12-31-2009

ERP Implementation as Design: Looking at participatory design for means to facilitate knowledge integration

Lene Pries-Heje
Copenhagen Business School, lpries@itu.dk

Yvonne Dittrich
The IT university of Copenhagen, ydi@itu.dk

Follow this and additional works at: http://aisel.aisnet.org/sjis

ERP implementation as design

Looking at participatory design for means to facilitate knowledge integration

Lene Pries-Heje
Copenhagen Business School, Denmark
lph.inf@cbs.dk

Yvonne Dittrich
IT University of Copenhagen, Denmark
ydi@itu.dk

Abstract. ERP system implementations are often problematic leading to numerous problems during and after the introduction. Although user involvement is regarded as an important success factor, the implementation difficulties indicate a lack of understanding of the process as well as methods and tools to mediate it. We argue that ERP implementations have to be understood as design processes requiring knowledge integration across different domains, and that appropriate activities should be performed to facilitate the knowledge integration. We use an in-depth case study of an ERP implementation to support our claim; applying a design perspective shows how knowledge integration problems lead to difficulties in the implementation process. Based on these results we discuss different remedies from the participatory design discourse.

Key words: ERP implementation, participatory design, case study.

1 Introduction

In the last twenty years, organisations throughout the world have implemented configurable software products that allow integration of major business processes across the organization, and provide real-time data sharing. This kind of software products is often referred to as Enterprise

Resource Planning (ERP) systems, although the software may have a wider or different scope than enterprise resource planning.

Within ERP research, user participation and involvement are found to be success factors (Nah et al. 2003; Robey et al. 2002; Summer 2003). Research on how to support user participation in the context of ERP implementations is however limited. The widespread notion of misfits and user resistance associated with ERP implementations indicates a lack of understanding of the process, methods, and tools to mediate ERP implementations.

In this article, we argue that the implementation of ERP systems must be understood and mediated as a socio-technical design process. By design, we mean "a specific type of insight building process that is geared to produce feasible and desirable results within a particular domain" (Floyd 1992, p. 93). The design has to fulfil constraints given by the technical base for implementation, the context of use and the resources available.

We only consider design if there is more than one way to implement the desired result. In the context of implementing ERP package software in a specific organization, some might question whether IT design is taking place. In our view, the ERP package as sold by the vendor is only half a product; although implementing a huge body of generic functionality, a substantial part of the design is deferred to the implementation process. The configuration and the supply of master data finalise the design of the standard software in areas where the need for adaptation is anticipated. When the built-in flexibility is not sufficient, customizations are required: changes of the program and add-ons that modify the base functionality. Apart from this technical design, the organisational implementation of an ERP system requires (re-)definition and change of a company's business processes. Thus, the participation of the future users is important to develop a working socio-technical design. Understanding an ERP implementation as two interlinked processes—a technical design and an organisational change process—allows us to draw on concepts from participatory design (PD). PD can be seen as providing a set of approaches, methods and tools mediating user participation in a socio-technical design process. One problem that many of these methods and tools aim to address is the mutual learning between IT and domain experts.

In this article, we use a categorisation of knowledge integration challenges from the PD discourse (Kensing and Munk-Madsen 1993) to analyse the data from a case study of a five year ERP implementation in an engineering company (pseudonym Alfa). Problems arose when developing and integrating knowledge regarding previous work practices, the technical possibilities provided by the ERP system, and the future system. Inadequate functionality and a problematic adoption process were the result.

The case study indicates that activities performed, and techniques and tools used to support the activities lack the ability to support the knowledge generation and integration necessary for the socio-technical design. In the discussion, we make use of the Participatory Design literature to look for methods and tools to mediate the design process in a different way. We support our proposed solutions by going back to the field material of our case.

Thus, the argumentation outlined above addresses the following research questions: Does the understanding of ERP implementation as socio-technical design provide a frame for a relevant analysis of the occurred difficulties? And does it help to identify new directions to explore possible remedies?

28 • Pries-Heje & Dittrich

The remainder of the paper is organized as follows. The next section reviews ERP research literature on user participation and knowledge integration. As ERP implementations are not understood as design processes, only the provision of domain knowledge and technical knowledge is considered. In other words, the need to develop an adequate understanding of how to design the IT artefact and how the ERP implementation changes the existing processes in order to estimate the feasibility of the changes, is not considered. In section 3, PD is introduced. Cooperation between domain experts and IT professionals requires the integration of different professional knowledge domains and knowledge about the current design of the future system. In section 4, the research method is explained and a framework for analyzing the case is developed. We use an analytical process model in order to relate changes in the knowledge integration capabilities and the dynamics in participation. Section 5 contains an analysis of the case. In section 6, the findings of the empirical data analysis are discussed; understanding ERP system implementation as design facilitates a more relevant understanding of the problems in implementation and adaptation of ERP systems. Drawing on the PD toolbox, we propose different remedies for the difficulties indicated by the field material. Section 7 summarizes implications for practice, outlines future research, and draws conclusions.

2   User participation and knowledge integration in ERP implementations

Having users participate in ERP implementations is considered essential for success (Kawalek and Wood-Harper 2002; Nah et al. 2003; Robey et al. 2002) and is expected to provide a better fit of user requirements, achieving better system quality, improved use, and acceptance (Esteves-Sousa and Pastor-Collado 2000). The project team should be balanced or cross-functional, and should comprise a mix of external consultants and internal staff (Holland et al. 1999; Shanks et al. 2000; Summer 1999). Both business and technical knowledge are important (Haines and Goodhue 2003, Shanks et al. 2000; Summer 1999). Sharing information among the various parties involved is vital and requires partnership trust (Stefanou 1999), and the team should be empowered to make quick decisions (Shanks et al. 2000). Understanding the overall design of the ERP system and the new features of the ERP system is critical (Kim et al. 2005). Thus user participation and knowledge issues are widely recognized as important success factors, but how do they influence the implementation process?

The role of human knowledge and skills involved in the ERP life cycle is an underlying theme in Markus and Tanis' (2000) stage model of ERP implementations. Without discussing specific methods, Markus and Tanis emphasise the challenge that the configuration of the standard package to the specific use situation requires mapping the organizational requirements to the systems' business processes and the terminology used by the vendor.

User participation and knowledge issues are also central to Robey et al.'s (2002) multi case interview study, including both successful and less successful implementations of ERP packages. Robey et al. found that all participating users had difficulties obtaining sufficient knowledge to configure the system and assimilate the new business processes and management structures.

ERP implementation as design • 29

They argue that the presence of domain knowledge and successful communication between IT experts and users are necessary to overcome "the configuration knowledge barrier" and, in turn, participation and social bounding are essential for successful communication. In their study the more successful companies had a large core team with diverse expertise, the team members were rewarded to stay on the project until the end, and the team were staffed with respected business and technology managers. Robey et al.'s (2002) study emphasised the knowledge transfer from IT experts to user representatives for the configuration, and that in order to overcome the so-called "assimilation knowledge barriers" an incremental approach as well as formal training should be used. Although Robey et al. claim that successful communication is essential, their study provides no understanding of how the user representatives and consultants actually develop means for communication or how knowledge is generated and integrated over time.

In a study by Pan et al. (2001), knowledge integration in ERP implementations is identified as a key problem. They found that knowledge is embedded in complex organizational processes, in legacy systems, in externally based processes, and in the ERP system. Understanding and sharing this embedded knowledge is important in order to integrate knowledge. Bringing key participants together and solving conflicts between the different parties involved is difficult but necessary. Based on their case study, they argue that relationship building is critical both regarding interpersonal relations (one-on-one) and community relations (group-based).

Huang and Newell (2003) studied knowledge integration processes within cross-functional projects including ERP implementations. They found that knowledge integration is essentially about engaging participants through the promotion of project benefits and management of social networks. Their research had a focus on the organizational members and the development of a shared understanding within the organization; thus, the nature and the design of the IT artefact was not considered.

The same is to be said about a recent case study focussing on cross site learning and reflective capabilities in connection with a multi-site implementation by Fenema et al. (2007). Learning is connected to adapting a new standardised way of working, not related to understanding and using the design space an ERP package is providing.

In another case study Huang et al. (2001) identified the main processes involved in cross-functional knowledge integration as: (1) the penetration of different boundaries to obtain required knowledge and support; (2) the expansion of different paradigms to achieve shared understanding; and (3) the reconfiguration of organisational memory to create new organizational routines and knowledge. In their study, the knowledge integration between organizational team members and technology experts is addressed. The case indicates that the way in which the IT system was developed and modified caused difficulties in building paradigmatic overlap between the technology experts and the users involved. Although the modifications were deliberated by users and technological experts together, it was often difficult for both parts to explain why these modifications had been made. Externalizing knowledge that had been collectively constructed was difficult because of the limited overlaps of background knowledge. Most of the modifications had not been documented. Thus, the retrieval of related information and the change management became very problematic. The article stays on an abstract level; it provides no insight into how the modifications were designed and decided upon, nor does it specify which tools and techniques the team used to support this process.

