

Pilot implementation Driven by Effects Specifications and Formative Usability Evaluation

Anders Barlach, Morten Hertzum and Jesper Simonsen

Roskilde University

barlach@ruc.dk, mhz@ruc.dk, simonsen@ruc.dk

EXECUTIVE SUMMARY

This chapter reports on the usability-engineering work performed throughout the pilot implementation of an Electronic Healthcare Record (EHR). The case describes and analyzes the use of pilot implementations to formatively evaluate whether the usability of the EHR meets the effects specified for its use.

The project was initiated during the autumn of 2010 and concluded in the spring of 2012. The project configured and implemented an EHR at a Maternity ward at one hospital located in a European region and then transferred this system to another ward at another hospital in the same region.

The project was conducted using effects-driven IT development: a process comprised of workshops with specification of the usage effects by management and end-users followed by an agile development process progressing through mock-ups, prototypes and finally the pilot system. Effects were iteratively refined and evaluated to achieve alignment with the intended design, and quantitatively measured to document the desired effects.

The pilot implementation is analyzed, and the lessons learned are discussed in relation to usability engineering in general.

ORGANIZATION BACKGROUND

This section introduces the chapter and describes the experiences of the IT vendor CSC Scandihealth in working with pilot implementation on the basis of effects-driven IT development and the EHR client, a large hospital complex located in a European region, the Hospital for short.

A pilot implementation is defined as: “a field test of a properly engineered, yet unfinished system, in its intended environment, using real data and aiming – through real-use experience – to explore the value of the system, improve or assess its design, and reduce implementation risk.” (Hertzum, Bansler, Havn, & Simonsen, 2012). Pilot implementations are field trials and in that sense constitute a continuation of prototype evaluations into the field. In this chapter, we describe a case where the preparations were carried out before a pilot implementation included using workshops with mock-ups as well as several versions of prototypes. The pilot implementation was supported by so-called ‘effects-driven IT development’ (Hertzum and Simonsen, 2011) by which the desired effects of using the system were specified, used as specifications for the mock-ups and prototypes, and finally measured as part of a formative usability evaluation based on the system used during actual work as part of the pilot implementation. In the following, we outline the strategies of both the vendor and client and the circumstances making the pilot implementation and effects-driven IT development relevant. We set the stage and describe the effects specified to produce the input to the succeeding usability evaluation. Then, we describe the case and the pilot implementation including planning and design, technical configuration, organizational adaption, use of the system, and the learning that took place. We conclude the chapter by discussing challenges, solutions, and recommendations.

Organizational Facts and Strategy: CSC Scandihealth

CSC Scandihealth is a company within the Computer Sciences Corporation and is part of its global healthcare Vertical, which specializes in delivering IT to public and private healthcare providers. CSC Scandihealth (in the following referred to as CSC, for short) employs 375 healthcare IT specialists in Denmark with an annual turnover of USD78 million.

The CSC's mission is to “deliver a *solution* to the client”, rather than “delivering a *product* to the client.” This marks a decisive shift in attitude towards the vendor-client relationship in CSC. To achieve this goal, the company and its employees must engage in various processes that must be aligned with the clinical work the solutions are to support. The increase in intimacy with the client’s core business and the need of establishing technology consistent with individual healthcare providers is stated by CSC’s CEO Freddy Lykke:

The vision we have is that the healthcare community is increasingly integrated into the various sections of society, coming closer and closer together, and our mission is that we [CSC] can help to make the linkage of the Healthcare community (Barlach & Simonsen, 2011).

Freddy Lykke elaborates how this vision can be implemented through a bottom-up approach:

The bottom-up approach is basically another way of saying that it is the clinicians themselves that are to define how their work processes are to be supported, rather than we [CSC] come with a system where we have defined how we think the workflow should be at the various hospitals and their wards. We present a system that allows the clinicians to dynamically describe how the system should work in their specific situations (Barlach & Simonsen, 2011).

Organizational Facts and Strategy: The Hospital

Denmark is divided into five regions, each responsible for providing publically funded hospital services to approximately 1 million citizens. Rehabilitation and prevention is delegated to the municipalities within the regions and to General Practitioners, who are self-employed and funded directly by the national government. The Hospital is situated in one of the five regions and comprises:

- 1,753 beds plus an additional 50 beds in the patient hotel.
- Approx. 8,700 employees
- An annual budget of nearly USD1279 million.

The overall goals of the Hospital are to deliver a consistent service of high quality while remaining an effective organization within the financial limits of the region (Region North Jutland Consolidated, 2011). The Hospital management has defined five strategic focus areas for IT in the region:

1. Stable and reliable operation of the solutions supporting the end-users.
2. Effective utilization of IT in the region.
3. Coherence in services and decisions across the organization.
4. Innovation as a factor in creating growth and an attractive workplace.
5. Ensuring ownership and clear responsibility in management and planning.

The first two focus areas calls for the use of effect as the actual use of the systems becomes a success parameter for any IT implementation and therefore must be subject to an evaluation. The complexity of healthcare work and the uncertainties in requirements defined in advance of a project start, means that the Hospital needs a process enabling learning during IT projects. Experimentation becomes a mean to acquire knowledge on how to achieve desired effects. The tool is formative evaluation of effects specifications that set the success criteria of the design in use (pilot implementation).

The overall software development context

We look at the different life-cycle models present in relation to the case to illustrate how pilot implementation interacts with both models from an industry perspective. The development of the technical framework in a slow forward moving Waterfall life-cycle dictates how the prototyping process can evolve and adds to the complexity of mounting a pilot implementation. On the other hand the framework development has the potential to gain valuable knowledge from the prototyping and later pilot implementation.

By comparing Prototyping (Nielsen, 1993) (Jansson, Handest, Nielsen, Saebye, & Parnas, 2002) and “traditional” Water-fall (Royce, 1987) project life-cycle models in this client setting we aim to illustrate how the project benefit from the pilot implementation and help identify challenges.

The Waterfall model is characterized by an assumption that the lifecycle of software development is a sequential process beginning with specification and ending with test and delivery. Requirements can be identified in advance before programming is carried out and testing is intended to confirm the fit between the specified requirements and the actual needs of the end-users once implemented (Royce, 1987). A pilot implementation in this setting would not make much sense; since the majority of resources available to the project have already been spent on implementing the knowledge (requirements) available, and changes would not be scheduled before the system goes into production.

Prototyping differs from this approach by using experiments (prototypes) to learn as requirements are specified and understood in a simulated context of use before putting large amounts of resources into implementing the requirements (Floyd, 1984; Nielsen, 1993; Stapleton, 1998). Pilot implementation takes this approach a step further by evaluating the use in a restricted but real work environment (Hertzum, et al., 2012).

How come the Water-fall approach is applied in the first place? The reason could be collaboration between client and supplier is regulated by commercial contracts regulating the transfer of responsibility during the system development lifecycle. This transfer requires strict boundaries between requirements before the development and delivery of a system in accordance with predefined specs, and the Waterfall approach is deemed particularly suited for this type of process by the CSC and Hospital organizations. In the client case with the CSC, it is a public institution (The Hospital) with a standard contract that governs the overall implementation project. These contracts are regulated by government agencies and contracts are only beginning to allow for specifications to be discovered during the development lifecycle, but still with the lion's share of the risk carried by the vendor.

Another reason for choosing the Waterfall approach can be found in the complexity of the systems being developed. The complexity of making **Reliable, Adaptable** systems that can be **Maintained** and **Perform** in the environment for which it is intended requires much effort and time to be successful and advocates against the use of prototypes as the primary model for the project, in short these high level requirements are called RAMP (Davis, Bersoff, & Comer, 1988).

To illustrate the differences between the two life-cycle models we adopt the metrics suggested by Davis et al. (Davis, et al., 1988). The metrics suggested are; *shortfalls* as how far are the developed system from fulfilling the actual requirements while applying the two different models (Davis, et al., 1988). Another metric could be *lateness* since the process of learning is characteristic for most project models they all generate new requirements and has different degrees of latency from discovery to implementation (Davis, et al., 1988).

Adaptability to new requirements or longevity of the system as seen in how long before it needs to be replaced could be candidates for these metrics too. Finally, *inappropriateness* as the ability or lack of same, to meet new requirements without delay during the remaining lifecycle, e.g. ideally a new requirement can be met with an existing functionality (Davis, et al., 1988).

In the CSC-Hospital case, both life-cycle models are present for several reasons. The primary reason for the Waterfall model to be implemented is a consequence of the contract dictating changes of lifecycle phases where money and responsibility are exchanged between the client (The Hospital) and the vendor (CSC). This means specifications are made before development or programming begins and requires careful negotiations to be changed once the contract is signed. The other significant reason for the Waterfall model to be present can be seen as a consequence of the complexity of an EHR. The RAMP requirements in a mission critical system like an EHR, is demand for a stable and performing platform while being adaptable to a large organization as found in the case. These requirements are not specified by end-users directly but are found in the service level agreements of a typical client-vendor contract such as the one governing the project in the case.

The project needs both models; the handling of RAMP requirements by the Waterfall model and the agility of Prototyping while learning and adapting to end-user requirements. The solution is to implement a two layered design by building the system in a technological framework with configured elements.

The framework is developed in a strict Waterfall organized project environment, and covered by the main client-vendor contract. This means *shortfalls*, both originating from the contract specifications and changes discovered during the project, are approximately one year underway. This latency would be responsible for a high degree of lateness since the process of design, programming and test before being released means learning is slow to be incorporated. If this was the only way development was undertaken, the project would be too slow to learn. The

appendix of the contract allows for more agile configuration processes to negotiate the risks of *lateness* and *shortfall* as confirmation of requirements implemented in the system fit the actual need experienced by the end-users, would not be possible before the final system implementation was completed.

