# From Specific Project Challenges to Improvement of Mjølner's Software Process

Jens Bæk Jørgensen, Mikkel Yde Kjær, Inge Mølgaard, Søren Snehøj Nielsen, Niels Mark Rubin

Mjølner Informatics A/S
Finlandsgade 10
DK-8200 Aarhus N

{jbj, myk, imo, ssn, nmr}@mjolner.dk

**Abstract**
In this paper, we present a real-life software development project; analysis, design and implementation of a new home control in a joint effort between a large industrial company and the software company Mjølner Informatics A/S. We discuss the main challenges the project have faced and are facing, we reflect upon how to address the challenges and we consider how the specific project can provide input to Mjølner's general SPI program. The main challenges include getting in line with many diverse stakeholders, different approaches to the design process, dealing with requirements, and accommodating agile software development within the constraints imposed by traditional (non-software) industrial practice.

**Keywords**

Software process improvement, case study, user experience, stakeholder management, requirements, software development, agile process

# 1 Introduction

This paper is about a project that develops a new "home control" – a remote control device that has a touch user interface and is able to control several devices in a home. The project is carried out in a joint effort between a large industrial company ("the customer" in the rest of the paper) and Mjølner Informatics A/S (for commercial confidentiality reasons, the identity of the industrial company cannot be published).

Mjølner Informatics A/S (Mjølner) is a Danish software company, founded in 1988. Mjølner develops custom-made software for Danish and international customers, both in the private and the public sector. Mjølner has expertise in development of a broad range of system types. Mjølner has a User Experience Centre (UX Centre) dedicated to ensure that all solutions have a high degree of user-friendliness. One role of the User Experience Centre is to ensure that the users' needs are supported by the systems that are developed. Mjølner's trademark is to combine a high professional competence with effective and flexible project management and cooperation with the customers. The emphasis is on knowledge sharing, innovation and partnership.

In early 2010, the customer contracted Mjølner to develop the user interface of the new home control. At the time of writing this paper, the project has lasted about a year, and is planned to continue until the autumn of 2011.

This paper will present the main challenges encountered until now, and describe how these challenges have been addressed or have been decided to be addressed. The primary purpose is to put spotlight on areas where this particular project has inspired improvements of Mjølner's software development process [Kjær & Jørgensen 2010], based on the special needs in this project. In the end, future projects must benefit from the experiences gained in the particular project.

A short introduction to the project will be presented, and then four examples of main challenges where an adjustment of the process was needed will be given. Subsequently a list of possible improvements to Mjølner's software development process, derived from the findings in the paper, is presented.

# 2 The Project and its Main Challenges

The goal of the project is to design and implement a new version of a home control. The customer has developed a similar home control in-house a number of years ago, and this is in wide-spread use world-wide. Many users, however, find the current home control too difficult to use. The customer believes that it has the right functionality, but it is too difficult to find the functionality. The purpose of the project is to analyze the needs of the users, design a new graphical user interface that is in aligned with the users' needs, and implement the new graphical user interface.

The project started in February 2010 at a kick-off workshop with approximately 20 stakeholders. In March and April 2010 Mjølner's User Experience specialists conducted field studies, user tests, focus groups, and other user research activities in two of the main markets of the customer.

All the research information gathered in the user research activities was scrutinized, discussed with the customer and consolidated. Among the most important results from this process was the definition of personas [Nielsen 2004] that represent key user profiles, whose wishes should have proper priority in the design of the new user interface. Along with the personas a list of the main user scenarios for the home control was created in order to focus the following prototypes on the main usage and user needs. The analysis work was guided by Mjølner's software development process, which contains a user experience "tool box" that was applied for user research.

In August 2010, the preliminary results were presented to the top management of the customer. The top management wanted to approve fundamental decisions about the home control, which is of strategic importance. At this meeting fundamental decisions were taken, involving design and distribution of

the home control, and it was decided that the top management were to be involved in evaluating project milestones.