In summary, we find that the ERP literature reviewed has a focus on planned change and change management, implicitly anticipating that the factual properties of the IT artifact play a minor role in the implementation. Although user participation is widely recognized as a critical success factor, the literature provides very limited insight into how user participation can be organized and supported within the context of ERP implementations. Thus, the problems of ERP implementation have not been addressed as problems of multi-disciplinary design processes but as problems of involvement, change management and commitment. As we will show, understanding ERP implementations as design processes provides new insights into the knowledge generation and integration involved, thus helping to identify specific remedies.

3   User participation in participatory design

The facilitation of cooperation between domain experts and IT professionals when designing computerised support for specific work places is one of the main research themes in Participatory Design. PD is concerned "with a more human, creative and effective relationship between those involved in technology's design and its use, and in that way between technology and the human activities that provide technological systems with their reason of being" (Suchman 1993, page viii). Kensing and Blomberg (1998) describe PD as being mainly concerned with three issues:

1. The politics of design, addressing the interaction between technological development and power relationships in organization and society. The empowerment of users as actors in these relationships in order to co-determine the technological development is emphasised.

2. The nature of participation, addressing the conditions that different design constituencies pose on the cooperative process, and how cooperation between users and developers can be mediated in the different contexts.

3. Methods, tools and techniques, facilitating the cooperative design.

One of the central challenges of PD is the mediation of design cooperation between different professional practices. To develop a usable and useful product, expertise about the application domain, that is, about the work practices of the use context and technical expertise, has to be brought to bear on each other. Design artefacts like mock-ups and prototypes have been developed serving as boundary objects mediating cooperation across heterogeneous communities of practice. Both users and developers have to be able to contribute to the evolving software application, anticipate the implication of specific design decisions on the technical implementation and the changing work practice, and evaluate it with respect to these implications. See Bødker et al. (2004) for an overview of different methods and tools that can be used in a flexible manner.

Traditionally, PD is concerned with design of IT systems from scratch or in a situation where the technical base is considered unproblematic. (See also Kensing's (2000) treatment of the implementation base.) ERP systems on the other side provide a massive base of pre-designed functionality. User participation might become a means to 'find the thorns' (Kawalek

ERP implementation as design • 31

and Wood-Harper 2002) of the standard solution and doing only the most necessary changes rather than designing a meaningful socio-technical system. Looking at our case this perception does not match the reality we observed. Already the configuration of ERP systems requires an informed choice between different predefined options, often mutually dependent, which can have effects across the whole organisation. If customizations become necessary, e.g. because the standard does not support parts of the core business, the add-ons have to be designed to both meet the user requirements and fit with the standard system. In both cases, knowledge about the technical options might become more important than in development from scratch.

One of the seminal PD papers discusses knowledge development and integration in systems development. Kensing and Munk-Madsen (1993) define design as "bridge-building, since something new is created from two separate things" (p. 79). They claim that in the design of a new IT system, three knowledge domains are involved: the users' present work, the technological possibilities, and the future system the process results in. Knowledge generation depends on successful communication and "successful communication depends on the ability to establish situations in which mutual perturbations trigger changes in the state of those involved, which in turn lead to structural congruence among communicating partners" (Kensing and Munk-Madsen 1993, p. 79). Therefore, methods, tools and notations supporting knowledge generation and integration are especially important. Kensing and Munk-Madsen distinguish between abstract and concrete knowledge in the three different knowledge areas. The six resulting knowledge areas (two-by-three) are:

1. *Concrete experience with the users' present work* has to be acquired by the developers involved in order to be able to develop representations of this work relevant for the design process. Developers also need the knowledge to be able to understand the limitation that the work context poses for the technological support. The methods to achieve this learning are for example: apprenticeship with users, participatory observation, and interviews (Kensing and Munk-Madsen 1993).

2. *Relevant structures of users' present work* address the abstract knowledge of users' present work. Kensing and Munk Madsen emphasize that the kind of structures that is relevant depends on the purpose of the software. The professional abstractions the domain experts use might be adjusted and complemented to represent structures of the use context that are necessary in order to design adequate technological support. The standard software engineering analysis notations and representations, as well as less formalized representations like wall graphs and rich pictures (Kensing and Munk-Madsen 1993), can be used. In the context of ERP system implementation, models of the as-is business processes can be regarded as supporting the development and representation of this category of knowledge.

3. *Concrete experience with technological options* belongs to the professional realm of the developers. Users have to acquire knowledge in this domain in order to be able to anticipate the deployment of new technology to support their current work practice.

4. *Overview of technical options* supports the informed decision between different implementation alternatives. Kensing and Munk-Madsen propose literature studies as a way to address this area of knowledge. The technical options in the context of an ERP system

32 • Pries-Heje & Dittrich

implementation are constrained by the configuration and customization possibilities a standard package is providing. Reference models packaged together with the specific ERP systems can be seen as an attempt to show the different possibilities that a specific ERP system can support.

5. *Visions and design proposals* denote the abstract knowledge of the future software and its usage. Kensing and Munk-Madsen focus on representations that mediate cooperation in design. They suggest that software engineering design methods and notations should be complemented by proof of concept prototypes; scenarios and system visions should be used as means to cooperatively develop the knowledge in this area.

6. *Concrete experience with the new system* will make visible how the work practices are influenced. Here mock-ups and also experience with prototypes and similar techniques allow users to evaluate the usage of the software under development with respect to their former and future work practice.

Configuration and customization of standard systems is not a well-researched topic in PD. The configuration and customization of ERP systems integrating business processes across different departments have to take a heterogeneous user community into account. Few methods have been developed and tested for this purpose. An exception is the acquisition and implementation of a hospital information system documented in (Krabbel et al. 1996) and (Krabbel and Wetzel 1998). Krabbel et al. (1996) propose combining observation and interview-based task analysis resulting in scenarios about the present work practice and cross professional workshops. Moderation in the common workshops has to focus especially on allowing different members of the organization to contribute, thereby allowing them to put forward requirements and constraints in relation to their specific work practice. Krabbel et al. propose what they call 'point of view' pictures as a means of representing professional and role specific perspectives. The cooperation between different user groups and the change in their cooperation through the introduction of the standard system needs to be represented as well. Here, Krabbel et al. propose using a specific adaptation of rich pictures to visualise different channels of communication around complex tasks—like admitting a patient to the hospital—and the changes that will be implied through the implementation of an information system.

The follow-up article from 1998 (Krabbel and Wetzel) indicates that the task analysis did not prevent the implementation process from becoming problematic. The authors mention, for example, the lack of adequate specification of customization tasks (in their definition comprising both customization and configuration), problems with the flexibility—or rather the inflexibility—that the software provides, and organizational change.

Although the implementation of ERP systems is not well researched in the PD community, the PD concepts, methods and tools mentioned above might provide inspiration for the facilitation of ERP system implementation understood as design. We return to this topic in our discussion. In the analysis of our case study, we use the framework developed by Kensing and Munk-Madsen (1993) to show that problems experienced during the ERP implementation can be explained as design issues related to knowledge development and knowledge integration problems. The discussion section takes up the implication of this shift in understanding.

ERP implementation as design • 33

4   Research method

The aim of our research was to gain a deeper understanding of the difficulties when implementing ERP systems. Agreeing with Orlikowski and Iacono (2001), we focused on the IT artefact and the finalisation of its design during the implementation process. We decided to use a detailed interpretive case study in line with the interpretive tradition of information systems studies (Klein and Myers 1999). We based our analysis on the participants' descriptions of the implementation process and their reflections regarding the usefulness of the tools and methods applied.

Data collection was carried out through interviews with the ERP project manager, users serving as team leaders during the implementation (some of them later moved to the internal ERP competence centre), managers and end-users. The interviewees from Alfa's organization were selected to ensure a broad coverage of all functional areas within the scope of the ERP project. We were lucky that a consultant participating in the project on the vendor side, and the vendor's solution architect also agreed to participate in interviews. All 18 interviews were semi-structured and lasted 1½ to 2 hours. The interview guide included open-ended questions regarding experienced misfits, the interviewees' involvement in the ERP implementation over time, cooperation between user representatives and IT experts, tools and techniques used for requirement specification and design work, etc. The interviews were taped, transcribed and verified by the interviewee. The table below shows the number of interviews within different groups and the timeframe for these.

Role in the ERP implementation	Number of interviews	Interview periods
ERP Project Manager	3 interviews	February 2005, January 2006, May 2007
4 people from the internal ERP competence centre	1-2 interviews each	August - November 2005 June – August 2006
Vendor's solution architect and one consultant	1 interview each	November 2005, February 2006, July 2006
7 end users	1 interview each	February 2006 – June 2006
In total		*18 interviews each lasting 1,5 to 2 hours*

Table 1: Overview of interviews

As indicated in figure 1, the ERP implementation started in 2001. The research project only started in the beginning of 2005. Therefore, it was not possible to follow the project from its beginning. Thus, one part of the interviews was conducted with a retrospective focus and another part focused on the current situation. One of the problems using this approach was that the interviewees' interpretations of the past were influenced by events taking place later. Written project documentation was therefore used to verify the interviews where possible, and contradictions and conflicting statements were put forward for the interviewees to comment on. Alfa provided elaborate documentation including detailed requirement specification, documen-

tation from the evaluation of the candidate systems, business cases, gap analysis, and issue-log and change requests. All written documents were provided after the first interview, and were used preparing the following interviews. Thus the interviewee's role in the development of each artifact was discussed with the interviewee, along with the perceived usefulness of the artifact in the design process, during training, and after going live.