By splitting the system and project process into two, the project is making up for this latency in meeting requirements, and the ever changing organization of work carried out by the client end-users. With the introduction of prototyping and pilot implementation in the configuration process, the project is attempting to adapt and close the gap between specified and actual requirements. This configuration is organized with effects driven method and adoption of elements from different sources of process improvement, which have been integrated and adapted internally to local conditions. It involves elements of agile development methods such as the Dynamic Systems Development Method (DSDM) (Stapleton, 1998) and Scrum (Schwaber & Beedle, 2002). The purpose of designing with prototypes is to negotiate uncertainties in the specification and challenge the participant's visions, expectations, and hypotheses through concrete IT experience (Bødker, Simonsen, & Kensing, 2009; Simonsen, Hertzum, & Barlach, 2011). The purposes of effects in this prototyping and pilot implementation setting are dual; Effects are used in specifying requirements during the configuration process resulting in the prototypes. Effects are also used to evaluate how the end-users experienced the actual requirements as the system is first used in a laboratory setting before it is put to use in a pilot implementation. The EHR is built in a configurable standard system or technical framework; CSC Clinical Suite™ (CCS) as the Hospital's overall electronic healthcare record (EHR). CCS is implemented in overlapping stages as the Hospital gradually adopts the new technology.

The main EHR project has been underway for 4 years and has entered a phase where the technological framework is in place, and the need for adapting the configurable elements to the end-users' work requirements are the main focus. The pilot implementation presents the Hospital organization and CSC with an opportunity to learn about the new challenges facing them onwards in the implementations process. It further allows them to experiment with ways of achieving the goals set out by the Hospital management in the strategy; effective use of IT and a stable platform across the organization. Further the Hospital aims to establish the necessary competences to configure and organizationally implement IT systems within of the organization. In other words, Hospital staff must learn to undertake agile processes while experimenting with design to support clinical work. To meet the strategy's call for organizational learning and effective utilization of IT, the project studied in this case choose to apply a pilot implementation in the process.

Pilot implementation

It is well known to use various prototype versions of an IT system during the development process as it allows the designer to learn how end-users respond to the design (Floyd, 1984; Nielsen, 1993) and make a formative evaluation with input on how to improve the design. In order to determine the fit between the use experienced by the end-users and the design proposed, a common approach is to use mock-ups and prototypes (Floyd, 1984) during design and development to move still closer to real-use conditions. A pilot implementation is the last step before a system enters the lifecycle stage where it is finalized for operational service and, contrary to prototyping; it involves using the system in the field for real work. We are looking into the challenges of this transition from prototype to pilot-system. The elements of pilot implementation are planning and design, technical configuration, organizational adaptation, use, and learning (Hertzum, et al., 2012):

Planning and design, the project defines how it will acquire data to evaluate the design, logistic issues like where the pilot should take place, how long it will last and how participation among the involved organizations and end-users will be coordinated.

Technical configuration, this element considers the technical tasks required to perform the transition from prototype into the pilot-system. To some extent it resembles the final phases of the development process, but since the pilot is operating in an end-user context not yet covered by the main implementation project, the pilot implementation must handle the finalization to a specific instance of real-use. Considerable re-use of this work can be expected when concluding the development project, but it is often postponed until the final design has been implemented to save development costs.

Organizational adaptation is concerned with how the organization is prepared for the use of the system. Tasks like training of end-users, coordination with adjacent units which are supporting the work performed at the pilot organization or the adoption of new or different work practices due to the system being taken into use.

Use, the system must support work in a realistic end-user environment. During use, the pilot participants can

observe and experience how the design fits end-users' requirements and make measurements to support the formative evaluation.

Learning is the process of acquiring new knowledge regarding how the design was interpreted by the intended end-users, what was supportive towards the goals of the development project and which anticipated effects failed to emerge in the pilot context.(Hertzum, et al., 2012)

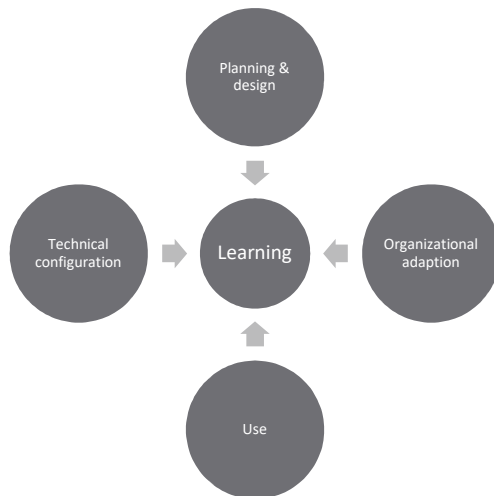


Figure 1: A model of pilot implementation showing the four elements; Planning & design, Technical configuration, Organizational adaption and Use contributing to learning (Hertzum, et al., 2012).

EDIT

The project needed a method that would allow for learning through formative evaluation and the need for pilot implementation to negotiate implementation risks while adhering to the strategy of both CSC and the Hospital. The choice was made to apply the effects-driven IT development (EDIT) method (Simonsen, et al., 2011). The development project was organized as a Rapid Application Development (Hainey, 2007) process involving methods like short iterations – typically 2 weeks – and prototyping following CSC’s effects driven IT-development. The Hospital provided the project management and the configuration of the electronic Partogram-Record while CSC was to provide process support, design suggestions and ensure documentation and evaluation of the project.

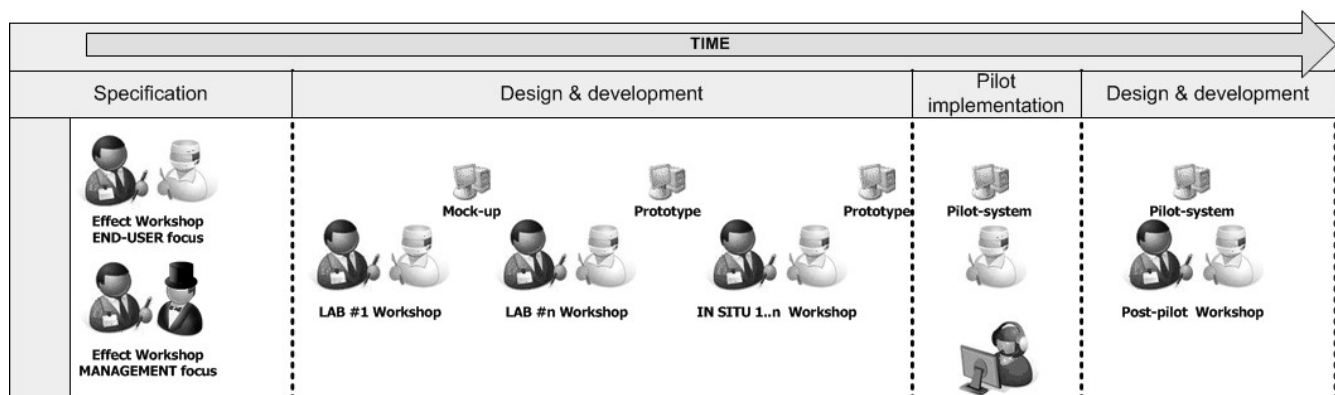


Figure 2: EDIT process, starting out with effects specification workshops, followed by iterations of workshops with mock-ups/prototypes before culminating with a pilot implementation.

The EDIT process is a series of workshops driven by effects as the specification and usability evaluation tool with participation of different usability actors or stakeholder representatives (Figure 2). The usability actors in

the project are:

- Hospital organization managers; senior management representing local management across the Hospital organization, responsible for budget and strategy implementation at the political level.
- Hospital organization project staff; responsible for the project deliveries, planning and configuring the actual system, employed by the Hospital organization
- End-users; clinical staff having first-hand experience with work performed in relation to the use of the system during the pilot.

The process is initiated with two parallel workshops involving management and end-users. They are held separately since experience shows the two groups have different priorities and needs regarding the process, and outcome. In both workshops design requirements are specified using effects rather than functional descriptions of the future system. Effects relate to the actual work the system is intended to support by taking into account variables like; who are the end-users and what is the expected outcome of the task supported. An example of an effect specified in the project could sound like this:

The Midwife on call must be able to gather the general picture of the relevant tasks performed and interventions planned regarding an individual patient in her care.

This effect can be evaluated as it is performed at the end of a watch/shift or while handing over the care responsibility of an individual patient. The midwife must experience sufficiency in the knowledge presented and not feel the need for looking elsewhere for missing information. The measurement of the effect lies in asking for the experience of the midwife in the context of use. E.g. “Do you miss any information after having used the system to gain an overview of the patient?”

Considering the **technical configuration** and the **organizational adaptations** (see figure 1) the project **learns** as it moves forward in iterations with new versions of the mock-up or prototype each time. Progress is documented by the fit of the Effects specification with the prototype as it is evaluated in the workshops, which aim to move still further away from simulating work and towards conducting actual work (**Use**).

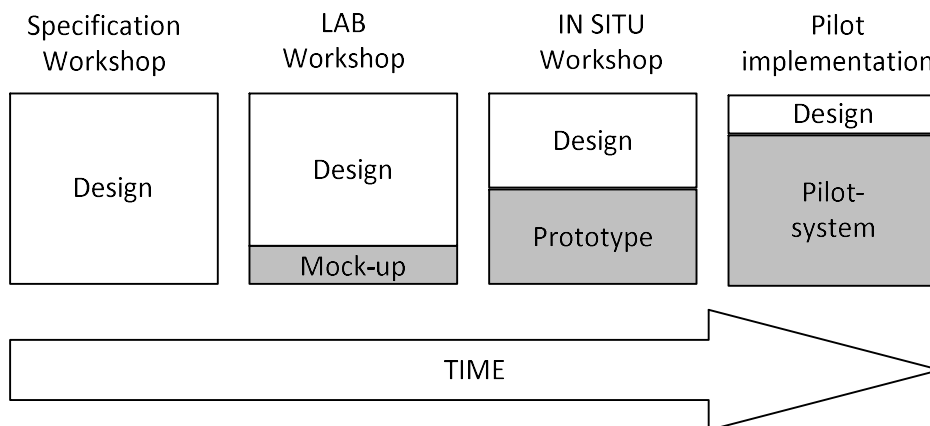


Figure 3: During the EDIT workshops, the role of the effects design diminishes in relation to the role of mock-ups & prototypes when performing the formative evaluation to determine progression towards enabling real use.

