



Københavns Universitet



Pilot implementation

Hertzum, Morten; Bansler, Jørgen P.; Havn, Erling C.; Simonsen, Jesper

Published in:
Communications of the Association for Information Systems

Publication date:
2012

Document Version
Peer reviewed version

Citation for published version (APA):
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.

Pilot Implementation: Learning from Field Tests in IS Development

Morten Hertzum¹, Jørgen P. Bansler², Erling C. Havn³, and Jesper Simonsen⁴

¹ Roskilde University, Roskilde, Denmark, mhz@ruc.dk

² University of Copenhagen, Copenhagen, Denmark, bansler@diku.dk

³ Technical University of Denmark, Lyngby, Denmark, havn@man.dtu.dk

⁴ Roskilde University, Roskilde, Denmark, simonsen@ruc.dk

Abstract. A recurrent problem in information-systems development (ISD) is that many design shortcomings are not detected during development, but first after the system has been delivered and implemented in its intended environment. Pilot implementations appear to promise a way to extend prototyping from the laboratory to the field, thereby allowing users to experience a system design under realistic conditions and developers to get feedback from realistic use while the design is still malleable. We characterize pilot implementation, contrast it with prototyping, propose a five-element model of pilot implementation, and provide three empirical illustrations of our model. We conclude that pilot implementation has much merit as an ISD technique when system performance is contingent on context. But we also warn developers that, despite their seductive conceptual simplicity, pilot implementations can be difficult to plan and conduct. It is sometimes assumed that pilot implementations are less complicated and risky than ordinary implementations. Pilot implementations are, however, neither prototyping nor small-scale versions of full-scale implementations; they are fundamentally different and have their own challenges, which will be enumerated and discussed in this article.

Keywords: pilot implementation, pilot system, information-systems development, prototyping, socio-technical change

I. INTRODUCTION

It has long been acknowledged that information-systems development (ISD) is a *learning process* that requires an iterative, incremental approach with feedback loops between successive development stages, such as requirements definition, systems design, coding, test, implementation, and operation [Boehm, 1988; Brooks, 1987; Floyd, 1984; Larman and Basili, 2003]. This insight goes back at least to the 1970s and 1980s when the ideas and concepts behind iterative, incremental, and evolutionary development were first introduced and discussed widely [Larman and Basili, 2003]. In recent years, interest in these ideas has been revived with the rise in popularity of agile methods [Abrahamsson et al., 2003; Williams and Cockburn, 2003]. Even though iterative development approaches have become widely accepted and used, it has proven difficult to obtain sufficiently realistic feedback about system use. For instance, while various prototyping techniques may provide feedback from simulated use in a laboratory environment, they do not provide insight into real user experiences with the system in action [Davis, 1995]. This is problematic, because many flaws in the design of a system and its organizational implementation do not surface until the system is in real use. Furthermore, as Boehm [2000, p. 99] has pointed out, “users may initially feel that they ‘know it’ when they see an initial demo or prototype. But their needs and desires change once they begin operating the system and gain a deeper understanding of how it could support their mission”.

In this paper, we focus on systematic field tests, called *pilot implementations*, to create a feedback loop from implementation to development in ISD projects with high organizational complexity, innovation

needs, or cost of failure. We define a pilot implementation 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*. Correspondingly, we term the properly engineered, yet unfinished system a pilot system. By subjecting a pilot system to real-use conditions before the system design is finalized a pilot implementation provides developers with feedback on how users actually experience working with the system. Pilot systems have much in common with prototypes, and proponents of a broad definition of prototypes may see pilot systems as a subclass of prototypes [e.g., Floyd, 1984]. We prefer to distinguish pilot systems from prototypes to emphasize that pilot systems are properly engineered and evaluated in their intended environment with real data [Rzevski, 1984]. At the same time, pilot systems are not finalized and are therefore restricted to limited implementation. Pilot implementation is a technique for use in ISD projects; it is neither a full ISD model nor an approach to full-scale implementation.

Pilot implementations are compatible with both plan-driven and evolutionary ISD approaches. In plan-driven approaches [Boehm, 2002; Boehm and Turner, 2004], a pilot implementation near the transition from development to implementation may provide reassurance that current development activities simply need to be finalized or it may show that revision of previous development activities is necessary. In evolutionary approaches [Basili and Turner, 1975; Floyd, 1984; McCracken and Jackson, 1982], which break a project into a series of successively completed subprojects or ‘increments’, experience from use is implicit in each increment and provides feedback important to the planning and content of subsequent increments. While an evolutionary approach reduces the risk of completing development without receiving real-use feedback [Alter, 2001; Boehm, 1988], we contend that an improved understanding of pilot implementations will also benefit evolutionary approaches because it provides a more explicit analysis of the challenges involved in learning from real-use experiences.

Unlike prototyping, pilot implementation is still an under-researched concept. Although many ISD projects have used pilot implementations to evaluate a proposed system’s capabilities and limitations [Lin and Pervan, 2003; Ward et al., 1996], little has been published about what pilot implementations are, why they are considered useful, and how they should be conducted. The purpose of this paper is to discuss the characteristics of pilot implementations in a systems-development context that involves new product development, as opposed to the configuration and implementation of off-the-shelf, so-called COTS, systems. We consider whether and how pilot implementations can extend prototyping from the laboratory to the field. In addition, we will discuss the key issues and challenges associated with conducting pilot implementations in practice. In doing so, we will draw on the existing, but rather sparse, literature on pilot implementation as well as our own experiences from three studies of pilot implementation in the Danish healthcare sector.

The article is structured as follows. First, we cover related work on pilot implementation. Second, we analyze the characteristics of pilot implementation, compare and contrast them with prototyping, and provide a model of the elements that constitute pilot implementations. Third, we elaborate the different elements of pilot implementation with three empirical illustrations. Fourth, we discuss challenges in planning and conducting pilot implementations and identify implications for research and practice.

II. RELATED WORK

Pilot implementation may be a valuable ISD technique if and when real-use feedback is important to the successful completion of development activities and difficult to obtain in other ways. Below, we briefly address the inadequacy of real-use feedback in widespread ISD approaches and then account for related work on the use of pilot implementation to avoid such inadequacy.

Inadequate real-use feedback

Markus [2004] argues that the risk that people will not adopt and use information systems and associated work practices has not been adequately addressed by the research on how to manage ISD projects. Along similar lines, Ward and Daniel [2006] propose an increased focus on real use by incorporating data about the benefits obtained from system use in their project model. They argue that such data are often absent, yet valuable. Thus, it appears that many ISD projects are completed on the basis of limited knowledge about real use of the resulting system. Feedback from real-use experiences is, however, important for several reasons. First, developers and users need a deeper understanding of how the system may support the users’ work and what they require from the system [Boehm, 2000; Rzevski, 1984]. Second, real-use experiences are necessary in order for organizational consequences

and opportunities of the system to emerge, especially in complex organizational settings where systems affect multiple interrelated stakeholder groups [Bossen, 2007; Orlikowski, 1996]. Third, real-use feedback makes it possible to gauge the extent to which envisioned benefits are obtained and potentially make project completion and success dependent on this [Ward and Daniel, 2006].

In his discussion of life-cycle models Alter [2001] divides the ISD life cycle into four phases: initiation, development, implementation, and operation. By mapping various ISD models to these four phases, Alter identifies limited coverage of post-development issues in several of the models, for example the waterfall model [Boehm, 1981]. For the life-cycle model that represents “a typical project that might confront a software development manager” Alter concludes that “this model ends before implementation in the organization begins” [Alter, 2001, p. 27]. This suggests that development activities are informed by little or no real-use feedback.

