

10-8-2007

Exploring the Importance of Participation in the Post-Implementation Period of an ES Project: A Neglected Area

Erica L Wagner

Cornell University, elw32@cornell.edu

Sue Newell

Bentley College, USA, sue.newell@sussex.ac.uk

Follow this and additional works at: <https://aisel.aisnet.org/jais>

Recommended Citation

Wagner, Erica L and Newell, Sue (2007) "Exploring the Importance of Participation in the Post-Implementation Period of an ES Project: A Neglected Area," *Journal of the Association for Information Systems*, 8(10), .

DOI: 10.17705/1jais.00142

Available at: <https://aisel.aisnet.org/jais/vol8/iss10/32>

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Journal of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Journal of the Association for Information Systems

JAIS

Special Issue

Exploring the Importance of Participation in the Post-Implementation Period of an ES Project: A Neglected Area *

Erica L Wagner
School of Hotel Administration
Cornell University
elw32@cornell.edu

Sue Newell
Department of Management
Bentley College
snewell@bentley.edu
and
Warwick Business School, UK

Abstract:

User participation during software projects has long been considered a prerequisite for system success, and yet these initiatives continue to be rife with troubles. This is particularly true of enterprise software such as ERP and CRM, which, in spite of its popularity, is difficult to implement and is prone to user resistance. This, then, begs the question of why these enterprise systems run into problems even with when garnering user participation. One response may be to question the importance of participation per se; a more considered response is likely to be one that emphasizes the need to more closely explore the relationship between participation and the system in use. To this end, we adopt a cross-case comparison to analyze the role of user participation during two ES projects. Through the theoretical lens of 'situated learning', we argue that pre-implementation user participation can be problematic so that post-implementation involvement will be more effective in garnering user interest and assistance.

Keywords: ERP, Enterprise Systems, participatory design, user involvement, user participation, implementation, post implementation

** This is a part of the special issue on Enid Mumford's contribution to information systems theory and theoretical thinking. Jaana Porra and Rudy Hirschheim were the accepting guest editors.*

Volume 8, Issue 10, Article 1, pp. 508-524, October 2007

Introduction

The software development market today is dominated by off-the-shelf software 'solutions'. This is different from the past, when customized, in-house developments were the norm. This change has led some to argue that the systems development life cycle needs updating (Markus and Tanis, 2000) to reflect what is actually happening in practice. A key component of this shift will necessarily involve an examination of the role and timing of user participation efforts over the lifecycle of a packaged information system. For example, Sawyer (2001) evaluates the 'waterfall' or stage-model of traditional information systems (IS) development in the context of purchasing packaged software. He notes that end users are less likely to be involved in the early lifecycle phases when an organization purchases software because of the more significant costs involved and the need to work with third party intermediaries. Sawyer states, "It is only during installation that users become deeply involved for the first time in assessing how the software meets their needs" (ibid; 100). However, research to-date has not explored the implications of this change in the timing of user participation within the context of packaged software development and use.

We are not questioning the importance of user participation. Indeed, participation has been described as 'institutionalized practice' (Howcroft and Wilson, 2003) so ubiquitous is its emphasis within the IS literature, dating back to the early work of the Tavistock Institute, generally, and of Enid Mumford, specifically (the focus of this special issue). Thus, we accept Mumford's assumption that it is important to create IS that take account of people's social as well as technical needs and that user participation is necessary to achieve this (Mumford, 1983b). Our objective is to consider how this can be realized in the context of packaged software when users cannot actually be involved in software *design*, precisely because the design is relatively fixed by the package. Thus, historically early user participation was considered necessary in order to specify the system requirements; after all, the system was being specifically designed for a particular group of users. However, today's off-the-shelf packages come embedded with 'best practices, which vendors and consultants encourage to adopt rather than change to follow their existing practices. This raises questions about the importance and usefulness of user participation in the early stages of an IS project given that the system design is pre-specified and non-negotiable, at least if the adopting organization wants to follow vendor advice and implement the vanilla system.

In this paper we consider user participation in the context of Enterprise Systems (ES), particularly Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM). These packaged systems should be simpler to implement compared to tailor-made IS given that they merely need to be 'configured' rather than designed from scratch. However, research has identified that companies struggle with their ES implementations, facing user resistance and running significantly over budget or time-lines (e.g., Wagner and Newell, 2004; Sauer et al., 2001; Scott and Vessey, 2002), even when users participate in these projects in a variety of capacities (e.g., Parr and Shanks, 2000; Robey et al., 2002). This, then, begs the question of why these ES run into problems even with user participation? One response may be to question the importance of participation per se; a more considered response emphasizes the need to more closely explore the relationship between participation and IS success. Markus and Mao (2004) begin unpacking this relationship, highlighting the need for a more refined exploration of both 'participation' and 'IS success' and consideration of the particular context. We respond to this call for refining our understanding of participation by highlighting a dimension that appears to have been ignored in the past – the timing of participation in the context of large and complex ES implementations.