Data analysis took place as a hermeneutic interpretative process (Klein and Myers 1999); that is, the data analysis was an iterative process going back and forth between coding and collecting data. We adopted an inductive approach, and did not specify theory a priori to guide the data collection. As the data were analyzed, relevant theories were investigated. We entered the research with a 'bias': being aware of the practical difficulties in taking advantage of pre-defined ERP software, our intention was to understand how ERP systems could be implemented so that they were useful and easy to use for multiple groups of end users. As the analysis progressed, we consulted different streams of literature that could provide insight into the empirical observations, e.g., literature that considers ERP implementations, user participation, knowledge integration, and design politics.

It became visible that knowledge issues played a major role in the implementation process, even as the effect of a knowledge breakdown did not necessarily show immediately. We therefore looked for a way to analyse the data that allowed us to focus on the process. The description of the case is organized using a process perspective separating the temporal trajectory into five episodes. Alfa's ERP experience can that way be divided into five distinct episodes (figure 1): (1) Requirement specification and ERP package evaluation, (2) Configuration and customization of the ERP package, (3) Training, go-live and stabilizing the new ERP system, (4) Follow-up project, and (5) Redesign and re-introduction of functionality. The separation between two episodes is motivated by major events changing the knowledge dynamics in the project. The episodes resemble the ones proposed by Markus and Tanis (2000). Their framework divides the ERP implementation process into four stages; Project chartering, The project (configuration and rollout), Shakedown, and Onward and Upward. Based on our analysis, we decided to split configuration, customization and training into two different episodes because an important change in the knowledge integration and the participation was observed. This resulted in five episodes. It could be interesting to compare our findings with other process models. As the process model in the context of the article is an analytical tool, rather than a result, this would lead us beyond the scope of this article.

Within each episode, the analysis of the knowledge integration issues was based on Kensing and Munk-Madsen's framework. The framework identifies six knowledge areas that should be covered and integrated when developing IT systems. Our perception of design as insight building fits well with Kensing and Munk-Madsen's understanding of IT development as knowledge integration: bridging knowledge about the technology and the user organization as a base for the design of the future system. Knowledge and knowledge integration are to a great extent taking place inside the head of people and is thus not observable. In the analysis, we therefore focus on actual activities and their traces e.g. in form of documentation that had—or could have—promoted knowledge development and integration as well as indications for the success of knowledge integration or the lack thereof. We therefore talk about 'knowledge integration capabilities'.
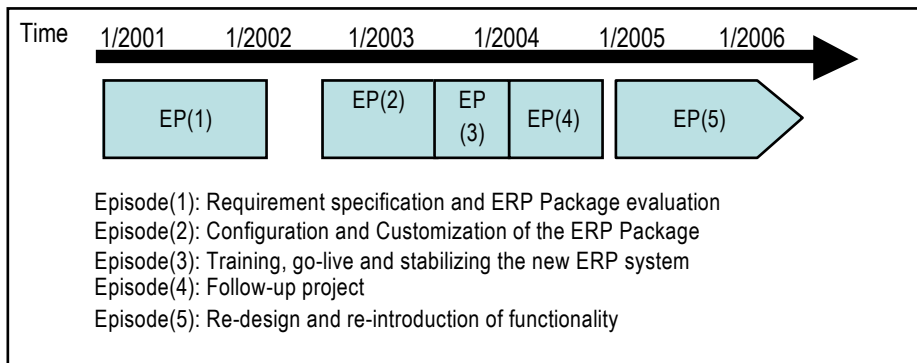
ERP implementation as design • 35

Figure 1: Timeline for the 5 episodes in Alfa's ERP experience

5 Introduction to the case

Alfa is an engineering company with more than 80 years of experience in supplying engineering services to the pharmaceutical and biotechnological industry. The organization has 1.200 employees in Europe, China and USA. A large number of the employees have a degree from typically a technical university. Most of the work in Alfa is conducted in large projects lasting several years and costing billions of US$.

Alfa came to the conclusion that an integrated ERP package providing real-time data sharing was necessary in order to enhance the quality of services offered to the customers, improve resource management, and provide better financial control. Managers as well as users were aware that it would require the organization to adapt to the ERP system.

The ERP project started in January 2001 at Alfa's headquarters in Denmark. A project manager with extensive ERP project management experience was hired. From the very beginning, the ERP selection and implementation were regarded as a joint project for management and employees in Alfa. It was never questioned that users would participate throughout the project as they always had in comparable projects. A project organisation was set up and user representatives for each functional area were appointed.

Alfa's core business is project administration and project management on behalf of their customers. The company does not produce or manufacture any physical products. Alfa was aware that ERP systems in general do not target this line of business. A thorough evaluation and selection process was conducted to ensure that the standard system would meet their needs. Alfa spent a year specifying requirements, evaluating candidate systems and selecting a system. It took more than 6 months to arrive at a final approval of the project, and then 9–10 months for configuration and customization of the system before going live in October 2003. In 2004, a follow up project was carried out addressing some major issues in using the system, and since January 2005 the organization has continually been implementing (minor) re-designed functionality as well as new functionality.

Before the decision to implement the ERP system, Alfa had very limited experience with standard systems, and no experience with systems that worked across functional areas in the organization. Only a few user groups had concrete experience with modified standard software used in their daily work. Users, the ERP project manager and top management at Alfa acknowledged the need for a new system and the intended approach. Thus, the project started out being widely accepted.

5.1 Episode 1: Requirement specification and ERP package evaluation

Before the specific ERP vendor was decided upon, all business processes within the scope of the new system were described using PowerPoint as a tool. The processes were related to four areas: finance, purchasing, project administration, and resource management. A large number of users throughout the organisation were involved in the process. The business processes served as a common reference for discussing the requirements, focusing on input data triggering a process, steps within and output from a process.

More detailed requirements for each area were defined in a dialogue between the project manager and the participating users. This turned out to be a difficult process as it involved a large number of users who had little or no experience at defining requirements. Alfa strived to have the requirements reflect existing processes and at the same time to be forthcoming towards processes within a standard system. Because of the users' limited experience with integrated standard systems, they did not know what to expect from an ERP package. To inspire them, a few ERP packages were demonstrated by different vendors.

Alfa defined more than 800 detailed requirements that were subsequently prioritised on a scale of 1–4. The requirements were then mailed to the candidate vendors, and the vendors sent a written reply; for each requirement they defined to what degree it could be supported by their system. In parallel with the requirements definition, a set of criteria for evaluating the vendor was defined. Knowledge about the industry and the vendor's desire to understand Alfa's business were among the more important criteria.

Based on the written replies, three vendors were invited to demonstrate their system in an all-day workshop partly using material defined by Alfa. 10–15 users participated in the workshops and evaluated the system and vendor performance based on an evaluation framework. A group of three people (IT manager, project manager and a user representative) visited implementations of the candidate ERP systems. The results from the evaluation process were summarized and presented as quantitative and qualitative scores in a number of different areas. A recommendation to the board of directors indicates the end of this episode.

5.2 Episode 2: Configuration and customization

Alfa's board of directors decided to follow the recommendation given by the project group, and Oracle was chosen as Alfa's new ERP system. Some of the users participated in general training (3–5 days) in using the ERP system provided by Oracle during the time that the con-

ERP implementation as design • 37

tract was being negotiated. Due to financial difficulties, the ERP project was asked to cut the project cost by about 700.000 € before even starting. To re-scope the project, Alfa's ERP project manager and user representatives from the different functional areas together with consultants implemented a 'Conference Room Pilot'; for each requirement, the implementation consultants would show their solution in Oracle. This process made visible that it would be necessary to add requirements as well. After two weeks, the revision was decided upon and a contract defining scope, price and so on was signed.

In the following nine months, three more Conference Room Pilots were conducted. They can be seen as iterations in the configuration process. Each time, the system to-be was (re-) scoped at a more detailed level and the configuration decisions were documented. The work was conducted in small workshops with user representatives and the consultant(s) from Oracle focusing on specific modules of the ERP package. As a part of the implementation method, the configuration and walkthrough of the system took its outset in the business processes pre-defined in the system. Oracle's process tool was used, linking the business process diagrams to the application. For each process, Alfa appointed a responsible user as 'process integrator'. The process integrator had the task of focusing on the interfaces and coordination between processes. Alfa documented the new processes and the configuration decisions very conscientiously.

Teambuilding activities were conducted throughout episode 2. Although the project was under time pressure, the Oracle consultants and Alfa's user representatives worked together in a good atmosphere. The project manager worked explicitly with the aim of creating a team spirit. The beginning of the training period indicates the transition to the next episode.

