moves based on the reflections of the feedback from testing and interviews, discussions within the group and feedback from the supervision session.

The supervision sessions can be looked on as reflection-on-action, as there we got feedback on what we had done and when we were suppose to have a possible shippable product, but it can also in some sense be looked on as reflection-in-action because the feedback was conducted while we were still working on the project, and it was not finished.

As we can see from this, Schöns' concept[4] of reflection-in-action it is in a way describing the Scrum framework with its iterations like we have used in in this project.

V. CONCLUSIONS

The research question was stated early in the process and played an important roll throughout the whole project, and testing was important in order to answer it, something we were aware of all the way from the start. We did make assumptions and scenarios about the use of the application, but to get a valid answer to the question we needed to talk to several users and test the application in a real-life setting.

The whole project we worked to get the prototype in such a state that it could be tested in a real-life setting without the testing being affected by bugs etc. As we can see on the last burndown chart we did not reach that goal completely, but even with some bugs we managed in the last iteration to conduct the real-life testing and answer the research question.

The research question can be divided into two parts. First, can Småprat be used to enhance conversation? Some users tested the application in secret but most people made their conversation partners aware of it during the testing. The feedback we got was that the application did enhance the conversation. Even when the questions was a bit weird people had to admit that it did start a conversation. If the conversation will be better depends on the user and the context it is used in. But for people that are having troubles finding good conversation topics this application could be good and really help the conversation going[8].

The second part was, is socially acceptable to use? Here we made an interesting discovery. Which made the answer to this part both yes and no. People said that they thought it would be OK to use it to prepare a conversation themselves, but on the other hand they would be a bit offended if someone used the application on them. Again we can see that it depends on the context and how it is used.

The project have been conducted in four iterations and the Scrum framework and artefacts have worked as a tool for learning and reflecting on our own work and design. The paper concludes that it can be difficult to include design in the Scrum frame work, but that it is necessary to focus on the design also during the construction, because some design errors are not visible before this stage. And then it is good to have designers that can make sure the usability is maintained.

ACKNOWLEDGMENT

I wish to acknowledge Anders Sundnes Løvlie, the lecture of the course for contributing in this project by giving us feedback and guiding us on the right direction throughout the process, the group which have worked so well together as a team and last but not least the students that participated in the user testing and made it possible for us to answer our research question.

REFERENCES

- [1] *The Scrum Guide* [Internet] K. Schwaber and J. Sutherland, 2011 [Loaded 10.11.2011]. Available from: http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf.
- [2] *Controlled Chaos: Living on the edge* [Internet] Advanced Development Methods, Inc., 1996 [Loaded 10.11.2011] Available from:
<http://controlchaos.squarespace.com/storage/scrum-articles/Living%20on%20the%20Edge.pdf>.
- [3] *The New Methodology* [Internet] M. Fowler, 2005, [Loaded 09.11.2011] Available from: <http://martinfowler.com/articles/newMethodology.html>.
- [4] D. A. Schön, "The Reflective Practitioner: How Professionals Think in Action", United States of America, *Basic Books*, 1983, pp.1-104.
- [5] *Scrum & Agile Introduction*, Accenture, [lecture, UiO] K. Johnsen, Sept. 14, 2011.
- [6] L. Wroblewski "Mobile First," New York, *A Book Apart*, 2011.
- [7] *Americans and their cell phones* [Internet] Pew Research Center, Washington, D.C., A. Smith, 2011, [Loaded 15.11.2011] Available from: <http://pewinternet.org/~media/Files/Reports/2011/Cell%20Phones%202011.pdf>.
- [8] *Småprat* [Presentation, UiO] M. Baas, J. Bording, A. Grenasberg S.B. Stølen, Å. A. Torpe, MEVIT4640, 2011.