Agile approaches such as Crystal [Cockburn, 2007], Dynamic Systems Development Method [Stapleton, 1997], and Scrum [Schwaber and Beedle, 2002] emphasize customer collaboration over contract negotiation, responding to change over following a plan, and other similar values. However, agile methods offer little guidance on how to conduct activities that give users a basis for providing feedback [Abrahamsson et al., 2003]. Thus, while agile methods advocate evolutionary development, they appear to lack concrete means of obtaining systematic feedback from the implementation of one release to the development of subsequent releases. A risk of inadequate real-use feedback is therefore also present in agile methods.

Pilot implementation

For our purposes it is important to distinguish between two related, but different, kinds of pilot implementation represented below by Rzevski [1984] and Janson [1986]. Rzevski discusses pilot implementation in relation to ongoing development activities and is an important inspiration for our definition of pilot implementation. Conversely, Janson describes pilot implementation as a post-development activity, which concerns the selection and implementation of standard (i.e., off-the-shelf) systems. This focus on standard systems discords with our focus because it does not involve feedback to development.

Rzevski [1984, p. 362] defines pilot systems by distinguishing them from prototypes: “In contrast to prototypes, pilot systems are computer-based systems properly designed and engineered, and therefore reliable and robust, offering only a small subset of facilities of the system under development. Pilot systems are designed to be gradually extended into full operational systems. Their projected useful life is much longer than that of prototypes.” Like pilot plants in chemical engineering, pilot systems are built to provide opportunities for experimentation and evaluation in a real-world setting while the system is still under development. The purpose of implementing pilot systems – that is, of pilot implementation – is “for both developers and clients to learn from each other and about each other and to incorporate all this knowledge into further parts of the main system” [Rzevski, 1984, p. 362]. That is, pilot implementations are not confined to the development phase but reach into implementation activities to feed real-use experiences back into the ongoing development activities.

Janson [1986] discusses pilot implementation in relation to standard systems and defines its main objective as “allow[ing] the user to verify that a standard software package meets the organization’s information needs, with at most only minor changes” (p. 210). Here, pilot implementation is confined to the implementation phase with its focus on organizational issues. Janson’s supplementary objectives of pilot implementation emphasize this focus by comprising observation of organizational reactions to the pilot system, detection of possible resistance to the system, provision of better training, and formulation of better implementation strategies. Janson defines a pilot system as a scaled-down version that offers a subset of the capabilities present in the full system without sacrificing robustness, completeness, and reliability. These characteristics “make it possible to install the pilot system in its intended environment and enable the user to test the proposed application package using normal data, under regular operating conditions, and at greatly reduced cost” [Janson, 1986, p. 210]. An important aspect of such pilot implementation is that “the pilot system functions under the user’s control” [Janson and Hammerschmidt, 1990, p. 48]. This is seen as contrary to prototypes, where the designer retains control of the situation.

It is notable that neither Rzevski [1984] nor Janson [1986] provide detail about how to conduct pilot implementations. Research on pilot implementation reports a scarcity of such considerations [e.g., Glass, 1997; Pal et al., 2008; Turner, 2005], and we are unaware of research that provides such details for pilot implementations performed while development is still ongoing. For post-development pilot

implementation, Glass [1997] proposes a detailed set of steps intended “to produce sufficient information to allow a decision to be made about the implementation and use of the pilot concept” (p. 87). With this definition pilot implementations, which Glass terms pilot studies, are decoupled from development and involve only a customer organization that considers adopting a system. The post-development focus is also reflected in the proposed set of pilot-study steps, which comprises five main steps: planning (6 sub steps), design (12 sub steps), conduct (4 sub steps), evaluation (10 sub steps), and use (3 sub steps). For example, the design step is entirely about preparing the organization for using the system and about devising ways of obtaining data about its use. None of the 35 sub steps are about fitting the system to the organization or about feeding insights from the pilot implementation back to development. The set of steps was used for retrospectively evaluating more than 20 pilot implementations conducted by a government agency. This empirical evaluation shows the steps to be perhaps too rigorous [Glass, 1997]. Other models for post-development pilot implementation of information systems contain fewer steps but are similar in spirit [e.g., Pal et al., 2008].

Turner [2005, p. 2] talks about pilot studies, which he defines as “part of a larger project or programme, undertaken to improve understanding of the main change or innovation being delivered by the project or programme, thereby reducing the risk and uncertainty associated with the change”. This definition emphasizes two points that appear important to pilot implementations that feed back into ongoing development activities. First, the definition makes the reduction of risk and uncertainty central to pilot implementation. Second, the definition emphasizes that a pilot implementation is conducted in the context of a larger activity. For pilot implementations that feed experiences back into ongoing development activities, this larger activity is the ISD project.

The use of pilot implementations to assess the viability of new technologies prior to full-scale implementation appears to be a common practice [e.g., Babar et al., 2006; Iredale et al., 2002; Liang et al., 2006; Pal et al., 2008; Reed et al., 2004]. Surveys of large Australian and UK organizations report that 81% and 87%, respectively, of respondents conduct pilot implementations when implementing information systems [Lin and Pervan, 2003; Ward et al., 1996]. Just as there is little prescriptive work about how to conduct pilot implementations, especially while development is still ongoing, there is little descriptive work of what pilot implementations look like in practice – focus is instead on the outcome of the pilot implementations. When pilot-implementation methodology is discussed in the literature it is often to acknowledge certain limitations. For example, Reed et al. [2004] acknowledge the limited duration of the pilot implementation as a limitation of their work because it did not allow the novelty of the system to wear off, Liang et al. [2006] acknowledge that the absence of an appropriate baseline against which to compare pilot-implementation results makes the results indirect, and Babar et al. [2006] discuss threats to the representativeness of their pilot-implementation participants.

III. PILOT IMPLEMENTATION AS AN ISD ACTIVITY

We focus on the use of pilot systems while development is still ongoing. Such pilot-system implementation or, for short, pilot implementation is summarized in Table 1 by contrasting it with prototyping.

Characteristics of pilot implementation

Pilot implementations can be performed in manifold ways but involve trying out a system on a restricted scale before finalizing the design of the system. Four characteristics of pilot implementation warrant further clarification: First, pilot systems are by definition not fully developed. In contrast to prototypes, pilot systems have considerable functionality and are sufficiently robust, reliable, and efficient to enable their implementation and use in a real work environment [Rzevski, 1984]. In contrast to releases of system versions, pilot systems are not finalized and must therefore be expected to lack some functionality and to malfunction and break down occasionally. Thus, a pilot system is only suited for limited implementation because it requires special precautions.

Second, a pilot implementation is limited in scope and time. One or a few sites are selected for the pilot implementation, and the experiences gained at these sites inform the subsequent development and implementation activities. Ideally, a pilot implementation should be long enough for learning curves to flatten, new work procedures to stabilize, and the effects of the use of the pilot system to materialize. Jurison [1996] finds that effects at the level of individual users can be observed within 6-8 months whereas effects at the organizational level may take a year to materialize. Concomitantly, a pilot

implementation must be short enough to fit within an ISD project. This tension may necessitate brief pilot implementations in some situations.