Afterwards Mjølner started to write a requirement specification. Though the new home control should have the same functionality as the existing control, no specifications existed and it was experienced that different stakeholders had different requirements. The requirements specification work obviously benefitted from the user research that had already been carried out. This work was expanded and supplemented. In particular, the business goals were clarified; the business goals are statements of the customer's ultimate purpose with the project, often in the form of increased revenue, higher attractiveness and bigger market share, which was also the case here. The business goals were coupled with the scenarios – many of which already existed – that describe how the business goals can be reached. Use cases decomposed the scenarios and it ended up with verifiable requirements, as shown in Figure 1.


**Figure 1 - Elicitation of requirements.**

Respecting the fact that requirements should be elicited and categorized on four levels as in this approach is an essential ingredient in Mjølner's software development process. It is a well-known approach similar to, e.g., Lauesen's categorization of requirements in goal-level, domain-level, product-level, and design-level requirements [Lauesen 2002].

In the autumn, two competing interactive and graphical prototypes were developed and tested with end-users and evaluated by the stakeholders including the top management. In December 2010 key decisions about the hardware, the purchase process, and graphical user interface were taken, which meant that the design process could now focus on more detailed design decisions. In January 2011 the software development started in close cooperation with a team at the customer.

In this paper the four main challenges that have been faced in the project, will be elaborated:

- A large and diverse set of stakeholders
- The customer was used to a functionality-based process
- Many requirements are implicit and not well-agreed upon
- A high degree of parallelization

Obviously, it is not the first time that we meet these and similar challenges, and, in general, we do have methods to deal with them. However, in the specific project we are discussing in this paper, we have seen these challenges materializing more strongly than often, and this is what have inspired us to pursue even better methods to address such challenges more effectively.

## *3 A Large and Diverse set of Stakeholders*

As the project progressed Mjølner gradually realized that the participants from the kick-off meeting were not the only stakeholders and decision makers in the project. Some of the other stakeholders include communication and brand managers, hardware specialists, various middle managers, translation and instruction manual representatives, and not the least top management.

The large and diverse set of stakeholders often had conflicting perspectives on the design and functionality requirements. E.g. at a certain point in time it was learned that the sales and marketing representatives felt that their wishes did not get sufficient attention since the home control appeared too technical and not sufficiently attractive to the market. Also the brand and communications managers, who ensure the brand consistency, were involved very late in the process after several design iterations. The involvement of the top management obviously is a prime example of key stakeholders that

make decisions that changes the foundation of the project. In general, the involvement of different stakeholders have changed and expanded the project a number of times. This of course meant changing a lot of the produced work products.

There are several reasons why relevant stakeholders were not always involved sufficiently early. One of them was because the product owners of the company had more focus on the entire functionality requirements than the overall appearance and brand factor. A second reason was a geographical distance and professional difference between the project manager of the customer and the product owners of the customer.

A software company must be aware not only to focus too much on the technical product at hand, how to develop it, and how it fits the users' needs. This is the case because important information might be missed from important persons at the customer. It is especially true when the customer is a huge organization with a strategically important product with many stakeholders. In this case a close cooperation with the entire stakeholder gallery is of utmost importance.

To get closer cooperation between the stakeholders and the development team at Mjølner, a number of full day workshops were arranged. At these workshops relevant stakeholders were gathered and encouraged to express their viewpoints and ultimately make joint decisions. These frequent, physical meetings improved the stakeholder management.

In general terms, the observation is that in a project of a huge organizational complexity it is important on a continuous basis to take all important stakeholders into account and ensure cooperation between these stakeholders and the project team. Mjølner's software development process – including project management process – does offer support for these activities, but will benefit from even better tools and methods to support the identification and involvement of many diverse stakeholders.

# *4 The Customer was Used to a Functionality-based Process*

When Mjølner's UX-team began the design process, the purpose was to meet all the functional requirements of the home control and make them fit into a small touch interface in a user friendly design. The users were a diverse group with many different products, needs and readiness towards technology. The analysis shoved that the vast majority of the users only utilized a small part of the current home control since the user interface was too complicated and they did not know the full potential of the home control.