The formative evaluation allows the project to learn and progress toward more complete representation of a system with each iteration. Using mock-ups and prototypes as tools during the EDIT process can be seen as a progression towards the pilot system, which is capable of supporting real work (Figure 3).

To understand how mock-ups and prototypes differ from each other and the pilot system we adopt an ANSI/SPARC (Tsichritzis & Klug, 1978) inspired perspective on what an information system consists of on a general level. In short, the model has three layers: an external layer represented by the graphical user interface (GUI), the conceptual layer containing functions and providing the necessary means to access data in the internal layer and transfers them to the external layer. The internal layer contains the data model and is an extension of the database allowing for persistence of data (figure 4).

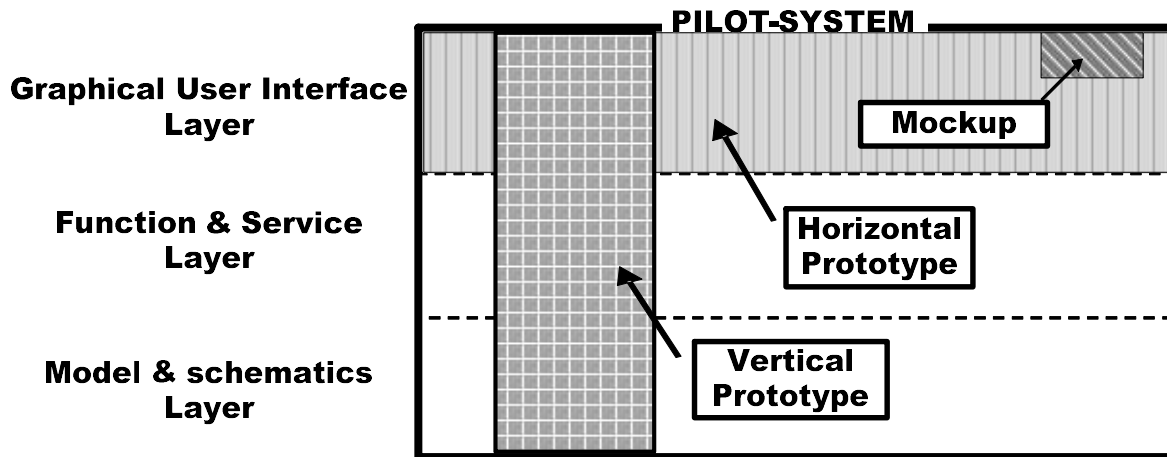


Figure 4: Differences among mock-up, horizontal prototype, vertical prototype and the pilot-system from an ANSI/SPARC three layer perspective.

If we look at the different representations of the pilot-system design during the EDIT process, they start out as mock-ups. This is a simple way to experiment with design ideas, and represent only a very small part of the GUI with no function or model representation in comparison to the envisioned pilot system.

Evolving through the EDIT workshops (figure 3) the prototypes can be horizontal or vertical (Nielsen, 1993) depending on the nature of formative evaluation the EDIT process requires.

The pilot system is the goal of the process, encompassing all the envisioned use performed by the intended usability actors and fulfilling the requirements set for the project. It is important to realize that each step made in the process requires effort as with each new prototype and finally the pilot system covering all three layers both vertically and horizontally must be realized.

SETTING THE STAGE

The prototypes and pilot system are developed in CCS, which is a highly configurable framework tool based on the Oracle Healthcare Transaction Base™ (HTB). It is a multitiered platform and can be configured by various mark-up technologies XML, XSL etc. to define the elements of the EHR; overviews, clinical notes, results, standard plans, work situations, and the structure of the patients' medical record within the framework of the CCS.

In accordance with the strategy of the Hospital and considerations within the team of Hospital IT configurators the project defined the scope for knowledge of the pilot implementation as;

- Can the CCS support work in a clinical context? If not CSC and the Hospital have a serious problem delivering a working EHR?
- Can the end-users in another location adapt to a solution implemented in one location without changing the design? Again this is a major concern with CSC and the Hospital as the cost of making a new design for each unit would challenge the entire project. Further extension of project activities would tie up CSC resources in a development project longer than anticipated jeopardizing other deliveries.
- Can the Hospital learn to develop solutions in CCS using pilot implementation? It is the intension of the Hospital to become independent of CSC when it comes to configuration and implementation of CCS. Being able to rely on internal resources provides the necessary agility to achieve an ambitious implementation plan. Alternatively the Hospital is forced to spend time and money buying these services from external partners on a consulting basis.
- Can effects (EDIT) provide the project with a tool for formative evaluation? The Hospital's strategy states that they want to be able to evaluate the use of IT and work focused towards effects from the use of the CCS.

Effects in EDIT

Effect specifications are descriptions of the effects that the usability actors or end-users would like to obtain when they start using the envisioned IT system. CSC uses a generic template for effects specifications. This template has five parts: *Effect* (the effect to be obtained in a specified situation), *agent* (the end-user of the system in this situation), *practice* (a description of the clinical activity and intervention involved in the situation), *outcome* (the result of the activity), and *evaluation* (a description of how to assess the extent to which the effect has been achieved). The template represents an anticipated effect generated by the end-user in a specific work situation and when performing a given activity using the system.

The effects to be obtained from using the system can be assessed from multiple perspectives and at multiple levels of abstraction. Therefore, the effects are specified in a five-level hierarchy, as described in Figure 5. They range from abstract to concrete, being more and more technical. The hierarchy organizes effects to describe them as means to achieve other more abstract ends. This means–ends hierarchy is inspired by cognitive work analysis (Rasmussen, Pejtersen, & Goodstein, 1994; Vicente, 1999) and the participatory design method; MUST (Bødker, et al., 2009).

The properties represented in the effects means–ends hierarchy are purposes and reasons at the top (high level of abstraction), general processes in the middle, and more specific information processes and the physical configuration of the IT system at the bottom.

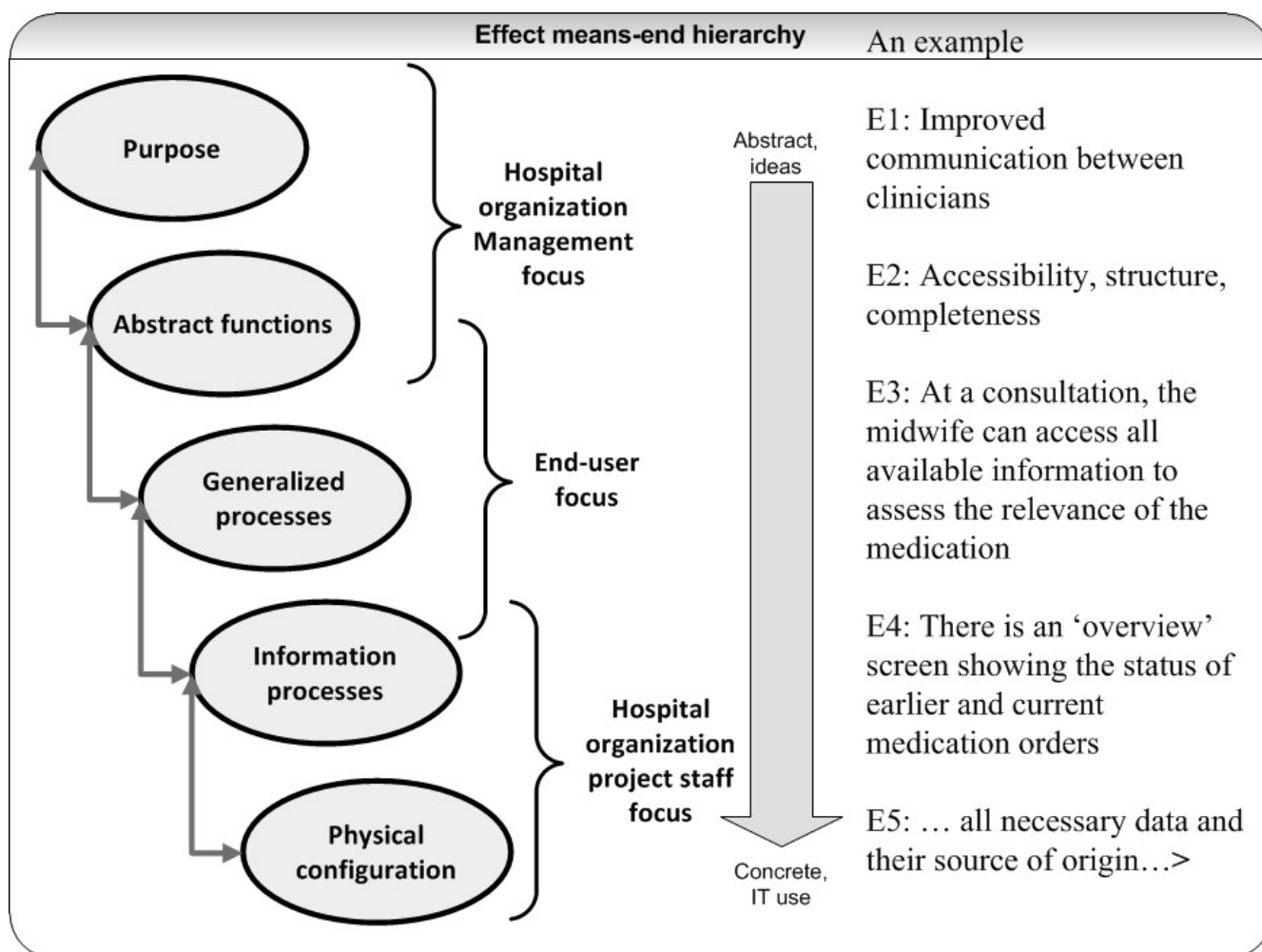


Figure 5: Effects specification in five levels, ranging from abstract to physical effects. The stakeholder groups are located at the different levels in the hierarchy.