Third, a pilot implementation is conducted in the intended use environment using real data but it is part of an ISD project. As the pilot system is not yet fully developed and it is only used at some sites, the use of the pilot system may entail certain restrictions, for example in interactions with sites that are not part of the pilot implementation. Such restrictions may be handled through careful explanation of the purpose of the pilot implementation or through temporary interventions such as simulations. The timing of pilot implementations involves balancing the size and nature of the restrictions imposed on the pilot implementation against the desire for early feedback from real use of the system. By defining pilot implementations as a technique for use in ISD projects, this feedback is to inform development as well as implementation.

Fourth, pilot implementations are conducted to learn about the fit between the system and its use environment and thereby to explore the value of the system, improve or assess the system design, and reduce implementation risk. Importantly, the purpose is to support the ISD project. That is, if the pilot system succeeds in supporting users in accomplishing their work, this is primarily valuable as feedback to the project about the quality of the system and only secondarily because it improved real work during the pilot implementation. More specifically, the purpose of a pilot implementation can be any of the following:

- To evaluate the usefulness and usability of a system in order to inform a decision about whether to continue the development of the system
- To improve the system design based on user feedback, experience from practical use, and performance measures such as productivity or quality data
- To identify necessary or desirable changes in the work organization and processes in which the system will be embedded
- To become aware of unanticipated change that emerges from using the pilot system and may call for preventive action to avoid unwanted change or supportive action to sustain desired change
- To provide input for formulating implementation strategies and plans, on the basis of users' reactions to the pilot system.

The major purpose, then, of a pilot implementation is to learn about how the system performs in a real environment and how users appropriate and use the system. Pilot implementations provide a means for developers and users to explore the affordances of the system and experiment with its integration into and transformation of existing practice. Experiences from these activities are fed into the finalization of the technical development of the system and into preparing full-scale organizational implementation.

Pilot implementation versus prototyping

The purpose of prototyping is, partly, to traverse a design space and thereby create knowledge about the final design and, partly, to manifest design ideas in concrete artefacts [Budde et al., 1992; Lim et al., 2008]. This purpose resembles that of pilot implementation but contrary to pilot implementation prototyping does not normally involve experience with the prototype in real use. For example, Beynon-Davies et al. [1999, p. 108] note that “the use of the word prototype tends to suggest the tentative nature of this artefact. It is early, unfinished or a model of something”. In the early stages of the development phase, prototypes may support the communication between designers and users by visualizing alternative system solutions and thereby facilitating discussion of initial ideas and the clarification of requirements. In the later stages of the development phase, users may try out a prototype to evaluate the proposed functionality and identify shortcomings in the design. However, the tentative nature of prototypes entails that they can mostly be explored and tried out in an environment separated from real use, for example in a usability laboratory. There are, thus, limitations to the kinds of learning that can be achieved by experimenting with a prototype prior to real use.

In contrast, pilot implementation involves that a system is installed and used in its intended environment, though for a limited period only. The resulting feedback can provide crucial guidance during subsequent development activities because the use of the system in its intended environment allows for emergent changes to surface and become recognized as drawbacks or opportunities, for users and developers to advance their understanding of the optimal fit between system and organization, and for gauging the preparedness of the organization to assimilate the system. Collectively, these properties of pilot implementation enforce realism. This realism constitutes the primary difference between pilot implementation and prototyping. In this sense pilot implementation can be seen as a supplement to prototyping. Both techniques have an overarching learning objective,

but pilot implementation focuses on learning from real work settings with their organizational context and mundane practicalities, which cannot be recreated in a laboratory. As a result the complications involved in performing pilot implementations differ from those of prototyping. For example, pilot implementation involves organizational adaptations (see below), which are not relevant to prototyping.

The elements of pilot implementation

To describe pilot implementations in more detail we have developed a model of their constituent elements. Figure 1 shows the five interrelated activities that comprise our model of a pilot implementation: planning and design, technical configuration, organizational adaptation, use, and learning. The first four activities are the standard ISD phases of initiation, development, implementation, and operation [Alter, 2001] adapted to the pilot-implementation context; the fifth activity – learning – is the objective that motivates performing the four other activities. Some temporal progression occurs from planning and design through technical configuration and organizational adaptation to use, but these activities do not form a linear sequence. For example, technical configuration and organizational adaptation can to a large extent be performed in parallel, and the pilot system may be modified during the period of its use. Learning goes on in parallel with the four other activities.

During *planning and design*, it is defined what issues the pilot implementation addresses and how they will be studied. This includes determining where and when the pilot implementation will take place, what facilities the pilot system will include, and how lessons learned during the pilot implementation will be collected. This activity corresponds to the planning and design steps in Glass' [1997] model of pilot implementation. During *technical configuration*, the parts of the system necessary for the pilot implementation are configured to fit the pilot site, data are migrated to the system, and interfaces to the users' other systems are developed or simulations are set up. This activity involves the technical adjustments necessary to run the pilot implementation; that is, it builds on rather than subsumes the development activities of the full ISD project. During *organizational adaptation*, the pilot site revises work procedures to align with the system, trains users in the system and the revised procedures, and possibly assigns extra staff to duplicate work according to normal procedures or maintain other safeguards against breakdown. As for technical configuration, organizational adaptation is directed toward the pilot site and builds on the implementation activities of the full ISD project. While technical configuration and organizational adaptation are absent in Glass' [1997] model of pilot implementation, they resemble main steps in, for example, Wulf and Rohde's [1995] process of integrated organization and technology development. During *use*, the system is applied at the pilot site, and information is collected about the issues addressed by the pilot implementation. This involves striking a balance between making the system just another part of normal procedures and, at the same time, maintaining a special focus on the system as an object under evaluation. While some evaluation information can be collected automatically and unobtrusively, for example data about system response times, other information must be obtained from users, including information about any unanticipated consequences of introducing the system and associated organizational changes. Finally, the four above-mentioned activities spawn opportunities for *learning*. More specifically, this includes opportunities for learning about a system and its use when it is employed (a) over a period of time, (b) for a realistic span of tasks, (c) to be handled by users with realistically diverse backgrounds and workloads, (d) in collaboration with the multiple, interrelated organizational units involved in the use of the system, and (e) in a technological environment with realistic hardware, network bandwidth, and data load.

Pilot implementations can be seen as opportunities for learning from failure [Scott and Vessey, 2000] in settings devised to limit the consequences of failure. Surveys of why pilot implementations are conducted show a consistent focus on learning but no preference for learning from failure. Ward et al. [1996] find that among the surveyed UK organizations pilot implementations are mainly conducted to evaluate technologies prior to full deployment, to understand better what benefits might be attained, and to learn how benefits might actually be realized. These aims were stated by 48%, 38%, and 40%, respectively, of the organizations that conducted pilot implementations. In a survey of Australian organizations, the same aims were stated by 71%, 53%, and 52%, respectively, of the organizations conducting pilot implementations [Lin and Pervan, 2003]. Learning from pilot implementations is complicated by differences in the involved people's background and interests. Gallivan and Keil [2003] show that the communication between users and developers in situations such as pilot implementations is fragile and may miss crucial opportunities for learning. Moreover, some users may withhold their opinion about a system or even try to derail pilot implementations by means of counter-implementation strategies [Keen, 1981]. Others may be overly positive, for example because they welcome a change or do not use the pilot system enough to encounter its limitations.