5.3 Episode 3: Training, go-live and stabilizing the new ERP system

At this point the project was under extreme time pressure. Within Alfa, it is not allowed to implement a new financial system during the last quarter of a financial year. Therefore, the system had to go-live at the beginning of October 2003. The training of the users took place alongside the final testing and data conversion. On the 8th of October 2003, Alfa's Oracle solution went live. Because of the time pressure, many reports were not yet implemented and consequently much promising functionality was left to a later phase.

An important change in the ERP organization and the roles of participants happened during this episode. The external ERP specialists stepped back a little and allowed the user representatives to take over the role as ERP specialists in the organization. Thus, user representatives developed the training material and performed the training of end users. At the same time end users entered the stage.

During training, resistance toward the system built up. Users perceived the system's usefulness to be very low and, e.g., prohibited that the project management module for allocating project members to tasks was taken into use. During the next months, the users were struggling with the system, learning to manage parts of the system. Other parts were rejected or used incorrectly, which in turn caused data quality problems and malfunctioning in other areas. In general, the users had difficulties understanding how their personal use (or rejection) of the system influenced the work of other departments.

After a very turbulent period lasting almost a year, the system was stabilized and the most important reports were developed. An internal 'ERP competence centre' was formed consisting of the project manager and some of the user representatives, plus a former Oracle consultant who was hired by Alfa.

5.4 Episode 4: The follow-up project

Members of the ERP competence centre suggested including all users in an evaluation of problematic issues. Meetings were set up where people from the competence centre met all user groups within Alfa. The analysts met the users with an open mind. All issues reported were noted. The process resulted in a list of more than 500 issues. Afterwards, the root causes for the issues were discussed and appropriate actions decided on. Some issues were obviated with end user education, some with reconfiguration or additional customizations, and some were researched thoroughly but could not be solved due to the ERP package architecture. Regarding the IT artifact changes included (re-)design of reports, minor and major changes to screens, development of new organization specific web-based user interfaces for some critical functions, changes to workflows, taking functionality out and including new, and much more. At the end of 2004 this process was completed, although the re-design of some of the problematic processes was still outstanding.

In general, the users' perception of the usefulness of the software was slowly starting to change. However, many users still avoided using the system or enacted it in ways that caused as little change to the old work processes as possible, e.g., by delegating ERP related tasks to secretaries.

5.5 Episode 5: Re-design and re-introduction of functionality

In the final episode, the focus was on improving the use of the system, re-design of the already implemented functionality, and design of new functionality. End users throughout the organization and members of the competence centre were continually working to increase the quality of use. They customized the software to change the original capabilities of the ERP package, reconfigured the system and used the software in unanticipated ways.

The relations between the users and the internal ERP competence centre were still somewhat tense. The ERP experts, however, considered the functionality within the new system to be useful for most parts. Users' complaints were seen as resistance to change; yet, some of the ERP experts engaged in a dialogue with the users. Members of the competence centre communicated directly with the functional managers and end users to help them understand the new ERP software and the ERP package's capabilities. ERP experts observed users using the system and engaged in discussions. Super users and end users began to help each other on an ad hoc basis across functional departments. User groups formed to share experiences. Employees from the competence centre and some end users also participated in groups outside Alfa sharing experiences about the Oracle ERP system. The general perception of the usefulness of the system improved although some users still avoided using the system.

ERP implementation as design • 39

6 Knowledge integration and knowledge integration capabilities

Alpha provided for a very thorough representation of the future users in their ERP implementation process, as the overview given in the previous section shows. So why did the project team discover major mismatches only late in the process? Could the problems have been discovered before? Below, we analyse the development of the knowledge integration capabilities of different actors. However, instead of the two groups, users and IT professionals that Kensing and Munk-Madsen (1993) discuss, we find four different groups of actors throughout the implementation process: User representatives started together with external consultants, who acted as ERP experts during the first two episodes. Thereafter, they leave the scene. During episode 3, some of the user representatives acted as experts on behalf of the ERP project team. Their role got more pronounced, as the external consultants either left or were employed by Alpha. The further design and implementation now took place between this group of internal ERP experts and the end users. The final section sums up the development of the knowledge integration capabilities and highlights issues for further discussion.

6.1 User representatives' knowledge integration capability

As described above, Alpha provided for a thorough representation of the future users to ensure that the specificities of the different professional practices were supported in the best possible way. This group partly built the core of an internal ERP competence centre. In this section, we analyse their knowledge integration capabilities before this move takes place. Kensing and Munk-Madsen's (1993) framework introduced in section 3 is used as an analytical tool. The framework has six knowledge areas and we analyse five knowledge integration areas within the framework: (A) integrating knowledge area one and two; (B) integrating knowledge area three and four; (C) integrating knowledge area four and five; (D) integrating knowledge area two and five; (E) bridging knowledge area five and six.

Episode 1: Requirements and systems evaluation: The user representatives met already before the specific ERP vendor was decided. They took part in developing the requirement specification and evaluating the different vendors. Their knowledge integration capabilities during that first episode can be analysed as follows:

A. *Present work abstract-concrete level:* The user representatives were employed by Alfa and had extensive practical experience within the functional domain they represented. However, the users' knowledge about cross-functional work processes was limited. As a base for the requirement specification, existing work processes were articulated within functional groups using PowerPoint as a tool. Thus the user representatives' ability to integrate knowledge at the abstract-concrete level, i.e. relate present practice to the abstract representations, was good.

B. *ERP Package abstract-concrete level:* The vendor demonstration provided a very shallow abstract understanding of the specific ERP package, and no firsthand practical experience. The user representatives we interviewed, indicated that they only much later realised that watching consultants operate the system gave a false impression of the ease of use and the systems' (poor) handling of exceptions and error situations. Furthermore, the users had no or very limited understanding of how different chunks of functionality mutually excluded each other or what derived effect (e.g., across modules) a specific parameter setting would have. Thus knowledge development within the two areas was limited and virtually no bridging of the abstract and concrete level was possible.

C. *ERP Package – New system abstract level:* Here the written reply to the requirement specification, including the suggested customizations, and the vendor's demo was the only support for knowledge development. Consequently, the user representatives' knowledge within the two knowledge areas and their ability to bridge them was very limited

D. *Present work – New system:* The all-day workshops performed by the vendors gave the users participating an impression of the look and feel of the system as well as the 'chemistry' between the vendor's consultants and Alfa's participants, but very limited knowledge to relate the present work to a possibly new (customized) system. Although a formal evaluation framework was developed, Alfa's participants honestly admitted that the evaluation was based primarily on intuition, as well as the look of the user interface and the interaction with the consultants. Thus, during episode 1, the user representatives only developed a very weak idea about the new system and their ability to relate the present practice to a future situation on an operational level was virtually impossible.

E. *New system abstract-concrete level:* The standard ERP software was (in theory) available. So were process diagrams representing the functionality of the ERP package. In practice, the complexity of the ERP package made it impossible for user representatives to operate and make sense of the ERP package on their own, and in Alfa's case it was decided not to make it available to the user representatives in episode 1. The vendors' written reply was the only base for the user representatives' participation, no organization specific version of the new system existed. Thus this area was not covered in episode 1.

Episode 2: Configuration and customization: The user representatives participated in the actual scoping, configuration and design of customization in cooperation with the vendor consultants.

A. *Present work abstract-concrete level:* The requirement specification and some of Alfa's work documents were used in the dialogue with the ERP experts. The process diagrams (see episode 1) were dismissed by the ERP experts, and no further attempts were made to articulate Alfa's present work. Thus bridging the concrete-abstract level of present work was difficult.

B. *ERP Package abstract-concrete level:* As a basis for participating in the design activities the user representatives was given some training in the ERP package. Later, design workshops were performed; here, the ERP experts demonstrated the capability of the ERP

ERP implementation as design • 41

package and possible Alfa specific solutions was discussed. Pre-defined process diagrams illustrating the work processes implied by the ERP software were supplied with the ERP package and were used together with a modelling tool during the workshops. User representatives found them too abstract and open for interpretation. They often caused false perceptions of common understanding. Summing up, the user representatives developed some knowledge about the ERP package and its abstract representations, and as the knowledge increased their ability to bridge the abstract-concrete level also improved.

C. *ERP Package – New system:* The process diagrams did not provide knowledge about configuration options. Therefore, the user representatives had to gain an understanding of the technological options through discussion with the ERP experts. Often, the ERP experts would collect information from the user representatives, set up the system, and then come back to show the user representatives the result. Because large parts of the future system could be finalized using configuration, the modified (configured) ERP package served both as an advanced prototype and an emergent finalised design object. Due to the fact that user representatives had gained limited knowledge of the ERP package they had to rely on the ERP experts to explain its capabilities and develop design suggestions. Their own ability to bridge the capabilities of the software and the design of the new system was limited.