Each level is described in the following section, inspired by work by Rasmussen and Vicente (Rasmussen, et al., 1994; Vicente, 1999):

1. *Purpose*: This is the highest level of abstraction and represents the goals and purposes in relation to the organizational environment and the goal pursued through the lower levels. It is typically identified as policies, service goals etc. regarding quality and efficiency at the enterprise level of the client organization.
2. *Abstract functions*: This level addresses the prioritization and allocation of resources to the various generalized processes and activities on the *Generalized processes* level below. The level describes the client's response, or strategy, to the environmental demands from level one and often relates to efficiency or the quality of service.
3. *Generalized processes*: This level represents business processes in terms of recurrent input–output processes and overall activities which are general and well-known in the work domain. It is not a detailed specification of an activity but might be compared to the “black box” metaphor because subprocesses or sub-activities are not specified at this level.
4. *Information processes*: This level represents information-processing tasks that define the generalized processes, including the human activities, as well as the use of equipment. Typically, these tasks precede or succeed a clinical intervention. Example: One of the tasks during the preparations for a consultation involves looking at the overview listing past consultations to determine whether there are any topics or events of relevance to the upcoming consultation. At this level, it is possible to map activities to the forms and views in the prototype.
5. *Physical configuration*: This is the lowest level of abstraction and consists of tools or objects which are the sources of information for a given tasks. At this level, detailed descriptions of user interfaces are given as screen mock-ups or interactive prototypes.

Figure 5 also shows the focus of the usability actors. Hospital management is involved in specifying effects at levels one and two, usually by formulating strategies. These overall effects are the reference point for the end-users who specify the effects they want to obtain in their use or clinical practice. The Hospital project staff then interprets the effects at level three and translates them into effect requirements at levels four and five.

The effects specified were primarily concerned with issues regarding viewing information at critical stages of the patient pathway. They typically described hand-over of responsibility from the outpatient clinic to the maternity ward or among staff during the shifting work periods of the day.

The process was initiated in accordance with the EDIT method (figure 2). The workshops progressed from the specification of effects with the involved stakeholder from the Hospital (figure 5). CCS accommodated the prototypes during workshops and after an IN-SITU workshop, testing the prototype in the ward intended for the pilot implementation, it was concluded that the design was ready for pilot implementation, figure 6 EDIT pre-pilot phase. Preparations began immediately after and are described in the case, figure 6.

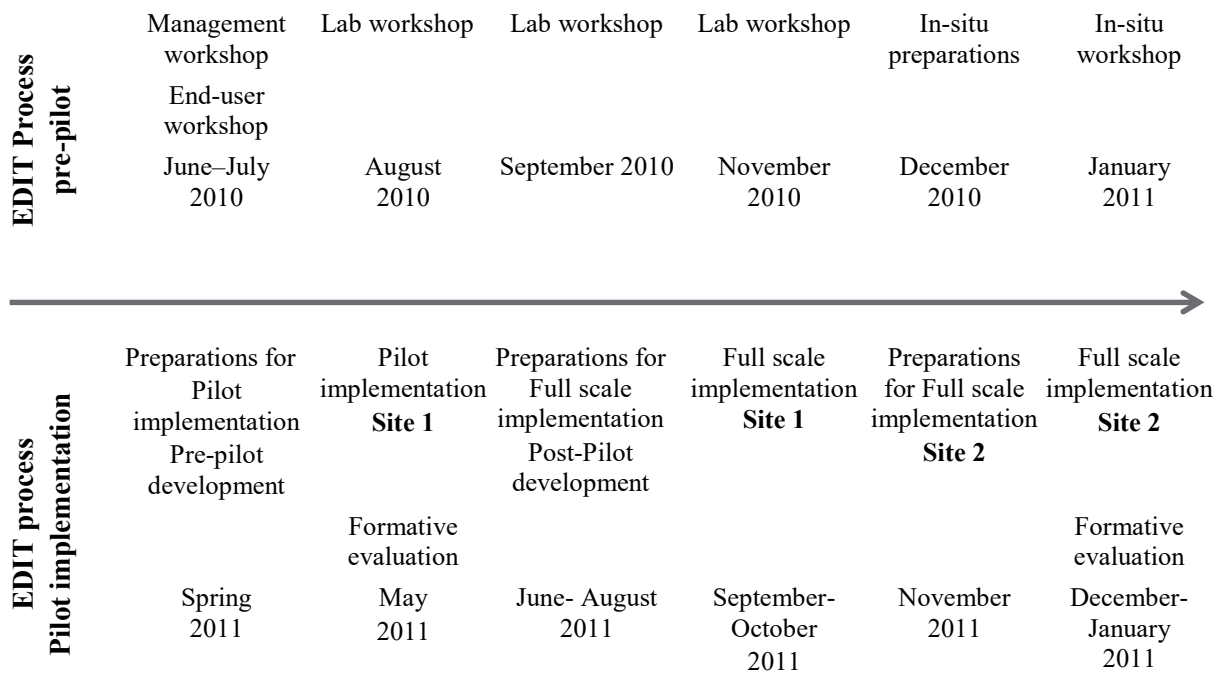


Figure 6: EDIT pre-pilot - The workshop process of design and development of the Midwife record, up until the preparations for the pilot implementation, at **Site 1** (University Hospital Maternity ward). EDIT pilot implementation – the timeline from pilot implementation and transition to full scale implementation at Site 1 and 2.

The lessons learned, regarding CSC’s use of effects in the specification and development work with the client, are listed below:

Effects are easier for clinicians to understand, formulate, and prioritize than functionality specifications (Barlach & Simonsen, 2011).

One of the major concerns during requirement specification is the representation of end-users’ actual needs in various work situations. In traditional specifications at the CSC, there is a tendency for documentation to be biased toward the configuration staff in terms of being dominated by technical vocabulary and system functionalities rather than exploring solutions to work problems in a language familiar to the end-user or clinical staff.

By specifying the system in terms of effects, the intension was to reduce the distance between what end-user expressed as requirements and what their actual needs were while working. This meant the effects could be challenged in a qualitative investigation during the workshops by asking the participants whether they could recognize the effects and whether they considered them relevant. During the pilot implementation, the effects were evaluated in a quantitative fashion using questionnaires as the usability-actors used the system at work.

Effects are stable, tempting, yet ambiguous for the IT developers (Barlach & Simonsen, 2011).

Effect specifications do not change after the early EDIT workshops (see Figure 2), and in CSC’s projects they have proven to be very stable.

Effects create a tempting innovative “free space” for the configuration technicians. The technical domain is allowed to more or less freely translate the requirements related to specified effects without the constraints of traditional technical specifications of IT requirements. This “free space” (and dealing with the lack of technical requirements) can, however, prove problematic and require more experimentation with prototypes. The ambiguity inherent in the effects as they are subject to interpretation by the technical staff is a serious issue to be taken into consideration by any project that applies the effects method and relies on the collective knowledge of the configurators responsible for implementing the pilot system (Barlach & Simonsen, 2011).

The inherent ambiguity lies in the effects' requirement notation capability, and, as reported by earlier work on goal methods; effects can also be disruptive as communication and pose a risk by not providing any detailed technical specifications (Jureta, Faulkner, & Schobbens, 2008; Stacey & Eckert, 2003). In this project and previous projects end-users tend to tolerate these imperfections and value the negotiation benefits stemming from the speed or lack of lateness of using workshops with prototypes (see Figure 1).

Effect measurement at implementation requires technically robust framework systems of a reasonable quality (Barlach & Simonsen, 2011).

The systems for which effects are to be measured and evaluated must be technically both robust as defined by the RAMP requirements (Davis, et al., 1988) and flexible so that *shortfalls* (Davis, et al., 1988) identified can be corrected within a relative short time-frame in the configured elements – otherwise it will interrupt the implementation process and put the pilot implementation at risk of being terminated prematurely. The quality of the solution must be sufficient for effects to be realizable. This entails that framework changes must not emerge as necessary when the clinicians' start using the pilot system because framework implementation is performed according to a Waterfall process and would seriously slow the overall project and contract completion. However, it is possible to ensure the prompt inclusion of immediate and emerging requirements in the technological framework, but it requires careful planning of available resources from vendor and the acceptance of re-prioritizing other deliveries from the client.

CASE DESCRIPTION

In May 2011, a pilot implementation was planned at the first site (figure 6), including both the maternity ward at a University hospital and 4 affiliated pregnancy outpatient clinics (Site 1).

After the pilot implementation, the design was evaluated, and while preparations for full-scale production were made on the technical element, the organizational adaptation was undertaken among the remaining end-users at Site 1. This included end-user training and dissemination of the documentation model among the clinical staff.

After implementing the system at Site 1, the Hospital turned its attention to the next site (Site 2). Site 2 is a smaller hospital with a maternity ward and one affiliated pregnancy outpatient clinics (figure 6). The preparations undertaken at Site 2 were not technical configurations, but primarily organizational adaptation as the documentation model devised by the end-users at Site 1 had to be adopted by the clinical end-users at Site 2. Site 2 went into full-scale production on the system in December. The evaluation performed during the pilot implementation (Site 1) and the full-scale implementation (Site 1 and Site 2) provided input to the management both at hospital level but also at the corporate level with reference to the Hospital management and product strategy and the considerations outlined in the section *Setting the Stage*.