After several workshops and different design proposals it became evident that the product owners from the customer and the UX-team at Mjølner had a very different approach to the optimal design process. The two different views can be classified as:

1. The functional-based approach: Focus the design concept on the total set of detailed requirements.
2. The UX-driven approach: Focus the design concept and the interfaces to the main usage scenarios.

The first approach suggests that the user interface design is based on the entire set of functional requirements, meaning that the interface is created with the full suite of features in mind and prioritizing the advanced users as much as the main users.

In contrast, the UX-driven approach suggests that a simple user interface can only be the result of focusing on the mainstream usage [Colborne 2011]. This means that each element in an interface must be evaluated against the main use scenarios and possibly discarded when it is not a need of the main user. Creating a user friendly interface is about considering each dialogue so it does not contain elements that are rarely used [Nielsen 1994]. This approach is based on terms such as user friendliness and aesthetic experiences and it recognizes that the user's emotions – both immediate and obtained through use - are important factors of the user's dialogue with a product [Hallnäs & Redström 2002].

This meant that every time that the UX-team tried to prioritize the scenarios and user interfaces to the main users the product owners were concerned that the small and technical details were forgotten. The customer frequently brought up these issues during the discussions of the main usage scenarios. This was a challenge since the main purpose was to make the interface user friendly and these discussions were a symptom of a difference between how the product owners of the customer wanted the home control and the findings from the user research.

A combination of the two approaches was needed to get the process going and make sure everybody was committed. To ensure a focus on including all functionality requirements, a parallel process of writing down all the requirements was started – see next section for more details about this. This assured the product owners of the customer that no requirements were forgotten. At the same time the workshops were focused on a design for the main personas and usage scenarios, and the requirement specification helped drive this process.

As another mean to combine these two different approaches, Mjølner used the personas actively in all workshops and discussions throughout the entire development process. The personas were also included in all the use cases in the requirement specification as well as in a priority-matrix of all the functionality mapped with each persona. This priority-matrix aided the discussion of what functionality was relevant to which users and consequently, what functionality should be easily available and what should be tucked away in a lower level of the interface. This persona priority-matrix was furthermore used in the iterative development plan in order to develop the most essential functions first.

In general terms, the observation is that when engaging in a UX-driven design process with very functionality-minded product owners, care should be taken to combine the two approaches in order to make workshops more effective and focused. In addition, the documentation of all use cases and the persona priority-matrix made the hierarchy between personas and their needs more clear to the customer. This meant that the design and development process could be focused in accordance to the mainstream user and usage, and the customer could still evaluate that all important functionality would be available in the end.

## *5 Many Requirements Were Implicit and not Well-agreed Upon*

During the early stages of the project in the spring of 2010, the main activities were user research activities, including identification of personas and main usage scenarios. It was assumed that the existing functionality should be kept and the task was to define a new interaction structure and presentation layout.

As the work progressed, it became clear that the basic product requirements were neither clear nor stable. Many requirements were implicit and not well-agreed upon. To address this problem, the writing of a requirement specification was started. With reference to Figure 1, the work until now had focused on business goals and, in particular, scenarios. The requirement specification should consist of system requirements and include use cases and verifiable requirements.

The goal was that all expectations to the product should be stated in a way that gave the UX-team the freedom to choose different user interface designs, without a new negotiation of the requirements.

In general, the requirement specification describes the functionality by scenarios and use cases [Cockburn 2000]. Each scenario is an imagination of how a user would solve everyday tasks with the home control. In this approach, a scenario consists of a number of use cases (we are aware that in the literature, it is sometimes vice versa). Every use case describes an interaction with the product by an "actor". Each actor has different skills and represents various personas. An essential characteristic of a use case is to avoid any use or reference to user interface controls like buttons, windows or text fields. By doing so, the user interface design may change after the requirements specification has been stabilized and agreed upon. To ensure that a number of fundamental properties are stated, every use case has a number of one-liner verifiable requirement statements. A requirement statement can be verified with a "yes" or "no". These one-liner statements can later be used as testable statements in test specification.