D. *Present work – New system***:** The user representatives very quickly experienced difficulties in letting the pre-defined process diagrams guide the mapping of the requirement specification to the new system. As one of the interviewees expressed it: "Now everything was twisted, we had formulated our requirements based on our business processes, but Oracle required us to use their processes configuring the system and designing the new business processes … e.g., in our minds the time aspects of a project are related to the financial process, but in Oracle it is part of the project process." The project team worked in functional groups related to the modules of the ERP package. The use of the pre-defined process diagrams made it difficult to use the experience with present work practice when trying to anticipate how the suggested processes would work in the organization. Apart from the emergent "prototype" of the ERP software and the requirement specification developed during episode 1, no shared representations of present work or the future system were developed. The user representatives each had to find their own way to relate the existing practice to the design suggestions. The complexity of the ERP software, especially the cross module dependencies was a major hinder. Developing test cases and performing tests of the new system provided another opportunity for a reality check. The user representatives, however, admitted that also the test cases to a large extent had a module focus. Only a limited number of people with specific domain knowledge were involved and the tests were not based on real live data/situations. Furthermore, testing customizations drew much attention. Evaluation of the usefulness of the design suggested was neglected.

E. *New system abstract-concrete level:* As described above (B + C), using the pre-defined process diagrams was very difficult. Thus they had limited effect mediating the com-

munication during the design process. The project manager acknowledged: "Looking back I can see that we were wrong assuming that sitting together with the users (user representatives) defining the new processes would make them work in practice …" The activities performed and the techniques used during episode 2 constrained the user representatives' ability to bridge the abstract-concrete level.

6.2 Consultants' knowledge integration capability

In the first two episodes, external consultants had the role as the IT professionals in the design, development, and implementation process. Most of the consultants were 'application consultants', meaning that they were specialists within a specific module, and knew the software's functionality and configuration possibilities seen from the use side rather than the technical side. A senior consultant, 'a solution architect' with extensive experience implementing the ERP package and cross-module knowledge, was associated with the project and participated on a need basis. Finally, technical consultants, e.g., database experts or programmers, participated in the technical installation of the software and implemented the customizations specified. In this analysis the focus is on the application consultants, because they were the ones actually participating in the design work performed.

Episode 1: Requirements and systems evaluation: Though the consultants were not participating in the preparation of the tender process, they entered the scene when preparing an offer based on the requirement specification.

  A. *Present work abstract-concrete level:* Some of the consultants had practical experience and/or an education related to one of the functional domains covered by the ERP software, but they had no concrete experience from Alfa's organization. During the first episode, the requirement specification and some additional information about Alfa were the only actual knowledge they had about Alfa's work practice. The PowerPoint process diagrams developed as basis for the requirement specification were not shared with the vendors. The consultants' knowledge within the two knowledge areas was very limited, as was their ability to bridge between them.

  B. *ERP Package abstract-concrete level:* The consultants were familiar with the ERP package, the related training material and internal documentation. They had experience implementing ERP software in other organizations. Thus, their ability to bridge between abstract representations of the ERP software and the concrete level was relatively good regarding the modules under their responsibility.

  C. *ERP Package – New system:* Because they knew the ERP package well and had seen it work in other organizations, the consultants had an idea how it could work. Their perception of Alfa's new system was, however, only supported by the requirement specification.

  D. *Present work – New system*: Bridging knowledge from these two areas was only facilitated by the requirement specification during episode 1.

ERP implementation as design • 43

E. *New system abstract-concrete level:* Due to the limitations in knowledge development explained in C and D, the consultants' understanding of the new system was mainly based on their knowledge about the ERP package. They had not yet developed abstract representations of Alfa's specific context.

Episode 2: Configuration and customization: After the tender process the actual design and development began. Note that one of the first activities was a re-scoping in order to save part of the costs, which also resulted in a revised requirement specification, fitting with the specific ERP package.

A. *Present work abstract-concrete level*: The requirement specification and organizational work documents: contracts, invoices, legal information, etc., were the only formalized representations of the present work used during episode 2. In some of the configuration workshops, ad hoc drawings were constructed around a white board, but seldom preserved. Only one of the four functional groups spent a few hours visiting the related work place. Thus, the consultants' firsthand experience with Alfa's work practices was very limited. Consequently, their ability to interpret the (few) abstract representations and discuss them with the user representatives was to a large extent based on their previous knowledge about similar work processes in other organization.

B. *ERP Package abstract-concrete level:* See episode 1.

C. *ERP Package – New system:* Using Oracle's process tool and the pre-defined process diagrams (design proposals) made the consultants relatively comfortable as they could use their existing knowledge about the ERP package when picturing the new system. Unfortunately, the processes mirrored the module structure of the ERP system and therefore provided limited cross-module knowledge development. Thus, in the third iteration of the so-called conference room pilot, a major knowledge breakdown was experienced: Both the finance and the project management consultants expected the other group to provide a solution for a specific requirement, but none of them was able to. Summing up, within each functional module the application consultants were able to bridge the ERP package—new system knowledge fairly well, but they lacked that capability for cross module issues.

D. *Present work – New system:* The consultants relied on the users to make the mapping. As long as the design proposals did not challenge the scope of the project and the users provided the necessary input to configure the ERP system, the consultants were not concerned with the existing or the future work practice.

E. *New system abstract-concrete level*: The consultants were able to go back and forth between design proposals and the configured ERP software. However, they were not able to evaluate the usefulness of the new system in the organization. The requirement specification was still used as a checklist.

6.3   Internal ERP experts' knowledge integration capability

In this section we meet the user representatives again. A part of them slowly developed to become the core of an internal ERP competence centre.

Episode 3: Training and go-live: The impact of the implementation became visible, when the user representatives took part in introducing the new system and developed training material and documentation for the other users.

- A. *Present work abstract-concrete level*: The abstract representations developed during episode 1 were of no use in this episode, and since no additional representations of the existing work practice had been developed during the design process, no shared artefacts articulating the existing practice existed. Thus, when communicating with end users, e.g. explaining the changes, the internal ERP experts only had their own experience as employees in Alfa to draw on.

- B. *ERP Package abstract-concrete level:* The internal ERP experts gained more insights into the abstract representations of the ERP package and concrete experience with the ERP package when developing the training material, performing end user training and testing the system. Their ability to bridge the abstract and concrete level increased.

- C. *ERP Package – New system:* The understanding of the ERP package developed with understanding of the new system. Since the internal ERP experts developed more knowledge within both areas, their ability to bridge them also increased.

- D. *Present work – New system*: More knowledge about the factual properties of the new system, training end-users, and helping them make sense how to work when using the new system, increased the internal ERP experts' ability to bridge knowledge between present work and the new system. In the communication with end users and evaluating the design of the new system, they could draw on both their experience with the development of the new system and their work experience.

- E. *New system abstract-concrete level*: As the internal ERP experts participated in developing training material and process documentation, and gained more hands on experience with the new system, their ability to bridge the abstract-concrete level improved.

Episode 4: The follow-up project: After going live, the internal ERP competence centre was in charge of supporting the users, collecting feedback about problematic issues and initiating redesign. The present work now included the use of the existing ERP system. When talking about the new system, we refer to modification of the existing ERP system.

- A. *Present work abstract-concrete level*: Most of the internal ERP experts were no longer involved in the daily work after the ERP implementation. Their first hand experience of the present work practice changed and their ability to bridge the abstract-concrete level to some extent decreased. Of course, their prior experience and belonging to a specific functional domain within Alfa was an advantage in relation to communicating with end users.

ERP implementation as design • 45

B. *ERP Package abstract-concrete level:* By working with Alfa's specific version of the ERP system, the internal ERP experts also gained insights into the capabilities of the ERP package. This knowledge increased their ability to bridge the abstract-concrete level, e.g., cooperating with external ERP experts or interpreting generic system documentation provided by the vendor.

C. *ERP Package – New system:* The increased understanding of the ERP package's capabilities and its internal design enabled them to develop more qualified design suggestions.

D. *Present work – New system:* As already discussed under (A), the internal ERP experts had the advantage of prior firsthand experience within a functional domain in Alfa's organization. Compared to the external consultants, their ability to communicate with the users and relate to the work context was much better.

E. *New system abstract-concrete level*: The internal ERP experts' ability to bridge the abstract-concrete level improved due to increased experience with the ERP system and the design artefacts used.

Episode 5: Re-design, re-introduction and design of functionality: One year later, the re-design process moved into a second iterative circle, now the internal ERP competence centre was well-established in its new role.

A. *Present work abstract-concrete level:* See episode 4

B. *ERP Package abstract-concrete level:* Over the years, the internal ERP experts developed a good understanding of the capabilities of the ERP package. Their ability to understand abstract representations like system documentation and communicate with external ERP experts was well developed.

C. *ERP Package – New system:* Due to the improved knowledge about the ERP package's capabilities and experience with ERP design work, their ability to develop and evaluate design suggestions improved dramatically.

D. *Present work – New system:* Informal user networks were formed in Alfa, and some of the internal ERP experts were included. In these networks, knowledge about the ERP system was shared, and solutions to difficulties using the ERP system were discussed. As a result, important cross-functional process knowledge was developed. Thus, engaging so close with the users and expanding their knowledge about the ERP system's capabilities resulted in better-informed design suggestions, i.e. design suggestions that better met the needs of the organization.