PLANNING AND DESIGN (Figure 1)

There are several considerations involved in planning a successful pilot implementation. It requires effort of the entire organization either coming into direct contact with the pilot system or dealing with the consequences of work related to other aspects of using the system. The project team planning the pilot must consider both the resources readily available for the project to draw upon and the planning of contingencies of unforeseen events. The project manager and participants in general must consider the triple constraints allocated (time, resources and content) at every step in the pilot implementation, starting at the planning of the pilot. Some of the key considerations are discussed in the following:

Duration, how long can the organization handle the risk of the pilot implementation and allow staff to involve themselves in the system design? From the project point of view, too short a period of real-use will not allow for sufficient learning, while too long a period will tie up project resources in a redundant process and delay the completion of the project. The required length is dictated by the measurement and evaluation considerations, but there is a negotiation with the line-management involved since it is their budget and responsibility to ensure that "business" prevails without endangering the safety of the patients. As part of this negotiation the project promised hotline support on-site and a direct line to the management on the ward/outpatient clinic, in case events presented themselves warranting a fall-back manoeuvre. Experience from earlier pilot implementations with the

CSC suggests that one week is a minimum duration to achieve a level of stable use allowing for an evaluation. Two weeks are preferred, but it requires a steady hand from management and in this case the project was the first-mover, and there was no previous experience within the organization to rely on.

Measurements, if there are to be a formative evaluation it is necessary to gather data with regard to user experience while using the system. How long must the users experience the system to be able to evaluate it with a satisfactory degree of confidence? This depends, among other things, on whether the project intends to apply qualitative, quantitative or statistical methods of evaluation. A spin-off from the pilot implementation can emerge if the evaluation is adopted by the line management, considerable goodwill can be expected as good design resulting in “business as usual” or improved quality or efficiency is in the direct interest of these usability actors and management.

From the project’s usability point-of-view, speed and agility is of the essence since a credible formative evaluation means input to either design changes or adding confidence in the existing design supports the end-users’ work and brings the project closer to completion. Measurement instruments like the TLX or TAM are described in more detail in (Hart & Staveland, 1988; Venkatesh, Morris, Davis, & Davis, 2003) or how they can be utilized in Hertzum et al (Simonsen, et al., 2011). During the pilot implementation, different methods of measurement were utilized; questionnaires handed to individual usability-actors providing responses about their experience of using the EHR. Another approach was the observations made by on-site support staff from the Hospital and CSC. The measurements and observations were collected in a spreadsheet and evaluated by project staff during and after the pilot implementation.

Practical issues regarding the duty rosters. To understand the challenges involving clinical staff, it is necessary to understand the terms which they work under, and the complexity of planning depending on which parts of the organization that is involved in the pilot implementation.

There are typically 3 types of clinical work organizations interacting in conjunction over time: bed ward, outpatient clinic and day-hospital ward.

- The bed ward or just ‘ward’ is a traditional location with patients admitted for treatment and care around the clock; it is active 24 hours a day and can be found in highly specialized or general versions depending on the size of hospital and affiliation to the university. Emergency rooms, operation departments etc. are considered bed wards although they do not have patients staying overnight, but they are staffed and perform they function 24 hours a day.
- The Day-hospital ward is similar to the bed ward; however, it is not staffed at night, and consequently cannot house patients during the night. The only impact this has on the work or use of the Pilot-system is the nature of treatment and care which is limited to daytime and can be left to the patients to monitor at home.
- The outpatient clinic has a different workflow and often a supporting role in relation to the bed ward. Patient flow and tasks are well-defined and can be performed with short interactions between the staff. Typically tasks are follow-up on treatment and care initiated by the ward, but the patients do not need a continued monitoring by doctors and nurses to recover.

When planning the involvement of one or more of the above mentioned types of wards, it is easier to handle day-time operation since staffing are present in greater numbers. However, complexity and workload is higher during daytime and there are more staff and adjacent organizations to coordinate with, train and support. From a project perspective, planning must start ahead of the development process for two weighty reasons. First, keeping the momentum of the development process and still allowing for feedback from the formative evaluation means that planning cannot begin until the solution has reached the required degree of completion mentioned in the technical configuration section. Second, the planning must take into consideration that the Hospital must continue to operate at normal capacity. A pilot implementation is not undertaken without effort being invested by the Hospital. Rosters are made ahead and regulated by rules that cannot be manipulated by the project. E.g. a duty roster must be available 8 weeks before it is scheduled to be carried out. The pilot implementation described in this case thus began planning as early as in February for the pilot implementation to take place in May.

Patient-safety: Another issue regarding the successful pilot implementation is related to the fact that clinical work is sometimes a matter of life and death. It was agreed, although the design had proved itself at the IN-SITU workshop in January (figure 6), that unanticipated events leading to breakdown in communication or reports would have to result in a fall-back procedure where the organization returned to the paper record. This provided one of the requirements for preparation for the pilot; in case of fall-back, the end-users must be able to print a hard copy of the record. This was an issue for the technical configuration of the pilot implementation as print was not implemented on the framework version available for production. Further a work procedure was agreed upon between the project management and line-management. The projects had staff on location during the entire pilot implementation, and there was always a “duty-officer” on-call in case of events described by the fall-back procedure. Each morning and afternoon the project management made a status and could evaluate whether fall-back was required. The pilot progressed however without any incidents leading to a fall-back to the paper record.

Post pilot: What happens after the pilot? It is necessary for both the project and management in general to consider what is going to happen after the pilot, regarding the system and the work involved in Use. Does the project need more design iterations (post-pilot, figure 2) or is the pilot-system ready for full scale implementation (figure 6)? While the project may only consider implications regarding the pilot-system there are consequences when moving into the clinical domain. In clinical work, the record is a legal document regulated by law by the healthcare provider. It describes all clinical interventions performed. This means the project cannot scrap data created during the pilot and will have to consider either making a hardcopy or continue documenting in the system. The last option means the project has to be ready to keep up the momentum of the implementation, and continue to train and involve staff in the remaining organization.

In this case, the Hospital decided that if full scale implementation was postponed they would make hardcopies the records made during the pilot implementation to avoid historic information being inaccessible. In case of a successful pilot, they made plans for a rapid escalation and involvement of end-users and usability-actors resulting in the remaining staff being trained and the organization upgraded to full scale within 4-6 weeks after completing the pilot implementation.

TECHNOLOGICAL CONFIGURATION (figure 1)

One of the major concerns while using an agile process in the pilot implementation was how technically robust the framework was. To keep momentum and eventually measure effects, errors identified initially must be corrected immediately – otherwise an interrupted implementation will result in the expensively built end-user motivation being lost and the effects-driven process being terminated prematurely. In addition, reasonable changes in the framework design must be handled within the transition process (figure 7) without workshops in order to prevent interrupting the process. It is possible to ensure the robustness of a technological framework and to establish a prompt inclusion of immediate and emerging requirements, but it requires careful planning of available resources from CSC. It was anticipated that some minor changes were required to the framework.

To illustrate some of the challenges in preparing the prototype for the pilot we have adopted the ANSI/SPARC inspired model described in figure 4 to illustrate the evolution at different system levels from the mock-up, prototypes towards the pilot system (figure 7).

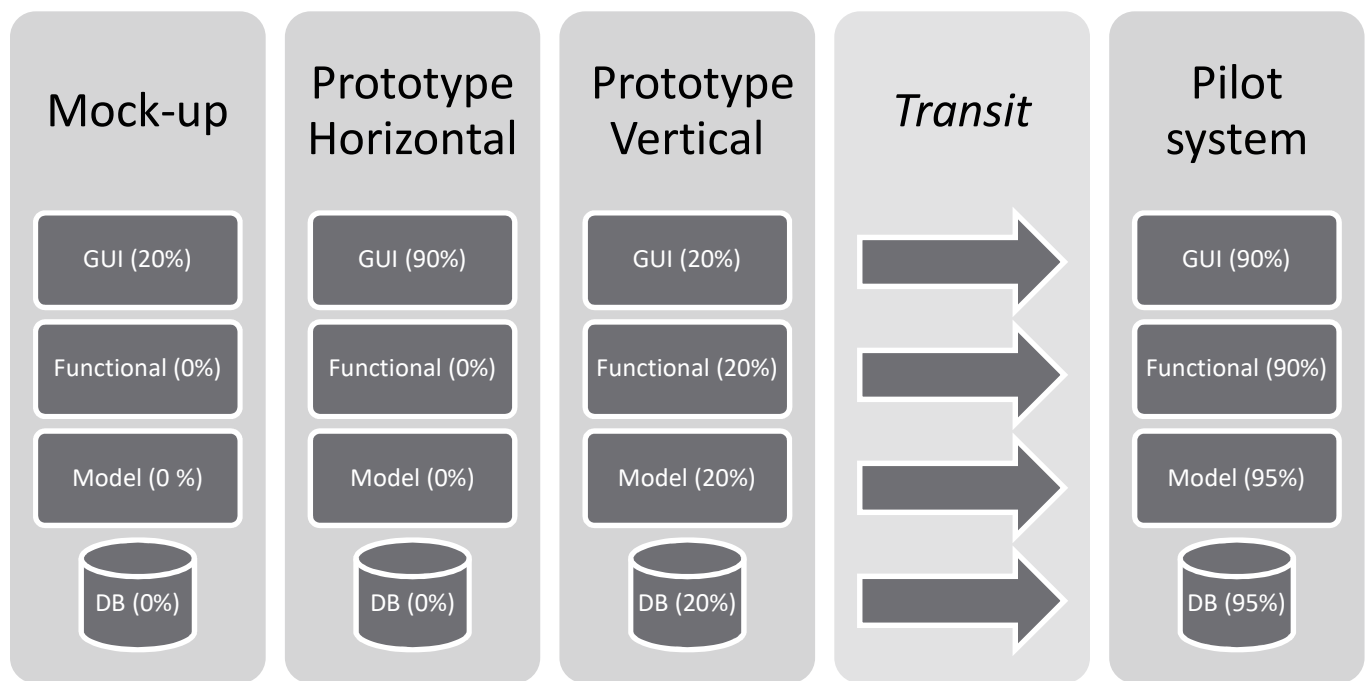


Figure 7: An example on the evolution of the different layers of the system during the process. The final transition required by the system from prototype to pilot-system, presents several challenges to be negotiated by the project organization.