During the specification phase, there were many meetings and coordination activities between the requirement stakeholders on the customer side and Mjølner's project team. The pace of the work on user interface design was reduced, because the requirement stakeholders were uncertain that the user interface design covered all thinkable scenarios. When the first draft of the requirement specification was ready, it was difficult for the stakeholders to adapt this abstract and general kind of view of the home control behavior. After an introduction period, the requirement stakeholders became accustomed to reading it and the use case approach made a lot of things clearer for both parties.

The cooperation about the requirement specification was a positive and fruitful process and the further writing and maintenance of the requirement specification has become a joint project between the requirement stakeholders on the customer side and Mjølner's project team. The user interface design specification, which could now easily include examples of design and behavior, was finished afterwards, based on clear fundamental requirements. The two documents described jointly the expectations to the home control.

The observation, more generally, is that when engaging in a UX-driven design process, a more technical process of identifying and writing down the entire set of requirements should be done in parallel. The purpose is to get a solid foundation for agreement about the entire functionality and to ensure the customer that no requirements are forgotten even though the main user scenarios are in focus.

## *6 A High Degree of Parallelization*

Mjølner's software development process is an iterative, agile approach [Larman 2004]. Overlaps between and iteration of analysis, design and implementation are a part of the process. For Mjølner's projects it is common that the user interface is not completely designed before the implementation phase is initiated. Of course, if the interface could be designed fully before development it would be a more simple matter to convert the complete set of fully designed graphical user interfaces into an actual implementation. In practice, however, the project in question has seen several reasons why such a sequential approach will prove impossible:

- Project deadlines dictate a high degree of parallelization. There are even periodical milestone deliveries which include some amount of work from both UX-team and the software developers.
- The UX-team requires input from the developers in order to design the user interface. This could be performance data such as achievable frame rates or which animation effects can be implemented.
- The developers require input from the UX-team in order to create a solid architecture. Some knowledge about how the user interface would look like is necessary.
- The UX-team needs to elaborate on design details and continually evaluate the implementation, which means they need to continue to work on the project at least in some extent during the implementation phase.

So it was obvious that a parallel effort by the UX-team and developer teams was required. The software development started before the user interface was completely designed and finally approved. This meant that the software developers were faced with several challenges: Since the UX-team had focused on the key scenarios, a considerable part of the user interface was largely unknown. Still, the software developers needed to come up with a coherent software architecture that adequately supported the secondary scenarios and not-yet-designed user interfaces.

Arguably the most important aspect of solving inter-team dependencies is effective communication. To that end, a project office accommodating both the UX specialists and the project team was set up early in the project. Daily meetings with all project members attending is used, and everyone gets CC'ed on important information. This is the standard way to set up a project in Mjølner.

The software architecture was defined by an approach inspired by Presenter First [Alles et. al, 2006] which enabled the developers to concentrate on core functionality and defer user interface implementation until later. This was true in theory but in order for the developers to translate the use cases to code they needed to know what the user interface would look like. This entwined UX-team and developers further.

Being agile does not mean that all kinds of changes are always embraced. It is certainly preferable that some key decisions are made and remain unchanged from early in the project. Specifically in this project constraints have been seen, which are difficult to deal with in an agile fashion, and which are not addressed by the measures taken. A main problem has been that the target hardware platform is not yet available, so it is difficult for the developers to produce the input needed by the UX-team regarding performance and possible animation effects. Estimation of the software development efforts and estimation of platform memory usage – the latter being input for the hardware design team – are other examples.

It was a big challenge to solve of performance estimation needed by the UX-team. An overly optimistic estimation may yield disappointed customers when the product is finalized, whereas a too conservative approach could result in a product that was subpar. A performance study on a similar hardware platform was conducted, and this gave a rough performance indication. Since that particular platform was not optimized for this specific purpose, it was decided to adopt the performance numbers as worst-case. The developers were then able to do simple calculations on the effects requested by the UX-team and provide their best guess as to whether implementation was feasible. Through this input, constructive dialogue between UX-team and developers usually provided a compromise both could support.