E. *New system abstract-concrete level:* The internal ERP experts, at times, experienced difficulties predicting how the new system would be received in the organization. Some functionality was re-designed several times without achieving real success. However, in general new or re-designed functionality was adopted more easily in the organization at this stage.

6.4   End users' knowledge integration capability

As discussed above, the end users entered the scene again when the user training started and the system was taken into use. Below the end users' knowledge integration capabilities during episode 3–5 are analysed.

Episode 3: Training and go-live: During end user training, it became clear that some of the defined processes would not work in practice. The result was hostility towards the new system. Some of the planned functionality was taken out just before going live.

  A. *Present work abstract-concrete level:* The implementation approach did not focus on generating shared abstract representation of present work. No cross-functional representations were developed that allowed an integration of knowledge across the organization. Thus, this area was hardly covered at all.

  B. *ERP Package abstract-concrete level:* As no design was done in this episode, no new abstract knowledge within this area was relevant.

  C. *ERP Package – New system*: See (B).

  D. *Present work – New system:* Given the complexity of the new system, it was difficult for the users to develop an adequate understanding and find ways to perform their work in an effective and efficient way. Additional training, new training material including parts of the processes not supported by the ERP system, and local documentation of procedures, were developed. However, most of the knowledge had to be built by using the system. Thus, in the beginning the knowledge integration capability was very limited, but improved somewhat during the episode.

  E. *New system abstract-concrete level:* The new processes documented in Oracle's process tool was supposed to help the end users build knowledge about the new ERP system and understand how to perform the new working tasks. They were, however, never put into use. The end users did not understand them, and, in many cases, the new processes could not work. Due to the difficulties using these abstract representations, the support for understanding the complex ERP system at the concrete level was limited.

Episode 4: The follow-up project: In this stage, the users in cooperation with the internal ERP experts start to make things work. Now the practical experience with the ERP system facilitated the learning process.

  A. *Present work abstract-concrete level*: Ad hoc material was developed, helping to facilitate communication within and across functional departments.

  B. *ERP Package abstract-concrete level:* Practical experience with the ERP system improved the users' ability to understand abstract representations. Due to the complexity of the system, users experienced difficulties understanding how local changes would affect other parts of the system. Thus, their understanding of "technological options" and derived consequences only improved a little.

ERP implementation as design • 47

C. *ERP Package – New system:* Practical experience with the ERP system helped users individually explore the ERP package functionality. Their ability to understand and contribute with design suggestions improved accordingly.

D. *Present work – New system:* Experience with the ERP system improved the ability to envision how re-designed or new functionality would work. Nonetheless, to fully understand the consequences of a design proposal before going-live remained a challenge.

E. *New system abstract-concrete level*: Additional training in the use of the new system was given. Users helped each other, that is, they developed and shared small representations of important functionality. Users who had participated in the configuration and customization contributed official 'quick guides'. However, the complexity of the new system caused difficulties to generate sufficient abstract knowledge regarding dependencies between different parts of the system.

Episode 5: Re-design, re-introduction and design of functionality

A. *Present work abstract-concrete level*: Over time, very detailed user manuals were developed. Thus, in familiar areas, the users had good possibilities of bridging the concrete and abstract levels related to the use of the ERP system. In general, the users did not engage in developing or interpreting codified abstract representations of the post implementation work processes. Some ad hoc groups worked at improving the use of the system and/or re-design of functionality. In case re-design of ERP functionality was needed, users were requested to use a specific template when specifying requirements. In the controlling department, users worked systematically to develop a controlling manual (process documentation) in order to provide a common understanding among controllers how to use the ERP system. This documentation exposed flaws in the design of the ERP system that resulted in poor data quality. Unresolved taxonomic issues, e.g., impacting comparability across project and over time, were discovered.

B. *ERP Package abstract-concrete level:* In general, the users did not explore the standard version of the ERP software, but focused on the configured and customized software. Users who engaged in re-design and redefinition of data expressed great difficulties in understanding the complexity of the software.

C. *ERP Package – New system:* See B.

D. *Present work – New system:* Some users only developed a fragile understanding of how to use the ERP system for their present work. Informal networks of employees were formed, providing a well-developed cross-functional understanding of how to use the system. Users who engaged in re-design were typically part of such informal networks and had a good understanding of the software and cross-functional work practices.

E. *New system abstract-concrete level*: Alfa's users still experienced some difficulties evaluating design suggestions, especially when these involved customizations to be programmed externally.

6.5   Summing up the five episodes

How episodes 1–3 were performed shows that in practice a waterfall-like approach was used. Despite iterations of the conference room pilot, the process provided no means to achieve the necessary knowledge integration. In episodes 4 and 5, the knowledge integration capability had changed dramatically. When re-design was requested (re-configuration, re-definition of data or customizations), each participant covered more knowledge areas, and the communication between users and ERP experts was supported by shared experiences and a shared vocabulary. The internal ERP competence centre still required change requests to be initiated by a requirement specification and a business case, but practical experience with the system and informal cross-functional networks in the organization provided insights that allowed a cooperative design approach.

Figure 2 provides a graphical illustration of the knowledge integration capabilities of user representatives/end users and the ERP experts during each episode. Arrows with different filling are used to illustrate the vigour of the integration capability between two knowledge areas: An empty arrow illustrates a weak ability to bridge the two knowledge areas caused by no or very limited knowledge in one or both knowledge areas; a hatched arrow indicates that some knowledge in both areas was developed, allowing some bridging; and a solid arrow means that extensive knowledge in the two knowledge areas was developed, allowing serviceable bridging.

Prior to episode 3 (testing and training), neither the user representatives nor the IT experts (consultants) had developed knowledge within all six knowledge areas. In that regard, episode 2 has two important weaknesses:

First, the external IT experts used a design strategy that involved user representatives as informants only. As our field material indicates, it was not possible for the user representatives to relate their knowledge well enough to the design proposals. This weakness resulted in several serious design issues. One example is related to the design of project management functionality: To update the allocation of a project member on a task in a project, data had to be entered on three screens for each person on each task for each month. Considering the fact that project managers have to update the allocations at least once a month, and that a project can have 200–400 people working on one or more tasks over long periods of time, this design was not acceptable.

The second weakness during episode 2 was the lack of cross-functional, cross-module knowledge integration, both among user representatives and external IT experts. Only shortly before going live, a configuration workshop including all the functional teams was performed. The workshop brought up several cross-functional design issues at this very late stage. One of the issues related to joint work between accountancy and project management around project budgets was so serious that it almost closed down the project - a major customization was necessary. Prior to the workshop, the external ERP consultants in both the finance and the project management group had assured the user representatives that the functionality was provided in the other module.

Another example: The ERP system's functionality providing coordination between the purchase department and project management anticipated that an item number referring to a unique item or a bill of material would be a useful way to specify what should be purchased. In Alfa's case, most of the material used in projects had to be designed to order (at the vendor
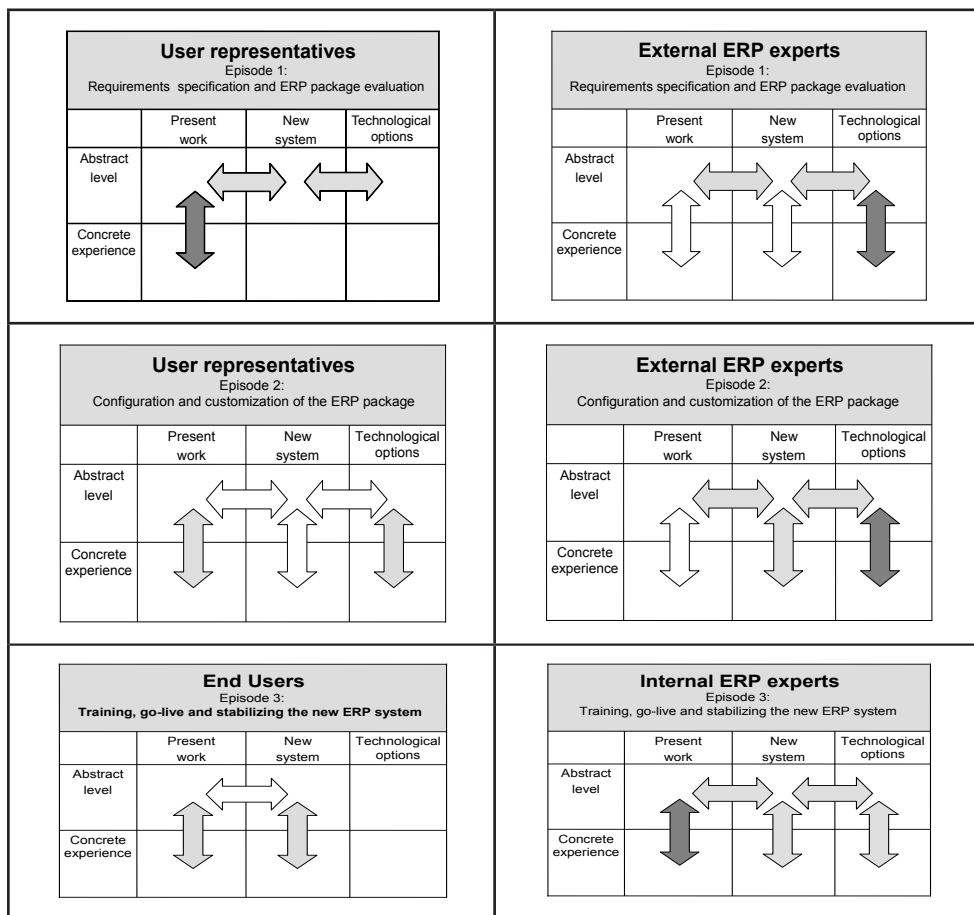
ERP implementation as design • 49