The different degrees of completion are illustrated by exemplary percentages. In the Case several prototypes were made and compiling those made up the pilot system. The transition from prototype to pilot system is not trivial and involved much effort as each layer must be considered during the transition:

Two versions of Satellites visualizing the labour process

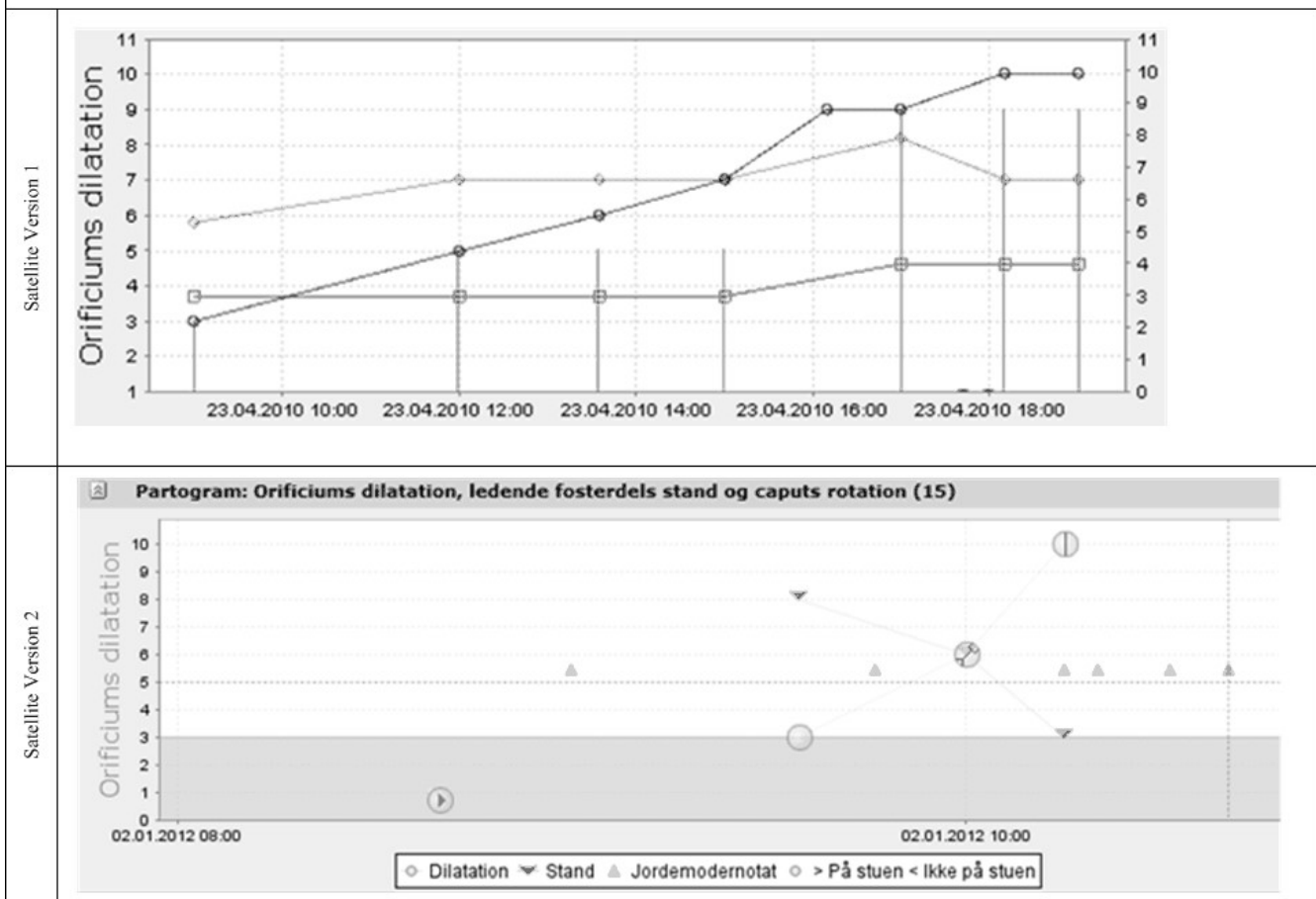


Figure 7: Satellite visualizing the labour process regarding the number and level of uterine contractions, expansion of the birth channel and position of the infant. The first version being a traditional graph drawing where the midwife must hover the mouse cursor over each entry to get a detailed reading. The second version incorporating a small illustration of the top of the infant's head allowing the midwife a direct reading of position and rotation.

Considering the GUI layer (figure 7):

The formative evaluation performed during the workshops resulted in the improvement of the solution presented to the usability-actors; see the examples of the satellites in figure 8 (version 2). The first version could convey the status of the birth to the midwives, but interpretation was slow and cumbersome. E.g. the dark line in figure 8 (version 1) shows the distance travelled by the infant in the birth channel, but it could not convey how the infant was rotated. This information had to be accessed in another satellite using a clock-dial analogy to convey the message of how the infant is rotated. E.g. 12-6 being in the middle of the birth channel facing upwards. The change in design was the result of a formative evaluation of the effects stating it should be easy and quick for the end-users to determine the progression of the birth of an individual pregnant on record. The new design shown in figure 8 (version 2) has a satellite illustration using the position and rotation of the infant's head as well as the distance travelled to convey the same message as in Satellite version 1. The icons are stylized representations of the top of the infant's head, and by looking at them and their progression upwards on the chart shows the midwife the progression of the birth in one piece of information.

Technical uncertainty should not become a disincentive to explore the functional innovation during implementation and use of the system. And if missing functionality is discovered in the technological framework

and this functionality is necessary to implement a requested effect, this must immediately be addressed by the project.

Considering the Functional layer (figure 7):

The technical framework CCS is highly configurable, but in order to provide this degree of freedom much effort must be invested in implementing the functional layer in CCS.

CCS framework evolves continuously and does not contain all anticipated functionalities. Identified functionalities are described in the roadmap for the development of the framework in the near future, say 1-2 years. When a workshop discovers a new requirement or raises a need for a specific service the project is left with three choices: (1) Make do with what is available, (2) try and make a workaround or (3) ask the development organization within the CSC to implement the requirements ahead of schedule. During prototyping, the end-users can be asked to envisage the missing functions or a lack of coherence in the prototype.

In order to make the apparently small change of allowing icons in satellites, development on the framework was required. This means the project has to take into consideration the unforeseen task of specifying and testing new functionality before it can implement the solution. To illustrate the extent of a change of the type we see in figure 8, a change in GUI may also require functional changes. Fortunately, no change was needed in the model layer. Consequently, a request for a functional-layer change was issued to the development team in CSC. Making this new functionality was not identified by the framework development project, and the pilot implementation was the sole owner of the request. Depending on the effort required to implement such requests, pilot projects risk not getting the solution they need. In this case, there was a lot of political interest as both Hospital and CSC management had an interest in the pilot implementation, and this made it possible to accelerate development within the pilot implementation time-frame.

Environment versioning is an issue as prototypes usually only mimics the real system. The system consisting of a technical framework and configuration elements means the configuration elements rely on functionality and background data such as terminology and classifications provide and managed by the framework. Moving the prototype from an environment where inaccurate terminology can be accepted towards a production grade environment means different rules for updates and another speed at which tasks can be completed. This increase of effort required for production grade configuration and background data, usually stems from activities such as quality assurance; design reviews and test or having to consider all three layers of the for coherence (figure 7).

While the pilot-project only included a small area of the hospital organization, and while the rest of wards depend on the existing record for their work, changes in terminology design may have a consequence on a larger scale than just the project scope. For example introduction of Problem classifications required the project to select and map relevant official diagnosis classifications with the terminology recorded in order to support the reporting requirements dictated by national law.

Another major task in the preparations for pilot implementation, and pre-pilot development is not confined to programming, but also relies on the organization being able to implement new framework functionality required to support the configurations made during design workshops. An example was the usability regarding the reading of notes which needed substantial improvement and the project had to choose if it would go for a hard-coded “display print” to support this need better or wait for the framework to implement the correct solution. As indicated in figure 7, the pilot-system has to come close to 100% in all layers in order to support real use as defined by the pilot implementation (figure 1). In this case, the project chooses to invest time and effort to configure a workaround, rather than wait for the framework to present an easy solution. Special overviews were made for the sole purpose of serving as print templates; they provided readability improvements otherwise scheduled for later releases of the CCS framework.

Considering the Model layer (figure 7):

The final layer of the system is the model layer and database. Moving from prototype to the pilot-system environment means the solution must co-exist with the rest of the IT portfolio. An example was the terminology implemented with the pilot system; the EHR is a distributed system, which means that it communicates with many other systems. For example, the records made by the midwives contain reports of clinical procedures; these procedures end up in national databases and are used in calculating performance and funding of the Hospital. In the case, CSC and the Hospital had to invest the time and effort in finding the correct terminology and load it into the CCS for the pilot system to operate on equal terms with the remaining systems in the

Hospital portfolio.