The next step was to implement a PC-based simulator of the target system. Because the hardware was unavailable, the need for a simulator was rather obvious unless development was to halt completely, but there are additional derived benefits from having a simulator. It enables early evaluations of the user interface. The simulator is also used in milestone deliveries to the customer to document progress throughout the project. Since it can easily be emailed, it can even be used to provide context for text translators or education of support personnel.

In more general terms, the conclusion is that was necessary to move forward with the implementation, even though the target hardware was not available; this was an external constraint that could not be removed, but only accepted as a given fact. Introducing the simulator was the solution to deal with this constraint and it has proved very useful. Furthermore it was found that because UX needs input from developers and vice versa a very close collaboration between UX-team and developers is imperative for a project like this.

# *7 From Project Findings to Software Process Improvements*

Four main challenges that the project have faced and are facing have been presented. To sum up, these challenges – and the key observation made about them - are:

- A large and diverse set of stakeholders: It is important for projects of this type to actively involve a broad range of stakeholders and put them together in workshop like environments to get good decisions fast.
- The customer was used to a functionality-based process: Theory might note a UX-driven design process as the best, but in cases where the customer is focused on functionality a mix of a UX-driven and a functionality-based approach makes collaboration easier.
- Many requirements are implicit and not well-agreed upon: Even though the customer has a functionality-based approach and the product developed is a new version of an old product, it is not certain that a clear picture of the requirements exists.
- A high degree of parallelization: Due to external constraints this project had to do things in parallel even though a more sequential approach would be more efficient. Building a simulator

and using close cooperation and communication – also with the development team at the customer – have been essential.

These observations may obviously be applied to other similar projects as well. Therefore, each of the four observations should have an impact on Mjølner's software development process.
At the time of writing the observations reported in this paper have resulted in the following proposals for improving Mjølner's software development process:

1. The stakeholder management process should include best practises on how to get the customer involved actively and involve all relevant different stakeholders as early as possible.
2. Our UX design process should to a higher degree embrace the fact that different stakeholders at the customers have different opinions.
3. Our UX design and requirements processes should be updated with the knowledge gained in this project about combining UX-driven and a functionality-based approach.
4. Decoupling of functional requirements and the user interface design should be adopted as a best practise in our requirement process
5. Our project management process should to a higher degree emphasis the need to map out the need of parallel activities and the relations between these activities.

# *8 Conclusion*

In this paper, we have discussed a particular project – development of a home control in a joint effort between a large industrial customer and Mjølner - and its challenges. Moreover, we have described preliminary considerations about how these findings can be generalized to have an impact on Mjølner's software development process. Making the actual change to Mjølner's software development process is work that will be in progress in the near future.

We believe that reporting our experiences are of interest to other software companies as well. We think that the challenges discussed are common challenges in many software projects – especially for projects that have a high degree of user interface design. And the challenges are even more relevant when a software company is delivering to an industrial company, whose main business is not software, but something else.

We hope that the descriptions in this paper contribute to fruitful discussions in general about how to address problems of this type in software development projects. Moreover, we hope that discussions within the EuroSPI community may guide Mjølner to find ideas to improve the software development process so that it more generally deals with challenges like the ones we have discussed in this paper.

# 9 Literature

[Alles et al 2006]: M. Alles, D. Crosby, C. Erickson, B. Harleton, M. Marsiglia, G. Pattison, and C. Stienstra, *Presenter First: Organizing Complex GUI Applications for Test-Driven Development,* 2006, http://www.atomicobject.com/files/PresenterFirstAgile2006.pdf.

[Cockburn 2000]: A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.

[Colborne 2011]: G. Colborne, *Simple and Usable. Web, mobile, and interaction design.* New Riders, 2011.

[Hallnäs & Redström 2002]: L. Hallnäs & J. Redström, *Abstract Information Appliances; Methodological Exercises in Conceptual Design of Computational Things*. In Proceedings of DIS2002*,* London, ACM Press.