IV. THREE EMPIRICAL ILLUSTRATIONS

To exemplify the complications of pilot implementations we provide three empirical illustrations. The illustrations concern pilot implementations of information systems in different parts of the healthcare sector, ranging from general practitioners over municipal healthcare centres to hospital wards. We chose pilot implementations in healthcare because electronic health records are currently being developed and implemented at considerable cost in hospitals across Europe and North America [Haux et al., 2004]; because the healthcare sector is an organizationally complex domain [LeRouge et al., 2007]; because many efforts to introduce electronic health records encounter adoption barriers [Sobol et al., 1999]; and because pilot implementations are, therefore, highly relevant in this context. For reasons of brevity each of the illustrations focuses on a subset of the five pilot-implementation elements. While all three illustrations address the planning-and-design element, the first illustration targets learning and use, the second technical configuration, and the third organizational adaptation. Table 2 provides a summary of the empirical illustrations; more information, including detailed information about the methodology used in data collection and analysis, can be found in the references provided for each of the empirical illustrations.

Electronic patient record for a stroke unit

As part of the activities involved in the project tender and bid for a strategically important contract concerning the development of an electronic patient record (EPR), the clinical-process module of the EPR was developed and pilot implemented at the stroke unit of a hospital. The pilot implementation, which we investigated by means of an action-research study [Hertzum and Simonsen, 2008; Simonsen and Hertzum, 2010], lasted five months and culminated in a five-day period of use. The EPR replaced all paper records in the stroke unit and was available on all its computers. To simulate a fully integrated EPR, a ‘back office’ was established and staffed around the clock. Record entries that involved paper transactions with other hospital wards were simulated by having the back office continuously monitor the EPR, identify such entries, mail them in the conventional fashion, wait for the results, and immediately type them into the EPR. Thus, the clinicians at the stroke unit experienced the EPR as if all transactions were fully IT supported. To learn about performance, data for 300,000 patients were migrated to the EPR before its use.

The pilot implementation was conducted to learn about planned changes, which were measured as differences between the prior use of paper records and the pilot use of the EPR. Baseline measurements of the use of paper records were performed a month before the pilot use of the EPR; similar measurements were performed during pilot use. To safeguard against misunderstandings, which might have entailed risk to patient health, the clinicians had around-the-clock access to support staff who knew the EPR well. The EPR had positive effects on team conferences, nursing handovers, and medical ward rounds. Most prominently, mental workload, measured by the NASA task load index [Hart and Staveland, 1988], tended to decrease during team conferences and medical ward rounds.

The changes that occurred during pilot use were, however, not restricted to those planned ahead. Some changes emerged spontaneously as a result of the ways in which the clinicians changed their work practices in face of the EPR. For example, the nurses engaged in a process of collective reading at their handovers, during which the EPR screen was projected on the wall and thereby visible to everybody. The electronic records were inspected by the group of nurses and they collectively participated in interpreting the status and condition of the patients, guided by the nurse team leader. The nurse team leader navigated the EPR and read selected passages aloud to draw attention to them and set a shared flow in their reading. This collective reading was a marked change in the nurses’ work practice. During nursing handovers with paper records the nurse team leader provided an oral report of each patient by scanning the patient’s record and reading key information out loud; patient records were seldom seen by clinicians other than the nurse team leader.

This change in the nurses’ work practice was unanticipated but experienced as positive. Along with the achieved planned changes, it exemplifies the learning that may result from trying out a system in real use. The pilot implementation also led to the realization of a need for a pending-tasks facility in the EPR. During the pilot implementation this need became obviously important, and it was fed back to development as a high-priority facility likely to be valuable in multiple systems. In spite of the short use period, this pilot implementation generated important insights into planned as well as emergent qualities of the EPR.

Workspace system for healthcare centres

Municipal healthcare centres were established in Denmark in 2007 with a special focus on chronic and lifestyle diseases such as diabetes, obesity, coronary heart diseases, and certain forms of cancer. As part of the establishment of the centres a healthcare centre workspace system (HCWS) was developed and scheduled for pilot implementation in three healthcare centres in 2008. The pilot implementation, which we investigated through an action-research study [Barlach and Simonsen, 2008], was planned to include a three-month period during which the HCWS was in use at the three healthcare centres. This period of pilot use was, however, postponed several times and finally cancelled.

When the three-month period of pilot use initially started, the healthcare centres reported that the system had severe performance problems. It was not until the vendor staff responsible for the technical configuration of the HCWS visited the healthcare centres that the severity of these problems became clear to them. Contrary to the usability tests that had been run at the vendor, some drop-down menus suffered delays of about 60 seconds. This necessitated renewed work on the technical configuration of the HCWS and renewed planning of the pilot implementation. It took almost three months to diagnose and solve the performance problem. During this time the healthcare centres gradually lost interest in the pilot implementation and focused increasingly on receiving the improvements of their daily work expected to result from full-scale implementation of the HCWS. The reason it took three months to solve the problem was partly that it was hard to diagnose but mainly that the vendor did not treat it as a high-priority issue.

The HCWS had a three-layer architecture consistent with the ANSI/SPARC standard [Brodie and Schmidt, 1982]: a user-interface layer, a functionality layer with the system's different functional modules (e.g., a booking module), and a data-model layer consisting of a generic clinical framework. The three layers also represented three different vendor units. The user-interface layer was used by the configurators to build the user interface for the specific customer. The functionality layer served multiple healthcare customers and was maintained by a separate developer group. The data-model layer was maintained by a third developer group and served as a generic development tool potentially for all the vendor's healthcare customers. This layered architecture and the distributed development organization made the performance problem difficult to handle because it could arise from any of the three layers and from interactions among them. Analysis of the problem and negotiations among the vendor units resulted in a decision to resolve the performance problem in the data-model layer. However, the developer group responsible for the generic clinical framework that constituted the data-model layer assigned higher priority to systems in operation than to a pilot system, and the performance problem was therefore not solved until the next regular release.

This illustrates how the vendor judged the technical configuration of the pilot system as insufficiently important to upset the release plans, which were directed at systems in operation. When the performance problem had been resolved the customer no longer wanted to give priority to the learning objective of the pilot implementation, and the HCWS was instead released for operational use. Two years later the HCWS was in the process of being redeveloped with another clinical framework as its data-model layer.

Electronic pregnancy record

In Denmark, care during pregnancy and childbirth is usually organized as a shared-care arrangement involving the woman's general practitioner (GP), a midwives' clinic, and a public hospital. To improve information sharing between the parties and, by implication, improve the continuity of care and the pregnant woman's participation in her own care, the Danish national e-health portal, *sundhed.dk*, decided to develop a national electronic Pregnancy Record (ePR) providing web access to pregnancy records wherever and whenever needed. The ePR was supposed to replace an existing paper form – the so-called Pregnancy Record (PR). The PR is commenced at the pregnant woman's first appointment with her GP, who records her personal details, history, blood pressure and so forth. The record is given to the woman so that she can bring it with her to all subsequent appointments during her pregnancy, including those with the midwife and at the hospital. At each visit, the care provider must record pertinent information concerning diagnostic and treatment decisions in the PR.

After spending more than a year developing the software, conducting technical tests, and testing it for usability on a small number of prospective users, a pilot implementation commenced in October 2005 with the purpose of assessing the usability and usefulness of the application in actual clinical work. The pilot implementation, which we investigated by means of observation and semi-structured interviews [Bansler and Havn, 2010], was planned to last 12 months and involve 10 GPs, one midwives' clinic,

the department of obstetrics at a public teaching hospital and approximately 100 women. However, in May 2006, after less than seven months, *sundhed.dk* decided to abort the pilot. The premature ending of the pilot rendered the outcome indeterminate and inconclusive. What had happened?