ORGANIZATIONAL ADAPTATION (figure 1)

The task of ensuring the correct terminology led the project to a revision of work procedures and new praxis of registration and reporting. It was discovered that a considerable amount of redundant registration was carried out because communication between local and national databases could not be mapped.

A spin-off was the discussion initiated among the staff regarding the documentation model that had been adopted by the small group of midwives directly involved in the project. It concerned a large overview with a graphical representation of the progression of the birth. The illustration in figure 8 is the electronic version of it. This model was developed by midwives for midwives but had not been fully assimilated into the organization. The design was made from the principles outlined in this model and end-users were required to adhere to it in order to benefit from the effects of use. The hospital management was concerned that it could spark unrest when the group of staff that had not been directly involved were going to use the system or later when Site 2 was to implement the EHR. This was one of the scope goals for knowledge set out in the beginning of the pilot implementation by the Hospital and CSC.

Another aspect of adaptation was the physical facilities surrounding the midwives. They were located often in small rooms with the woman giving birth, the father, sometimes a doctor and often bulky equipment (monitors etc.). This meant ergonomics and issues of network access had to be raised and addressed by management and local IT support. Another related issue was the training of the staff, as the final version of the pilot system was not ready until the week before the pilot implementation. Staff had to be trained in the actual version to understand the documentation model in order to have sufficient skill with the pilot system without having to call hotline too often. Training was carried out by two members of the project staff in classroom sessions of 1½ hours, and the staffs were issued a short-list to carry with them as a first resort when in doubt.

Post-pilot consideration mentioned in planning should include choosing a part of the organization which is well defined and preferably can be contained in case of fall-back or escalating the implementation. In this case, the midwives were chosen because they had a well-defined area of responsibility with regards to patients (pregnant women), and relations to other units at the Hospital. Escalation was relatively easy to plan and execute since there are only one main group of staff (midwives) and the doctors were a small and specialized group. In contrast, most other wards are staffed with nurses, doctors and various therapists.

In this case, the planning had included two scenarios: fall-back to the paper record or escalation of the EHR to the rest of the staff. The later scenario was the outcome of the pilot implementation and two months later when site 1 was at full scale the implementation team moved to site 2.

USE (figure 1)

The goal of the pilot implementation was to allow the staff to perform “business-as-usual” with the exception of records being made entirely in the EHR. The midwives at the hospital performed the usual amount of births during the pilot without extra staff and without needing to apply the fall-back procedure for reverting to paper.

During the pilot implementation, hospital management insisted on support staff being readily available. The project had called upon project participants and staff from the remaining implementation organization to perform the role of an on-site hotline service. A hotline roster had been prepared to ensure a person with clinical knowledge and technical skills was available at any time during the pilot. In addition, a person with the role of managing a potential fall-back and being able to stop everything had to be designated and to be on call.

It was the responsibility of the Hospital to ensure that the staff used the EHR as planned. They were not allowed to keep a paper record for private use. Further they had to use the system as it was described by the design. E.g. if they wanted to learn the status of the birth they had to open the EHR and read it there before seeking additional information elsewhere. On a larger scale, the staff at the outpatient clinics had to record their work in the EHR since their colleagues relied on the information when reading about the pregnant women when they arrived at the maternity ward. If the staff failed to support the pilot implementation in this way it would have been impossible to measure the effects of sharing information among maternity and outpatient clinic.

LEARNING (figure 1)

Three groups of end-users have the possibility to learn from the pilot implementation experience. They each has their own perspectives and areas of responsibilities spanning from across the organization to practical issues and tasks.

The three groups are;

- Hospital management – senior management representing local management across the Hospital organization, responsible for budget and strategy implementation at the political level.
- Hospital IT configurators responsible for the project deliverables, planning and configuring the actual system.
- End-users; midwives from the entire organization and clinical staff working with patients on an everyday basis.

The knowledge they can expect to acquire is defined in the scope for the pilot implementation including support of a clinical context, if change is necessary when implementing at a new location, whether the Hospital organization achieve independency of the CSC configurators and finally how can EDIT support the overall process?

Hospital management

The EHR platform as a Hospital organization spanning project is a complicated undertaking. It involves the entire organization and the primary concern of the senior management was to see whether the CCS platform could support clinical work without compromising patient safety and could it do so within the budgets and staffing of the business unit (Maternity). Secondly there was a concern that local differences in work organization and practices would require adaptations that were costly in both time and effort each time a new location implemented the system. The pilot implementation answered these concerns by example; although staff was kept on call in the event that recordkeeping proved more time consuming, the amount of staff involved at the maternity ward and the outpatient clinics was not increased, neither was the intake of pregnant women admitted or referred for consultation reduced. The business units operated at normal capacity during the pilot implementation. Patient safety was not compromised as the project management did not use the fall-back procedure but recommended the continued implementation after the pilot first at Site 1, and later at Site 2.

When implementation was conducted at Site 2, the Hospital management was assured that the concerns for huge costs due to local demands for configuration were unfounded. Again the evidence for these claims is that production was not interrupted, and patient safety was not compromised, and no problems were reported regarding the adaptation of the documentation model supported by the design.

In contrast to previous learning experiences, the strategy involves that effects evaluation must be undertaken to ensure “*Effective utilization of IT in the region*” and the effects specifications used during design and development was used to measure the effects of use by the end-users. This documented not only the individual user experience but also the aggregated experience of a group of users with regard to how they rate the effectiveness of the IT implemented during the pilot.

Hospital IT configuration unit is charged with the responsibility of implementing the CCS EHR and business strategy in practice.

From their point-of-view, the greatest learning experience from the pilot implementation is that of putting a system into production as they have been concerned with the prospect of upgrading 6000+ users from paper to IT in an extremely short period of time. Their implementation strategy relies on the scalability of design and the ability to start using the system in rapid successions throughout the Hospital.

The Hospital IT configuration unit had the opportunity to try real implementation with all the safety features available. They experienced confidence because they knew exactly what the pilot system could do and not do as they had themselves configured it from the beginning with the end-users.

By using effects, they had the opportunity to receive formative feedback from the end-users in a structured way and keep the design effort focused on the priorities agreed upon from the beginning of the project.

Secondly they could experience learning as the vendor-customer responsibilities shifted due to the Hospital business strategy demanding more “*Coherence in services ...*”, “*Innovations as a factor in creating growth and an attractive workplace*”, “*Ensuring ownership....*” This was in part made possible due to the nature of the CCS and through the initial organization of work in the project team. The Hospital IT configuration unit was

responsible for the project process as well as for the configuration of a design that met the effects specification. This was a new role for the business unit as it usually had a consulting role, and left the project management and development to CSC. In a way, the Hospital IT configuration unit was being an apprentice as a vendor, learning to reflect on all levels of the system (figure 5) to end up with a usable pilot system.

A final concern was that professional differences among the midwives could result in resentment towards the system since it had been designed at the university clinic (Site 1) and transferred unchanged to another group of midwives without direct representation from this group during design and development (Site 2). The risk of the end-users having a strong sense of professional integrity combined with a feeling of being left without influence in matters of professional values was a potential show-stopper. This would be the case if the local midwives at Site 2 could not accept the documentation model devised by the University group at Site 1. During the introduction and training, the midwives referred to this concern in various terms, by saying; “It is better to have this kind of structure rather than the old mess where it was up to everyone to decide what was relevant to document” or “I can see my profession in this and that is what is pertinent to my work” (Interviews with midwives during preparations for Full scale implementation Site 2, figure 6). Further evaluation of use at Site 2 showed no significant difference in the effectiveness or quality of service due to the new recordkeeping practice.

End-users

When we look at the end-user group, there are 2 significant learning opportunities: a technology related and a work related.

The most important aspect of work, when dealing with clinical staff, is the patient–professional relation. It is not the technology that motivates and drives the staff while delivering service. At best the technology involved in performing clinical work assumes a discrete presence, and in some cases, it is an endured necessity. In either case end-users had the opportunity to become familiar with the standardized documentation model, but also doing it in a new technology. In a way, they made two steps at once, learning and working with the Partogram could have been introduced on paper before they implemented the CCS. However as the Hospital IT configuration unit in charge of the planning and design experienced the absence of negative effects from using the CCS, they decided to make the implementation at Site 2 without trying on paper first.

The end-users saw that business (clinical work) could be performed as they were used to, and they did not experience issues threatening patient safety or causing a reduction in the quality of service. It was not without extra effort involved, as staffs were required to attend training before they could use the CCS effectively and in accordance with the design.

The effort experienced, besides training, was that they had to seek out a PC when they needed information or wanted to document their work. For example, at the maternity ward there is a national quality indicator stating that a midwife must be present continuously during the active phase of delivery. In the event that the midwife has to leave for more than 15 minutes, she would not adhere with this indicator, so she must quickly and continuously be able to document her presence. In the paper record, this was easily done by marking absence on a timeline, but in the CCS, they had to make a record of leaving by entering the time manually as they left the room and again when they returned. It was not possible to make an exception from this requirement to document quality and since there was no easy way of supporting this work practice, the end-users experienced extra effort involved in using the CCS during work.

Another learning experience was observed during discussions on how to adopt the standardized documentation model. This was undertaken during training or ordinary staff meetings, and it was emphasized that the midwife could carry out the same work they used to, and they had been represented during design by highly qualified senior midwives. The effects evaluation supported the experience by acting as a facilitator between individual opinions by offering a shared statement of the end-users experiences. E.g. if one midwife would say “I feel insecure relying on the CCS for information”, the effects stated that the majority did not share this experience. This made the discussion focused on what the end-user was missing rather than starting a discussion on shortcomings of the technology.