Figure 2A: Knowledge integration capabilities summarized for all five episodes

side). For budget reasons, a purchase (material need) had to be entered before the actual design and/or the vendor could be specified. To accommodate the needed coordination, both the user interface and the ERP system's logic were a serious challenge for Alfa. This was, however, not even realised during episode 2.
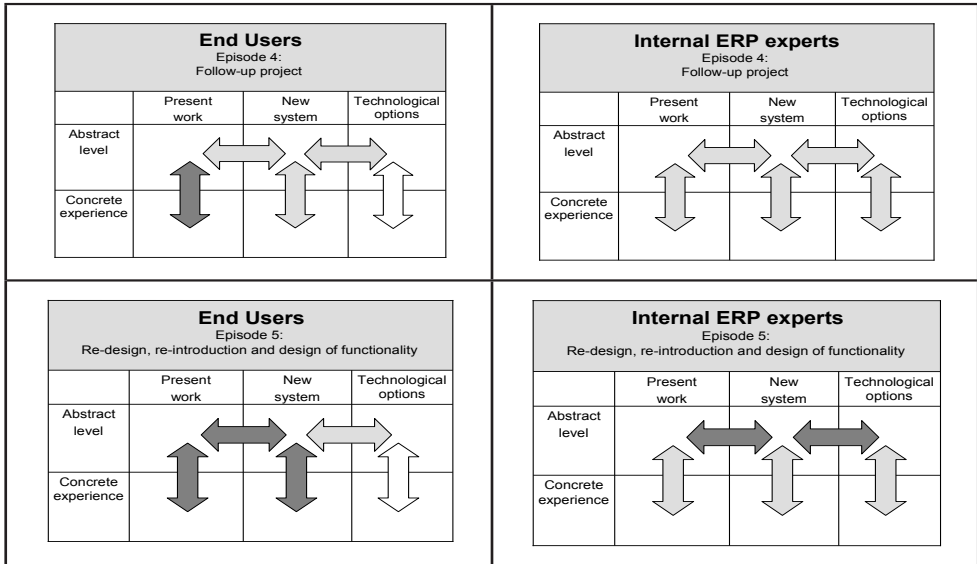
Figure 2B. Knowledge integration capabilities summarized for all five episodes

7 Discussion

Our analysis of the case study material indicates that an understanding of ERP implementation as socio-technical design provides a frame for a relevant analysis of the occurred difficulties. Moreover, it relates the observed difficulties to the lack of knowledge generation and knowledge integration considered necessary when understanding ERP implementations as design. Examples are: the lack of awareness of the influence of the scale of the projects on the usability of the project management module, the lack of understanding of the need for customization of the project budgeting functionality, and the above-mentioned purchase coordination. These problems turned up so late in the process that they caused delay and dismay for the project. In two of the examples we brought up, cross-functional knowledge integration on the vendor's as well as on the user's side was involved.

The remainder of the discussion addresses the second part of the research question: The exploration of possible remedies for the problems identified. The section focuses on three issues: (1) how to improve the knowledge integration earlier during the implementation, so that important mismatches show up in time to be handled better, (2) how to support cross-functional knowledge integration, and (3) the role of the internal ERP competence centre. We also indicate the need for future research.

Of course, the availability of methods does not guarantee their successful application. However, exploring tools and methods that have proved to support knowledge integration in the PD area might inspire and improve situations like the one observed.

ERP implementation as design • 51

7.1 Supporting early knowledge integration

Knowledge integration can be supported in many ways. Here, we discuss the following measures that should be combined to support early knowledge integration during ERP implementation processes: the organisation of cooperative design processes, shared representations to support them, and the use of prototypes, a specific design artefact, and incremental development.

Mediating Cooperative Design

In the ERP implementation at Alpha, several workshops that mediated a cooperative design took place. Especially the conference room pilot and configuration workshops addressed this dimension.

PD offers additional methods and tools: IT professionals can work for some time in the user organisation in order to understand the needs. Observation of existing practices might provide insight into needs as well, for example, taking part in planning activities around project management might have revealed the sheer size of the projects and might have triggered an adjustment of the implementation. Future workshops start with identifying concrete problems in today's work organisation as a base for developing visions for a changed practice. In design workshops, the cooperative design is mediated with the help of design artefacts—like mock-ups, rich pictures, or tangible representations of relevant aspects of the work practices—that provide a frame of reference to discuss and evaluate the future software and the changing work practices around it.

Organisational games as proposed in Bardram (1996) allow for an evaluation of the configuration in relation to realistic scenarios. Using them in an early stage in the configuration process, especially when designing and evaluating support for cross-functional coordination, might have revealed some of the problems discovered late in the implementation process.

Shared representations

When analysing the interviews, we recognised the lack of representations that could have formed a base for user representatives and external consultants to discuss the changes implied by the introduction of the ERP system. In the PD tradition, representations are discussed as a base for communication around design, not as a way to capture knowledge. Representations are understood as design artefacts that mediate communication between different professional groups. The late discovery of mismatches between the planned implementation and the specific requirements indicates that knowledge about the work practices, the technical possibilities and the specific implementation was not integrated well enough.

ERP systems present a serious challenge for the design process, as they already provide a relatively comprehensive body of functionality that constrains the design space. Reference models are a trial by ERP vendors to visualise the anticipated use. However, they do not allow discussing configuration alternatives as they—so far—only present a static version of the most standard configuration. Rosemann and van der Aalst (2007) propose the usage of configurable models for this. Research is needed to see whether this new kind of models and/or other representations can help to identify the design space provided of the standard system as well as its limitations.

For the representation of the existing work practices, business process modelling and the coordination pictures presented by Krabbel et al. (1996) are candidates. However, they have to

be used in a way that allows the consultants to understand the crucial aspects that the implementation has to take into account. They might help to support cooperative design workshops allowing domain experts to indicate potential mismatches. Research is needed to identify suitable representations and investigate how to make use of them in a fruitful way.

Prototypes and iterative design

Prototypes were used in configuration workshops and for the training that took place before going live. Nonetheless, important shortcomings were not discovered. In PD, the design and evaluation of prototypes is discussed in greater depth. For example, Blomberg et al. (1996) emphasize case-based prototypes based on real data allowing an evaluation based on realistic scenarios . Such prototypes could have revealed the problems like the project management module requiring three interaction steps per task and project member. Case-based prototypes can be combined with organisational games (Bardram 1996). That way problems can be identified before going live. This might be the only way for the users to develop enough understanding about the future ERP system in order to become serious co-designers.

7.2 Cross-functional knowledge integration

In the analysis, we could see that some of the main problems did relate to cross-functional, cross-module coordination. Both the project ledger and the numbering system for procurement items can be seen as coordination mechanisms (Schmidt and Simone 1996) between the project department, the accountancy, and the procurement department. The prototyping and configuration workshops mainly focussed on business processes within one department, neglecting the cross-departmental coordination.

PD does not have many methods that explicitly address the mediation between different professional groups in the user community. The above-mentioned combination of case-based prototypes and organisational games (Bardram 1996) is one of the approaches tried out. Krabbel and Wetzel (1998) propose an adaptation of rich pictures to both model cooperation, and envision the changes when introducing information systems supporting coordination. "View point pictures" allow communicating objectives between different professional groups within the organisation. However, little research has been reported in this area, and the usefulness of these methods in the context of ERP implementations needs to be established in further research.

7.3 The role of the ERP competence centre

Kensing and Munk-Madsen (1993) only describe the roles of two actors in the cooperative design process, the user and the IT expert. In a number of theses in the end of their article, they propose that users can only expect to acquire knowledge to relate their current practices to the future system. They leave the responsibility to relate all three knowledge domains to the IT experts. In our case, a third actor took over the overall knowledge integration and - in the end - was able to mediate between users and external IT experts. This was the internal ERP competence centre.

ERP implementation as design • 53

The development of similar roles has been observed in other contexts, where IT was used as an infrastructure requiring continuous development in order to support a developing work practice: Local designers take responsibility for the ongoing every day design-in-use as well as further evolution of the infrastructure (Dittrich et al. 2002; Kanstrup and Bertelsen 2006; Karasti et al. 2006)). The research on IS development as well as on PD might profit from paying more attention to the implication of the institutionalisation of practices that can be described as 'shop floor IT management' (Eriksén 1998).