Despite some teething problems, the organizational adaptation in the general practices and the midwives' clinic went relatively well. Of course, the ePR had shortcomings, but with practice the users gradually learned to cope with them. At the hospital, however, the implementation failed completely, mainly because the nurses and physicians only used the application sporadically and never became familiar with it. Consequently, they found it difficult to use, and they tended to view it as a nuisance imposed on them by an outside authority. The underlying problem concerned the design of the pilot implementation and the organization of hospital work.

The system developers had deliberately tried to limit the number of pilot users and organize the pilot implementation so that each pilot user would use the system regularly and thus quickly become proficient in using it. For instance, they made sure that all the women participating in the trial were referred to the same few selected midwives (a group of 5 persons) at the midwife's clinic. However, when it came to the hospital, it proved impossible to organize the pilot implementation so that it would only involve a few physicians and nurses. The organization of hospital work with its demand for continuous around-the-clock operation, rotating shift-work schedules, and many unanticipated events made it impossible to limit participation to a small group of users. For obvious reasons, one could not plan in advance when the participating pregnant women would give birth (or need medical care), and therefore all the physicians and nurses at the obstetrics ward had to be able to access and use the ePR. In other words, all 100+ physicians and nurses had to take part in the pilot implementation.

The project manager recognized that this was a potential problem, but she realized that the only way of solving it would be to increase the number of pregnant women enrolled in the pilot implementation and thereby increase the chances of coming across a woman with an ePR. Given that the hospital manages more than 3500 births per year, the number of women participating in the trial would have had to be increased dramatically (maybe to 1000+ women) to make a real difference. This, in turn, would have required the involvement of far more midwives and GPs (at least 100 more). It would have been exceedingly costly and the project would have begun to look more like a full-scale implementation than a pilot implementation. So, she decided to go ahead as planned and hope the best.

While the ePR aimed to support the multiple organizational interdependencies in care during pregnancy, these same interdependencies made it hard to set an appropriate boundary for the pilot implementation. The failure of the organizational adaptation at the hospital diminished the value of the ePR also to the participating midwives and GPs, because the records were incomplete. As a consequence, the usability and usefulness of the ePR could not be evaluated.

Taken together, the three empirical illustrations show some of the complexities involved in conducting a pilot implementation as part of an ISD project, complexities that stem from the fact that a pilot system – unlike a prototype – must be installed in a real setting and used by real users as part of their everyday work. This implies that the pilot implementation becomes subjected to all the vagaries and complications of organizational life as well as the imperfections and peculiarities of real-life technical systems and infrastructures. Only one of the pilot implementations was successful, the two others failed in the sense that very little could be learned from them, because the pilot system never became integrated into the daily work routine and users never became familiar with its workings. While we consider two of the *pilot implementations* as failed, it should be noted that we are not assessing whether the overarching *ISD projects* succeeded or failed. The main cause for the failure of the pilot implementation in the healthcare-centre case was rooted in difficulties with the technical configuration of the pilot system, resulting in severe performance problems. In the pregnancy case, the main cause was related to the organizational adaptation, and more specifically to the around-the-clock organization of hospital work and the associated problems of “sporadic use.” In contrast, meticulous planning and thorough technical and organizational preparations before the actual test period (it took five months to prepare for five days of use) as well as extensive user support and data collection during the test itself characterized the successful stroke case. This approach was time consuming and required a great deal of coordination, follow up, and hard work, but at the same time emphasizes the need for careful attention to all five elements of the pilot-implementation model.

V. DISCUSSION

Pilot implementation involves using a system for real work before the system is finalized and, thereby, provides for learning about how the system may eventually support its users in their work. Learning that is based on such real-use experiences goes beyond what users and developers can learn from prototyping in a laboratory. This makes pilot implementation a relevant ISD technique whenever systems are developed for implementation in complex organizational settings, projects aim to be highly innovative, the cost of system failure is high, or the project might be severely hampered by other aspects of inadequate real-use feedback. In these situations the possible benefits of a pilot implementation appear greater than the challenges that must be faced in conducting pilot implementations. Because these challenges may cause a pilot implementation to fail they are important to researchers as well as practitioners:

First, the learning objective may be contested or considered secondary to other objectives. Pilot implementations should be conceptualized as field tests and consequently be designed to maximize learning [Lancaster et al., 2004; Lauesen, 2002; van Teijlingen and Hundley, 2005]. It may, however, be difficult to maintain clarity about the purpose and importance of a pilot implementation in the midst of all the surrounding activities. Because pilot implementations involve that the pilot system is used for real work, the learning objective is added to the objectives that come with performing the real work. In healthcare settings, the top priority is quality treatment of the patients. In other domains, pilot implementations must similarly be expected to be a subordinate concern. The stroke case illustrates how special precautions, such as around-the-clock presence of support staff, may allow for learning without adverse consequences for the real work. Because the users are typically not observed and supervised throughout a pilot implementation, as they are during prototype tests in a laboratory, it becomes crucially important that the users are themselves committed to the pilot implementation and its learning objective. It is, therefore, an essential task for those who conduct pilot implementations to motivate users and maintain their commitment [Hertzum, 2006]. The learning objective of pilot implementations may, however, also be contested within the vendor organization. The healthcare-centre case, for example, illustrates how the learning objective was considered secondary to the scheduled maintenance of the systems already in operation. The vendor did not prioritize that a successful pilot implementation required shorter response times for technical configuration than the maintenance of operational systems. As a consequence, user commitment had degraded when technical configuration had finally been completed. Our experiences show that unless the learning objective is carefully managed throughout a pilot implementation, it is likely to suffer.

Second, it is nontrivial to define an appropriate scope for a pilot implementation. A pilot implementation is, by definition, conducted in a scaled-down fashion so only some users and organizational entities will be involved. For resource reasons, the scope should be as limited as possible, but at the same time one must ensure that the pilot implementation fulfils its purpose and that the involved users constitute a representative sample from the population [Liang et al., 2006; Pal et al., 2008]. Consequently, a pilot implementation aiming to test the effects of computer use on doctor-patient relations during consultations can be more narrowly scoped than a pilot implementation of a system that encompasses a wider range of users, activities, and organizational entities. In the pregnancy case the failure of the pilot implementation at the hospital can be traced back to an inability to define an appropriate scope. The organization and nature of hospital work made it impossible to limit participation to a small group of clinicians. The alternative of enrolling more pregnant women to ensure that all clinicians at the hospital had sufficient opportunity to work with the system was exceedingly costly because it required involving many more midwives and GPs. The use of a back office in the stroke case to simulate a fully integrated system by means of behind-the-scene manual work illustrates one way of defining a limited scope for a pilot implementation while at the same time handling the various interactions across the boundary between the pilot implementation and the rest of the hospital. The extent of such organizational adaptations also suggests the amount of planning and resources that may be involved in scoping a pilot implementation.

Third, a pilot implementation is, by definition, temporary and it is important to decide how long it should last. On the one hand, it is desirable to keep the pilot implementation as short as possible because the pilot implementation is itself costly and because “full-scale implementation awaits completion of [the] pilot” [Pal et al., 2008, p. 261] and finalization of the system design. Thus, extra time allocated to the pilot will delay the overall ISD project and result in lost-opportunity costs. On the other hand, organizations normally experience a dip in performance immediately after the introduction of a system because start-up problems have to be overcome, users need to become proficient in using the system, and new work practices have to stabilize. Also, users may feel frustrated and may therefore

be reluctant to adopt the system [Applegate et al., 2009, pp. 313-314]. The pilot implementation should be sufficiently long to overcome this critical period and reach a level of operation that allows for realistic assessment of the system. The five-day period of pilot use in the stroke case was critically brief and dictated by concerns external to the pilot implementation. This left only a very concentrated period for learning to occur. It must also be recognized that the brevity of the pilot implementation contributed substantially to making it feasible to have support staff and a back office available around-the-clock. In the pregnancy case the period of pilot use was longer but much more troubled, and in the healthcare-centre case the period of pilot use never overcame the initial critical period. More work is required to be able to make recommendations about the duration of pilot implementations.