CURRENT CHALLENGES FACING THE ORGANIZATION

It seems two potential and major challenges related to technical configuration and organizational adaptation were absent in the case: the need for local design and development of the CCS to support staff at Site 2, and there was no un-willingness with the midwives at Site 2 to accept a solution made by the midwives at Site 1. This does not mean that pilot implementation is the solution to all challenges among CSC and the Hospital. During the pilot implementation, CSC and the Hospital experienced that the role of apprentice and mentor was more complex than anticipated. The CCS has been envisioned as a standard tool, a framework in which technical staff with a relatively basic understanding of technology can develop the system required by the end-users.

The agile process adopted (figure 2) by the project made it necessary to work intensely between workshops. These bursts of peak activity with high workload challenged the Hospital. The Hospital IT configuration unit, although skilled in organizing projects, was not staffed and organized to cope with both the continuous support of other projects and the allocation of large tasks with critical deadlines. In addition, they had to learn technical skills as they went along as the CCS came in new versions at regular intervals, making new functionality available or setting the conditions for design differently. For example, readability is essential when clinical staff scans records for clues about the problems experienced by the patients.

The amount of work involved in design and planning was underestimated, and CSC was asked to assume responsibility for the configuration tasks required to develop the pilot system. It was not difficult to discover what to develop. The challenge was production coordination related to configuration of the pilot system. The Hospital IT configuration unit had to learn how to coordinate configuration tasks as a vendor rather than as a customer. Being responsible for planning meant estimating work tasks in order to monitor if the plan is on schedule and making progress. In order to make accurate estimates, experience with the task anticipated is needed and keeping the project on track requires the discipline of follow-up on progress and prioritizing when necessary to reach deadlines. This also put the CSC's strategy for client intimacy into perspective.

Another new experience was setting up the facilities of quality assurance. Being responsible for developing and managing meant that the Hospital IT configuration unit had to assume responsibility for prioritizing the implementation of changes discovered during workshops, but also internal testing and correction of errors discovered during internal work processes. Normally CSC would perform a system and delivery test before handing over the system for acceptance, usually only involving clinical staff as technical issues should be solved by the time the system reaches the end-user representatives. Again the pool of available and skilled staff was underestimated, and management of resources versus completion of tasks was a new experience for the Hospital IT configuration unit staff.

After the pilot implementation, the Hospital IT configuration unit was upgraded with staff skilled in both project management and configuration.

SOLUTIONS AND RECOMMENDATIONS

The difference between pilot implementation and prototyping can be seen in challenges raised by the different aspects motivated by the desire to evaluate the fit between simulated and real-use.

Technical configuration aspects raise the issues defined by the transition from prototype to pilot system. The project assumes responsibility for either substituting missing functionality in the technical framework or qualifying the system. This must be to a completion level where work can be performed sufficiently close to the intended use in the pilot-system without compromising patient-safety.

Use implies that usability-actors cannot be limited to specific work situations, e.g. using the CCS for one type of tasks and paper for another. The pilot implementation must support the work specified by the end-users and the project management can benefit from contemplating a reduction in complexity when selecting the business unit they want to target for learning. This means pilot implementation requires a relatively mature technical framework and a skilled project team to plan and monitor the elements of both organizational adaptation and technical configuration.

The Hospital wanted to learn how to make pilot implementation and chose the midwives for the pilot implementation because they form a homogeneous group of skilled professionals. Their work is defined by one particular group of patients (pregnant women), and they have been working with this clinical domain for a very long time. In addition, they only interact directly with doctors on well-defined tasks. For example, the midwife attends the patients independently from the beginning in the outpatient clinic to delivery at the maternity ward. If

complications introduce a need for treatments that are outside the midwife's competences, she contacts the doctor on call. This means that the pregnant woman can carry her child and give birth without seeing other hospital professionals than the midwife. She does however, see her General Practitioner (GP), but he or she only shares information with the midwives on a need-to-know basis. This information is typically entered in a small dedicated record; the "Pregnancy record" (Hertzum, et al., 2012); the midwives and GPs do not share information or coordinate work directly, although both groups use the EHR in their individual work. This may be the reason why a similar project aimed at sharing information among these two independent professions (i.e., midwives and GPs) turned out unsatisfactorily in the case described by (Hertzum, et al., 2012). This project involved a heterogeneous group of professionals with a pilot implementation as a way of investigating the collaboration and coordination between the groups.

Another recommendation for pilot implementations is the ability to assess and negotiate risks for first-movers, both when spanning different geographic locations in large organizations and when dealing with mission critical systems. All three groups of end-users had concerns regarding these issues and benefitted from the approach utilizing pilot implementation during the process.

REFERENCES

- Barlach, A., & Simonsen, J. (2011). Innovation in Partnership Sourcing from a Vendor's Perspective. In M. Hertzum & C. Jørgensen (Eds.), *Balancing Sourcing and Innovation in Information Systems Development*. (pp. 193-212). Trondheim, NO: Tapir Academic Publishers.
- Bødker, K., Simonsen, J., & Kensing, F. (2009). *Participatory IT design : designing for business and workplace realities*. Cambridge, Mass.: MIT.
- Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A STRATEGY FOR COMPARING ALTERNATIVE SOFTWARE-DEVELOPMENT LIFE-CYCLE MODELS. *Ieee Transactions on Software Engineering*, 14(10), 1453-1461. doi: 10.1109/32.6190
- Floyd, C. (1984). A systematic look at prototyping. In R. Budde, K. Kuhlenkamp, L. Mathiassen & L. Zullighoven (Eds.), *Approaches to Prototyping: Proceedings on the Working Conference on Prototyping* (pp. 1-18). Heidelberg: Springer Verlag.
- Hainey, T. (2007). Information systems development methodologies, techniques & tools, 4th edition. *International Journal of Information Management*, 27(1), 58-59. doi: 10.1016/j.ijinfomgt.2006.09.002
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index) - Results of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload*: Elsevier Science Publisher B.V.
- Hertzum, M., Bansler, J. P., Havn, E. C., & Simonsen, J. (2012). Pilot Implementation: Learning from Field Tests in IS Development. *Communications of the Association for Information Systems*, 30(1), 313-328.
- Jansson, L., Handest, P., Nielsen, J., Saebye, D., & Parnas, J. (2002). Exploring boundaries of schizophrenia: a comparison of ICD-10 with other diagnostic systems in first-admitted patients. *World Psychiatry*, 1(2), 109-114.
- Jureta, I. J., Faulkner, S., & Schobbens, P. Y. (2008). Clear justification of modeling decisions for goal-oriented requirements engineering. [Article]. *Requirements Engineering*, 13(2), 87-115. doi: 10.1007/s00766-007-0056-y
- Nielsen, J. (1993). *Usability engineering*. Boston: Academic Press.
- Rasmussen, J., Pejtersen, A.-L. M., & Goodstein, L. P. (1994). *Cognitive Systems Engineering*: Wiley.
- Region North Jutland Consolidated. (2011). *IT strategy 2014*. Niels Bohrs Vej 30 9220 Aalborg Oest.
- Royce, W. W. (1987). *Managing the development of large software systems: concepts and techniques*. Paper presented at the Proceedings of the 9th international conference on Software Engineering, Monterey, California, United States.
- Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice Hall.
- Simonsen, J., Hertzum, M., & Barlach, A. (2011). Experiences with effects specifications. In M. Hertzum & C. Jørgensen (Eds.), *Balancing Sourcing and Innovation in Information Systems Development*. (pp. 145-163). Trondheim, NO: Tapir Academic Publishers.
- Stacey, M., & Eckert, C. (2003). Against Ambiguity. *Computer Supported Cooperative Work (CSCW)*, 12(2), 153-183.
- Stapleton, J. (1998). *DSDM* (Repr. ed.). Harlow: Addison-Wesley.
- Tsichritzis, D., & Klug, A. (1978). ANSI-X3-SPARC DBMS FRAMEWORK - REPORT OF STUDY-GROUP ON DATABASE MANAGEMENT-SYSTEMS. *Information Systems*, 3(3), 173-191.
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *Mis Quarterly*, 27(3), 425-478.
- Vicente, K. J. (1999). *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work*: Lawrence Erlbaum Associates, Inc.

KEY TERMS & DEFINITIONS

Pilot implementation: Pilot implementation is an Information Systems Development (ISD) technique that aims to feed experiences from real use back into development by having users try out a system on a restricted scale before the design of the system is finalized. Pilot implementation is a field test during which a pilot system is used in its intended environment with real data. Pilot implementations are conducted to learn about how a system may support its users in their work and, thereby, to create information and insight about how to improve the system, adapt the organization, and capture the benefits of introducing the system in the organization. By providing feedback from target-environment use of the system to the on-going development activities, pilot implementation supplements prototyping, which in most definitions is restricted to the development phase. (Hertzum, et al., 2012)

EHR: Electronic Healthcare Record, a tool for planning and documenting clinical work, facilitating collaboration among clinical staff towards common patient related health goals.

Patient-safety: The considerations involved in preventing adverse effects from clinical interventions either directly or indirectly involving the patient. E.g. abstaining from performing specific actions because they pose a risk of harming the patient to an extent that cannot be justified by the anticipated benevolent effects from the action.

Effect: The anticipated clinical outcome of work performed by an agent typically being a person (doctors, nurses, ect.) in rare instances the agent can be a technology (drugs, ect.).

Technical framework: A set of electronic building blocks in which prototypes and applications can be configured by combining relevant features and functionalities offered by the framework to a specific customer.

Prototyping: The use of prototypes to facilitate learning during the software development process. The prototypes are the main artefact to illustrate the interpretation of user requirements and feedback from stakeholders.

Learning: The process of acquiring or modifying information into knowledge. In this case knowledge is applied to the design and implementation of an EHR.