[Kjær & Jørgensen 2010]: M.Y. Kjær and J.B. Jørgensen, *Mjølner's Software Process Improvement – a Discussion and Strengthening Using the SPI Manifesto*, Proceedings of EuroSPI2010, Grenoble, France, Springer, 2010.

[Larman 2004]: C. Larman, Agile & Iterative Development, Addison-Wesley, 2004.

[Lauesen 2002]: S. Lauesen, *Software Requirements – Styles and Techniques*, Addison-Wesley, 2002.

[Nielsen 1994]: J. Nielsen, *Ten Usability Heuristics,* 1994, http://www.useit.com/papers/heuristic/heuristic_list.html

[Nielsen 2004]: L. Nielsen, *Engaging Personas and Narrative Scenarios,* PhD dissertation, Copenhagen Business School*,* 2004.

# *10 Author CVs*

**Jens Bæk Jørgensen**

Jens Bæk Jørgensen works as senior project manager at Mjølner. Jens has a Master's Degree from 1991 and a PhD degree from 1997, both in computer science from the University of Aarhus. Previously, Jens has worked in the software industry for Meta Software Corporation in Cambridge, Massachusetts, and for Systematic Software Engineering in Aarhus, in addition to an employment as assistant and associate professor at the University of Aarhus. Jens has experience with a broad range of roles in software development projects – developer, analyst, architect, consultant, and project manager. Jens has been doing research on software engineering for many years, in particular on requirements engineering; he has authored approximately 40 peer-reviewed scientific papers (see http://www.daimi.au.dk/~jensbaek/), most of which are published in premier journals (e.g., IEEE Software) and well-respected conferences (e.g., International Conference on Requirements Engineering).

**Mikkel Yde Kjær**

Mikkel Yde Kjær works as SPI coordinator and line manager at Mjølner. Mikkel has a Master's Degree in computer science from the University of Aarhus from 1999. Previously, Mikkel has worked for Systematic Software Engineering in Aarhus, where he has been actively involved in getting the company certified as CMMI level 3, 4 and 5. Through the years he has been involved both as a project manager and by working 1½ years as Change Agent in the SPI department. Mikkel has been using Scrum and other agile methods since 2004 and has a special interest in Lean Software Development. Mikkel has a broad software experience – developer, project manager, consultant, change agent, teacher, and facilitator.

**Inge Mølgaard**

Inge works as UX (User experience) consultant at Mjølner. Inge has a Master's Degree in Information Science from the University of Aarhus from 2006. Previously she has worked as a UX-consultant for the digital agency Creuna and for the Alexandra Institute that ensures a closer cooperation between the world of research and private corporations and public institutions. She has been in charge of large-scale web- and intranet solutions as well as it-solutions integrated in libraries and museum throughout Denmark. As a UX consultant she is used to collaborate closely with customers in order to gain insight in the organization and business goals for the it-system and to manage processes that collects information and requirements from diverse groups of stakeholders.

**Søren Snehøj Nielsen**

Søren Snehøj Nielsen has a Master's Degree in Engineering from the University of Aarhus and works as a senior software developer at Mjølner. Søren has 7 years experience developing embedded software for a wide range of industrial systems. Over the years Søren has been active in many aspects of the Software Process Improvement activities at Mjølner, especially relating to architecture and project management.

**Niels Mark Rubin**

Niels Mark Rubin works as senior project manager at Mjølner. Niels has a Bachelor Degree in Electronic Engineering from Faculty of Engineering at University of Southern Denmark (former Odense Teknikum) from 1979. Working in the software area for over 30 years, Niels has experience with the many aspects and challenges of software development. He has been employed at Brüel & Kjær, Nærum for 10 years working with development of an OO operating system and applications - development of Element Management Systems at DSC Communications, Ballerup – developing mobile supervision and control solutions for industrial control systems at Resource Software Engineering, Aarhus - designing and managing complete flow control solutions for industrial turn-key systems at KJ-Industries, Aarhus and at Systematic Software Engineering, Aarhus. Niels has experience with a broad range of roles – developer, team-leader, product manager, architect, consultant, and project manager.