Fourth, pilot implementations are not simply small-scale versions of full implementations. It is sometimes assumed that pilot implementations are less complicated and less risky than normal, full-scale implementations [Pal et al., 2008, p. 264], but the challenges above show that pilot implementations have their own complications. A further complication is that a pilot system is not fully developed; rather, it is being pilot implemented to get feedback for its finalization. Users are therefore likely to experience some malfunctions and breakdowns [Bossen, 2007; Peute and Jaspers, 2007], leading also to increased demands on project management and support staff. The root cause of all these complications is that the purpose of a pilot implementation is not, per se, to implement a system in an organization, thus making a pilot implementation fundamentally different from a full implementation [Glass, 1997]. Instead, a pilot implementation involves that one must plan and design the means by which data and user feedback will be collected, technically configure the system to handle that it is used by only some organizational entities, organizationally adapt these entities to minimize the effects of extraneous factors, and use the system in order to learn about how it matches and affects the organization. This requires, especially during the initial part of the use period, a readiness to respond quickly to emergent needs for user support, system fixes, and organizational adjustments. A pilot implementation constitutes a step between prototyping and actual implementation. Lichter et al. [1994] propose that the transition from prototype to fully implemented system can proceed gradually: “After reaching a certain degree of ‘sophistication,’ the prototype is implemented as a pilot system and enhanced in cycles” and “there ceases to be any strict distinction between the prototype and the application system” (p. 826). In contrast, we argue that pilot implementations have distinct characteristics and challenges that differentiate them from prototyping as well as from normal, full-scale implementation.

VI. CONCLUSION

Pilot implementation is an 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. We have defined a pilot implementation as 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 ongoing development activities, pilot implementation supplements prototyping, which in most definitions is restricted to the development phase.

As we have pointed out pilot implementations have attracted little research interest and little is, therefore, known about how to conduct and use pilot implementations as vehicles for learning in ISD projects. We suggest that our model of the constituent elements of pilot implementation provides a useful schema for guiding future empirical research on pilot implementation. We will conclude by briefly indicating some urgent research questions:

Learning is essential for pilot implementations to succeed in providing useful real-use feedback but is under constant pressure from the need to get the normal work done and a concomitant reluctance to upset production schedules for the sake of a pilot implementation. How is a learning environment created, in which users are motivated to participate, errors are seen as opportunities for organizational learning rather than as grounds for blame and punishment, and data about the effects of using the pilot system are systematically collected and fed back to development?

The *planning and design* of pilot implementations involves defining their boundaries to limit the pilot implementation while at the same time accommodating that the users’ work extends beyond these boundaries, for example through collaboration with organizational entities not involved in the pilot

implementation and through tasks initiated prior to the pilot implementation. How can an appropriate scope and timeframe be defined for a pilot implementation?

Technical configuration involves configuring the system for the pilot-implementation site, migrating data to the system, and setting up interfaces to other systems at a time where the system is still unfinished and offers only a subset of its full capabilities. How can this be accomplished through the use of a combination of flexible development tools and simulation techniques?

Organizational adaptation involves the establishment of ad hoc organizational arrangements and procedures to integrate the pilot system into the work practices of the organization, while the system is at the same time being configured. How can the interplay between customer and vendor be organized to achieve alignment between the organizational adaptations and the technical configuration?

The *use* of a pilot system involves extra work and uncertainty for the users, in addition to their normal work, but they must still be able to concentrate on their normal work and perform without increased risk of unacceptable errors. How can compensation, support, and special precautions balance the demands of the users' normal work against the novelty and unfinishedness of the pilot system?

The questions above call for considerable further research on pilot implementation, including the development of guidelines and methods for practitioners who wish to conduct pilot implementations to supplement the use of prototypes. A specific implication for practice lies in not mistaking pilot implementation for normal implementation. Pilot implementations come with their own challenges and are likely to fail unless these challenges are addressed.

ACKNOWLEDGEMENTS

This work was supported, in part, by grants from the Danish Agency for Science, Technology, and Innovation (grant no. 07-024801) and the Danish Council for Strategic Research (grant no. 2106-07-0017). The article is the result of a collaborative effort spanning a period of 18 months. The authors contributed equally to the discussions reflected in the article. All authors contributed to the writing, but the first author wrote the larger part of the article text. Multiple versions of the article have been read and discussed critically by all authors. We gratefully acknowledge Anders Barlach's leading role in conducting the healthcare-centre case.

REFERENCES

- Abrahamsson, P., J. Warsta, M. T. Siponen, and J. Ronkainen (2003) "New directions on agile methods: A comparative analysis" in *Proceedings of the International Conference on Software Engineering*, Los Alamitos, CA: IEEE Press, pp. 244-254.
- Alter, S. (2001) "Which life cycle - Work system, information system, or software?", *Communications of the Association for Information Systems*, (7), pp. 17:01-17:54.
- Applegate, L. M., R. D. Austin, and D. L. Soule (2009) *Corporate information strategy and management*, New York: McGraw-Hill.
- Babar, M. A., B. Kitchenham, L. Zhu, I. Gorton, and R. Jeffery (2006) "An empirical study of groupware support for distributed software architecture evaluation process", *Journal of Systems and Software*, (79)7, pp. 912-925.
- Bansler, J. P., and E. Havn (2010) "Pilot implementation of health information systems: Issues and challenges", *International Journal of Medical Informatics*, (79)9, pp. 637-648.
- Barlach, A., and J. Simonsen (2008) "Effect specifications as an alternative to use cases" in A. Asproth, K. Axelsson, S. C. Holmberg, C. Ihlström and B. Sundgren (eds.), *IRIS31: Proceedings of 31st Information Systems Research Seminar in Scandinavia*, Åre, SE.
- Basili, V. R., and A. J. Turner (1975) "Iterative enhancement: A practical technique for software development", *IEEE Transactions on Software Engineering*, (1)4, pp. 390-396.
- Beynon-Davies, P., D. Tudhope, and H. Mackay (1999) "Information systems prototyping in practice", *Journal of Information Technology*, (14)1, pp. 107-120.
- Boehm, B. (1981) *Software engineering economics*, Englewood Cliffs, NJ: Prentice Hall.