7.4   Reflections on the use of the framework and limitations of the research

The framework used for the analysis has originally been developed by Kensing and Munk-Madsen (1993) in order to categorize techniques and tools used in relation to Information Systems development. In our research the framework was however used in a slightly different way; it was used: (1) as a lens to understand which knowledge categories need to be involved when implementing Information Systems, (2) as a way to identify who develops knowledge belonging to these different categories and when, and (3) as a way to analyze if and how the different knowledge categories are integrated in the case implementation.

Although we changed the usage of the framework, or maybe because we did so, it has been very helpful in identifying knowledge related issues. Especially combining the framework with a process perspective seemed to provide new insights. It highlighted that knowledge for design is developing over time, by different actors, and that more actors are involved than anticipated in the original framework. It made visible that the role of an actor in an ERP implementation can change over time. Finally, the framework helped to expose that no single actor covers all knowledge areas at the same time, and that it therefore is important to find ways to mediate knowledge integration within the heterogeneous group of actors involved in the ERP implementation.

Using the framework as an analytical tool also at times was very challenging. First, due to the fact that knowledge and knowledge integration is not material and therefore cannot be observed as such, we had to consider what could serve as indicators. Second, the distinction between concrete and abstract knowledge was difficult to work with, as the framework is not especially clear on this point.

Although we believe that the knowledge related issues identified in this research will apply to many ERP implementations, it is important to be aware that the research is based on a single case study, and that ERP implementations may differ. The case organization anticipated and allowed important customizations to the software, and furthermore whished for end user involvement in order to specify customizations and configure the ERP package software. Thus, end-users representatives were invited to influence the design; this may not be the situation in all ERP implementations.

8 Conclusion

In this article we propose and provide argumentation for a re-conceptualisation of ERP system implementation as socio-technical design. Taking this perspective allows analysis of the implementation process of our case study in a way that relates the problems observed in the implementation process to the (lack of) development and integration of knowledge relevant in socio-technical design processes. Further, this analysis allows us to discuss the concrete methods in the development process and propose the exploration of remedies in form of method and process improvements drawing on the PD discourse.

Our article thus contributes to understand and address well-known problems in ERP implementation. A further contribution is the underline of the importance of a developing role in organisational IT development: the internal ERP competence centre started to act as a local designer mediating between the needs of the organisation and the external IT experts.

The design processes around ERP system implementation pose different challenges than native development: How can the complex and powerful functionality implemented in the ERP system be related to the needs of a specific organisation in order to, on the one hand, make as much use of it as possible and, on the other hand, to recognise the specific points where the standard functionality has to be customized? A second aspect not thoroughly discussed in the PD literature is the design for (and with) a heterogeneous user group. Especially cross-functional coordination has to be designed carefully to support the business processes.

9 Acknowledgment

10 References

Bardram, J. E., (1996). Organizational prototyping: Adopting CSCW applications in organisations. *Scandinavian Journal of Information Systems*, (8:1): 69-88.

Blomberg, J., Suchman, L., and Trigg, R. H., (1996). Reflections on a work-oriented design project. *Human-Computer Interation*, (11:3): 237-265.

Bødker, K., Kensing, F., and Simonsen, J., (2004). *Participatory IT Design: Designing for Business and Workplace Realities*, MIT Press.

Dittrich, Y., Eriksén, S., and Hansson, C., (2002). PD in the wild: Evolving practices of design in use. In: *Participatory Design Conference*, Malmö, Sweden, pp. 124-134.

Eriksén, S., (1998). Knowing the Art of IT Management: An inquiry into work practices in one-stop shops. Department of Informatics, Blekinge Institute of Technology, Roneby, Sweden.

Esteves-Sousa, J., and Pastor-Collado, J., (2000). Towards the unification off critical success factors for ERP implementations. In: *10th Annual Business Information Technology conference*, Manchester, UK.

Fenema, P. C. v., Koppius, O. R., and Baalen, P. J. v., (2007). Implementing packaged enterprise software in multi-site firms: intensification of organizing and learning. *European Journal of Information Systems*, (16): 484-598.

Floyd, C., (ed.), (1992). *Software Development as Reality Construction*. Springer Verlag, Berlin.

Haines, M. N., and Goodhue, D. L., (2003). Implementation partner involvement and knowledge transfer in the context of ERP implementations. *International Journal of Human-Computer Interaction*, (16:1): 23-38.

Holland, C., Light, B., and Gibson, N., (1999). A critical success model for enterprise resource planning implementation. In: *7th European Conference on Information Systems*, Copenhagen Business School, Copenhagen, Denmark, pp. 273-287.

Huang, J., and Newell, S., (2003). Knowledge integration processes and dynamics within the context of cross-functional projects. *International Journal of Project Management*, (21): 167-176.

Huang, J. C., Newell, S., and Pan, S. L., (2001). The process of global knowledge integration: A case study of a multinational investment bank's Y2K program. *European Journal of Information Systems*, (10:3): 161-174.

Kanstrup, A. M., and Bertelsen, P., (2006). Participatory IT-support. In: *The Ninth Conference on Participatory Design: Expanding Boundaries in Design*, ACM, New York.

Karasti, H., Baker, K. S., and Halkola, E., (2006). Enriching the notion of data curation in Ee-science: Data managing and information infrastructuring in the long term ecological research," *Computer Supported Cooperative Work*, (15:4): 321-358.

Kawalek, P., and Wood-Harper, T., (2002). Finding of thorns: User participation in enterprise system implementation. *Advances in Information Systems*, (33:1): 13-22.

Kensing, F., (2000). Participatory design in a commercial context. In: *Participatory Design Conference 2000*, Palo Alto, CA, pp. 116-126.

Kensing, F., and Blomberg, J., (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work*, (7): 167-185.

Kensing, F., and Munk-Madsen, A., (1993). PD: Structure in the toolbox. *Communications of the ACM*, (36:4): 78-84.

Kim, Y., Lee, Z., and Gosain, S., (2005). Impediments to successful ERP implementation process. *Business Process Management Journal*, (11:2): 158-170.

Klein, H. K., and Myers, M. D., (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly, Special Issue on Intensive Research*, (23:1): 67-93.

Krabbel, A., and Wetzel, I., (1998). The customization process for organizational package information systems: a challenge for participatory design. In: *Participatory Design Conference (PDC '98)*, Seattle, Washington, USA, pp. 45-55.

Krabbel, A., Wetzel, I., and Radutsky, S., (1996). Participation of heterogeneous user groups: Providing an integrated hospital information system. In: *Participatory Design Conference (PDC '96)*, Cambridge, MA, pp. 241-249.

Markus, M. L., and Tanis, C., (2000). The enterprise system experience: From adoption to success. In: *Framing the domains of IT management: Projecting the future through the past,* R. W. Zmud and M. F. Price, (eds.), Pinnaflex Educational Resources, Cincinatti, USA, pp. 173-208.

Nah, F. F.-H., Zuckweiler, K. M., and Lau, J. L.-S., (2003). ERP implementation: Chief information officers' perceptions of critical success factors. *International Journal of Human-Computer Interaction*, (16:1): 5-22.

Orlikowski, W., and Iacono, C., (2001). Research commentary: Desperately seeking the "IT" in IT research: A call to theorizing the IT artifact. *Information Systems Research*, (12:2): 121-134.

Pan, S., Newell, S., Huang, J., and Cheung, A. W. K., (2001). Knowledge integration as a key problem in an ERP implementation. In: *Twenty-Second International Conference on Information Systems*, pp. 321-327.

Robey, D., Ross, J., and Boudreau, M., (2002). Learning to implement enterprise systems: An exploratory study of the dialectics of change. *Journal of Management Information Systems*, (19:1): 17-46.

Rosemann, M., and van der Aalst, W. M. P., (2007). A configurable reference modelling language. *Information Systems* (32:1): 1-23.

Schmidt, K., and Simone, C., (1996). Coordination mechanisms: toward a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work*, (5:2-3): 155-200.

Shanks, G., Parr, A., Hu, B., Corbitt, B., Thanasankit, T., and Seddon, P. B., (2000). Differences in critical success factors in ERP systems implementations in Australia and China: A cultural analysis. In: *8th European Conference on Information Systems*, Venna, Austria, pp. 437-544.

Stefanou, C. J., (1999). Supply chain management (SCM) and organizational key factors for successful implementation of enterprise resource planning (ERP) systems. In: *Americas Conference on Information Systems*, Milwaukee, USA, pp. 800-802.

Suchman, L., (1993). Foreword. In: *Participatory Design: Principles and Practices,* D. Schuler and A. Namioka, (eds.), pp. vii-ix.

Summer, M., (1999). Critical success factors in enterprise wide information management systems projects. In: *Americas Conference on Information Systems*, Milwaukee, USA, pp. 232-234.

Summer, M., (2003). Risk factors in enterprise-wide ERP projects. In: *Second-Wave Enterprise Resource Planning Systems,* G. Shanks, P. B. Seddon and L. P. Willcocks, (eds.), Cambridge University Press, Cambridge, UK, pp. 157-179.

ERP implementation as design • 57

58 • Pries-Heje & Dittrich