- Boehm, B. (1988) "A spiral model of software development and enhancement", *IEEE Computer*, (21)5, pp. 61-72.
- Boehm, B. (2000) "Requirements that handle IKIWISI, COTS, and rapid change", *IEEE Computer*, (33)7, pp. 99-102.
- Boehm, B. (2002) "Get ready for agile methods, with care", *IEEE Computer*, (35)1, pp. 64-69.
- Boehm, B., and R. Turner (2004) *Balancing agility and discipline: A guide for the perplexed*, Boston, MA: Pearson.
- Bossen, C. (2007) "Test the artefact - develop the organization. The implementation of an electronic medication plan", *International Journal of Medical Informatics*, (76)1, pp. 13-21.
- Brodie, M. L., and J. W. Schmidt (1982) "Final report of the ANSI/X3/SPARC DBS-SG relational database task group", *ACM SIGMOD Record*, (12)4, pp. 1-62.
- Brooks, F. P. (1987) "No silver bullet: Essence and accidents of software engineering", *IEEE Computer*, (20)4, pp. 10-19.
- Budde, R., K. Kautz, K. Kuhlenkamp, and H. Zullighoven (1992) *Prototyping: An approach to evolutionary system development*, Berlin: Springer.
- Cockburn, A. (2007) *Agile software development: The cooperative game* (2nd ed.), Upper Saddle River, NJ: Addison-Wesley.
- Davis, A. M. (1995) "Software prototyping" in M. C. Yovits and M. R. Zelkowitz (eds.), *Advances in Computers* (Vol. 40), Boston, MA: Academic Press, pp. 40-63.
- Floyd, C. (1984) "A systematic look at prototyping" in R. Budde, K. Kuhlenkamp, L. Mathiassen and L. Zullighoven (eds.), *Approaches to Prototyping: Proceedings on the Working Conference on Prototyping*, Heidelberg: Springer, pp. 1-18.
- Gallivan, M. J., and M. Keil (2003) "The user-developer communication process: A critical case study", *Information Systems Journal*, (13)1, pp. 37-68.
- Glass, R. L. (1997) "Pilot studies: What, why, and how", *Journal of Systems and Software*, (36)1, pp. 85-97.
- Hart, S. G., and L. E. Staveland (1988) "Development of NASA-TLX (task load index): Results of empirical and theoretical research" in P. A. Hancock and N. Meshkati (eds.), *Human Mental Workload*, Amsterdam: North-Holland, pp. 139-183.
- Haux, R., A. Winter, E. Ammenwerth, and B. Brigl (2004) *Strategic information management in hospitals: An introduction to hospital information systems*, Heidelberg: Springer.
- Hertzum, M. (2006) "Problem prioritization in usability evaluation: From severity assessments toward impact on design", *International Journal of Human-Computer Interaction*, (21)2, pp. 125-146.
- Hertzum, M., and J. Simonsen (2008) "Positive effects of electronic patient records on three clinical activities", *International Journal of Medical Informatics*, (77)12, pp. 809-817.
- Iredale, R., J. Gray, and G. Murtagh (2002) "Telegenetics: A pilot study of videomediated genetic consultations in Wales", *Journal of Medical Marketing*, (2)2, pp. 130-135.
- Janson, M. (1986) "Applying a pilot system and prototyping approach to systems development and implementation", *Information & Management*, (10)4, pp. 209-216.
- Janson, M., and J. P. Hammerschmidt (1990) "Managing the information systems development process: The case for prototype and pilot systems", *Information and Management Sciences*, (1)1, pp. 45-62.
- Jurison, J. (1996) "The temporal nature of IS benefits: A longitudinal study", *Information & Management*, (30)2, pp. 75-79.
- Keen, P. G. W. (1981) "Information systems and organizational change", *Communications of the ACM*, (24)1, pp. 24-33.
- Lancaster, G. A., S. Dodd, and P. R. Williamson (2004) "Design and analysis of pilot studies: Recommendations for good practice", *Journal of Evaluation in Clinical Practice*, (10)2, pp. 307-312.

- Larman, C., and V. R. Basili (2003) "Iterative and incremental development: A brief history", *IEEE Computer*, (36)6, pp. 47-56.
- Lauesen, S. (2002) *Software requirements*, London: Addison-Wesley.
- LeRouge, C., V. Mantzana, and E. V. Wilson (2007) "Special section on healthcare information systems research, revelations and visions", *European Journal of Information Systems*, (16)6, pp. 669-760.
- Liang, H., Y. Xue, and B. A. Berger (2006) "Web-based intervention support system for health promotion", *Decision Support Systems*, (42)1, pp. 435-449.
- Lichter, H., M. Schneider-Hufschmidt, and H. Züllighoven (1994) "Prototyping in industrial software projects - Bridging the gap between theory and practice", *IEEE Transactions on Software Engineering*, (20)11, pp. 825-832.
- Lim, Y.-K., E. Stolterman, and J. Tenenbergh (2008) "The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas", *ACM Transactions on Computer-Human Interaction*, (15)2, pp. 7:01-07:27.
- Lin, C., and G. Pervan (2003) "The practice of IS/IT benefits management in large Australian organizations", *Information & Management*, (41)1, pp. 13-24.
- Markus, M. L. (2004) "Technochange management: Using IT to drive organizational change", *Journal of Information Technology*, (19)1, pp. 4-20.
- McCracken, D. D., and M. A. Jackson (1982) "Life-cycle concept considered harmful", *ACM Software Engineering Notes*, (7)2, pp. 29-32.
- Orlikowski, W. J. (1996) "Improvising organizational transformation over time: A situated change perspective", *Information Systems Research*, (7)1, pp. 63-92.
- Pal, R., A. Sengupta, and I. Bose (2008) "Role of pilot study in assessing viability of new technology projects: The case of RFID in parking operations", *Communications of the Association for Information Systems*, (23)Article 15, pp. 257-276.
- Peute, L. W. P., and M. W. M. Jaspers (2007) "The significance of a usability evaluation of an emerging laboratory order entry system", *International Journal of Medical Informatics*, (76)2&3, pp. 157-168.
- Reed, G., V. Story, and J. Saker (2004) "Information technology: Changing the face of automotive retailing?", *International Journal of Retail & Distribution Management*, (32)1, pp. 19-32.
- Rzevski, G. (1984) "Prototypes versus pilot systems: Strategies for evolutionary information system development" in R. Budde, K. Kuhlenkamp, L. Mathiassen and L. Züllighoven (eds.), *Approaches to Prototyping: Proceedings on the Working Conference on Prototyping*, Heidelberg: Springer, pp. 356-367.
- Schwaber, K., and M. Beedle (2002) *Agile software development with Scrum*, Upper Saddle River, NJ: Prentice-Hall.
- Scott, J. E., and I. Vessey (2000) "Implementing enterprise resource planning systems: The role of learning from failure", *Information Systems Frontiers*, (2)2, pp. 213-232.
- Simonsen, J., and M. Hertzum (2010) "Iterative participatory design" in J. Simonsen, J. O. Bærenholdt, M. Büscher and J. D. Scheuer (eds.), *Design Research: Synergies from Interdisciplinary Perspectives*, London: Routledge, pp. 16-32.
- Sobol, M. G., M. Alverson, and D. Lei (1999) "Barriers to the adoption of computerized technology in health care systems", *Topics in Health Information Management*, (19)4, pp. 1-19.
- Stapleton, J. (1997) *Dynamic systems development method - The method in practice*, Boston, MA: Addison-Wesley Longman.
- Turner, J. R. (2005) "The role of pilot studies in reducing risk on projects and programmes", *International Journal of Project Management*, (23)1, pp. 1-6.
- van Teijlingen, E., and V. Hundley (2005) "Pilot studies in family planning and reproductive health care", *Journal of Family Planning and Reproductive Health Care*, (31)3, pp. 219-221.

- Ward, J., and E. Daniel (2006) *Benefits management: Delivering value from IS & IT investments*, Chichester, UK: Wiley.
- Ward, J., P. Taylor, and P. Bond (1996) "Evaluation and realisation of IS/IT benefits: An empirical study of current practice", *European Journal of Information Systems*, (4)4, pp. 214-225.
- Williams, L., and A. Cockburn (2003) "Agile software development: It's about feedback and change", *IEEE Computer*, (36)6, pp. 39-43.
- Wulf, V., and M. Rohde (1995) "Towards an integrated organization and technology development" in *Proceedings of the DIS '95 Symposium on Designing Interactive Systems*, New York: ACM Press, pp. 55-64.

Table 1: Contrasting prototyping and pilot implementation

	<i>Prototyping</i>	<i>Pilot implementation</i>
<i>Purpose (“why?”)</i>	To learn about the final design by traversing a design space, manifesting design ideas in concrete artefacts, and testing the fit between a proposed design and the user	To learn about the fit between the system and its context in order to explore the value of the system, improve or assess its design, and reduce implementation risk
<i>System (“what?”)</i>	Prototype, i.e. a model, early design or not yet properly engineered system	Pilot system, i.e. a properly engineered, yet unfinished system
<i>Setting (“where?”)</i>	In the laboratory, i.e. separated from real work	In the field, i.e. during real work but limited to one or a few sites
<i>Process (“how?”)</i>	Demonstrated or tried out in brief sessions simulating real use, with test data and test tasks	Used in its intended environment for a limited period of time, with real data and special precautions against breakdowns
<i>Time (“when?”)</i>	During development when it is feasible to test the system design	During development when it is feasible to test the design <i>and</i> its implementation
<i>Duration (“how long?”)</i>	Short, i.e. typically days or weeks	Long, i.e. typically weeks or months

Table 2: Summary of the three empirical illustrations

	<i>Electronic patient record for a stroke unit</i>	<i>Workspace system for healthcare centres</i>	<i>Electronic pregnancy record</i>
<i>Planning and design</i>	<p>Purpose and approach – to learn about planned changes using mainly quantitative methods and emergent changes using qualitative methods.</p> <p>Scope and duration – involved all staff and all patients at one stroke unit for a period of five days of around-the-clock use.</p>	<p>Purpose and approach – to evaluate the clinicians’ perception of using the HCWS and the patients’ perception of the received treatment, by means of questionnaires.</p> <p>Scope and duration – planned to involve all staff and selected patients at three healthcare centres for a period of three months.</p>	<p>Purpose and approach – to evaluate usability and usefulness using qualitative methods, and to solicit ideas for improvements to the design.</p> <p>Scope and duration – planned to last 12 months and involve 10 GPs, one midwives’ clinic, an obstetrics department, and 100 pregnant women.</p>
<i>Technical configuration</i>	<p>Main challenges – to design overview displays for selected clinical activities, to interface the EPR to other clinical systems, and to provide access to old patient data.</p> <p>Adopted solutions – overview displays were designed through a series of workshops with clinicians and designers, and considerable resources were spent developing interfaces to and migrating data from existing systems.</p>	<p>Main challenges – to agree on the level in the HCWS at which to address the experienced performance problems and, thereby, to determine who should solve these problems.</p> <p>Experienced problems – system response times were prohibitively long and the pilot implementation had to be postponed until these problems were solved.</p>	<p>Main challenges – to install digital signature software on all computers and ensure correct setup of firewalls, browsers, etc.</p> <p>Experienced problems – early on technical breakdowns were quite common, and the project was unable to react quickly and effectively to solve them.</p>
<i>Organizational adaptation</i>	<p>Main challenges – to simulate that the EPR was in use across the entire hospital, to safeguard against errors, and to prepare revised work procedures.</p> <p>Adopted solutions – a back office simulated use of the EPR also outside of the stroke unit, support staff was available around the clock, and the staff received training in revised work procedures.</p>	<p>Main challenges – to negotiate an understanding of the adaptations necessitated by the pilot implementation and its learning objective, as opposed to a normal implementation.</p> <p>Experienced problems – to keep the customer committed to the learning objective when the pilot implementation was postponed.</p>	<p>Main challenges – to train and motivate users effectively, to distribute memory cards with private keys to all pilot users at the midwives’ clinic and the obstetrics department.</p> <p>Experienced problems – the project failed to motivate staff at the obstetrics department to participate actively in the pilot, and the around-the-clock organization of hospital work led to “sporadic use”.</p>
<i>Use</i>	<p>Outcome – the EPR was in pilot use for five intensive days during which it replaced all paper records for all patients at the stroke unit.</p>	<p>Outcome – the period of pilot use had to be postponed and was finally cancelled in the sense that the HCWS was instead released for operational use.</p>	<p>Outcome – the pilot was stopped before time, mainly because the implementation failed at the obstetrics department due to organizational factors and lack of commitment from pilot users and management.</p>
<i>Learning</i>	<p>Outcome – it was documented that the EPR and associated work procedures led to several of the planned changes, additional positive changes unexpectedly emerged, and a long list of small-scale change requests was accumulated.</p>	<p>Outcome – little was learned about the clinicians’ perception of using the HCWS and the patients’ perception of the received treatment, because neither clinicians nor patients experienced the effects of the HCWS before the pilot implementation was aborted.</p>	<p>Outcome – little could be learned about the usability and usefulness of the proposed system design or how well it would fit with the organization, because pilot users never became sufficiently familiar with the pilot system to use it routinely as part of their daily work.</p>

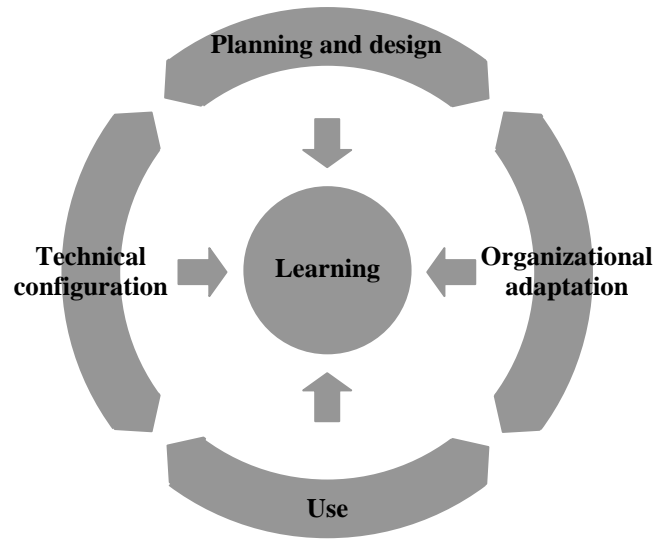


Figure 1: The five elements of a pilot implementation.

Author biographies:

Morten Hertzum, Ph.D., is associate professor of Computer Science at Roskilde University. His research includes user-centred design, achieving benefit from IT, and evaluation. Currently, his empirical work concerns IT in healthcare. He has published in, for example, *Information Processing & Management*, *International Journal of Human-Computer Interaction*, and *Information & Organization*.

Jørgen P. Bansler is a professor of Computer Science at University of Copenhagen. His research interests include information systems development, organizational implementation, CSCW and medical informatics. He has published in such journals as *ACM Transactions on Information Systems*, *Journal of the AIS*, *Information Technology & People*, and *Computer Supported Cooperative Work*.

Erling C. Havn is associate professor at Technical University of Denmark. His research interest includes organization theory, CSCW, medical informatics, organizational implementation of IS. He has published in such journals as *Journal of the AIS*, *Information Technology & People*, *Computer Supported Cooperative Work*, *International Journal in Human Factors in Manufacturing*, *AI&Society*.

Jesper Simonsen is Professor in Design Studies at Roskilde University. His research includes Participatory Design, in specific on offering theories and methods for IT design in an organizational context. Has published such journals as *Communications of the ACM*, *Scandinavian Journal of Information Systems*, *Human-Computer Interaction*, and *Computer Supported Cooperative Work*.