We address the following research questions: How do packaged solutions change the efficacy of user participation over the ES experience lifecycle? How does post-implementation user participation improve the acceptance of an ES? We do so by drawing upon the theoretical lens of situated learning and data from two qualitative case studies. Our work makes three contributions. First, it introduces the idea of post-implementation user participation as a normal and necessary part of systems development. Second, it documents the changing nature of software development and design and updates the discourse to reflect the dominant paradigm of purchasing off-the-shelf packages. Third, it highlights an inconsistency between a 'best practice' rhetoric as espoused by software vendors and the reality of post-implementation changes that occur in practice. Section one covers the relevant background literature. This is followed by the methodology section and the case descriptions. We then analyze and we draw conclusions in the final section.

Background Literature

In the 1980s and 1990s efforts were made in relation to participative approaches in large part due to problematic IS development methodologies that didn't accurately model the real world (Land, 1982) and relied upon the identification of

known requirements (Valusek and Fryback, 1985). This resulted in dissatisfied users who first experienced the IS at installation when it was seen to be too late to make changes (Avison and Fitzgerald, 1995). Research began to reveal how complex this IS development process is, leading to the questioning of some common assumptions such as: that users know precisely what their information needs are and can communicate these easily to system designers (Argyris, 1980); that relationships and communication issues between user and designer are straightforward (Argyris, 1980; Oliver and Langford, 1984); and that information needs are static (Land, 1982).

Not only was user participation a response to the "failure of conventional design but it was also based on a belief that users have a right to design their work environment" (Dearnley and Mayhew, 1983; 37). The work of Enid Mumford specifically emphasized the importance of participative system design. This emphasis has been accepted within the context of IS development (Howcroft and Wilson, 2003). For Mumford (1983c) participation is defined as "those democratic processes that enable employees to exercise control over their own work environments and work futures" (48). Her rationale was based on evidence that "treating people as adjuncts to machines does not work" (Mumford, 1983c; 47), leaving Mumford to conclude that involving users in design decisions will result in IS that take account of people's social as well as technical needs and so be more effective. This is set out in the principles of socio-technical design and incorporated in the steps of the ETHICS methodology (Mumford and Weir, 1979; Mumford, 1995), where system development is seen as inherently complex, requiring negotiations between different stakeholder groups. From this perspective the involvement of multiple groups in negotiations may be more work up-front but is central to system success so that requirements can be determined and accommodations made prior to implementation (Mumford, 1983a). This implies that users should be involved in ways that are substantial and active, rather than 'pseudo-participation' where user involvement is claimed, but IT professionals make the design decisions (Avison and Fitzgerald, 1995; p 90). The literature points out varying levels of participation (Dearnley and Mayhew, 1983; Avison and Fitzgerald, 1995; Mumford, 1983b) ranging from the *consultative*, where the user is interviewed at some point in the project, to the mid-range *representative* approach involving user spokespeople and analysts in the design process with both groups having a say in the decision making. The most participative approach involves all intended user beneficiaries throughout the design process making decisions based on a *consensus* model (Mumford, 1983b).

Ironically, the few studies that have focused on participation in ES project initiatives have found a degree of pseudo-participation where the role of the end-user is limited (Kawalek and Wood-Harper, 2002) and lacking influence (Howcroft and Light, 2006). Kawalek and Wood-Harper concluded that user participation was necessary for the project's success but had a particular "intelligence function" for the project manager, who needed to learn about the local environment in order to successfully implement the ES. Participation in this case was limited to determining the essential items to be configured into the standard package, versus those that could be handled with HR or process changes. Users did not help to shape the IS in any significant way because the solution had already been chosen. Similarly, Howcroft and Light (2006) found that the end-user role involved limited participation. They note that this is like going "full circle back to the early days of customized development when users had little involvement" (234). The authors mention "the myth of user involvement" (232) where lip-service is given to user involvement but where the actions don't actually support it. Specifically, end-users were broadly made aware of the project (prior to selection), but their involvement was not sought until three weeks before the planned implementation date.

Promoted for over 20 years, prototyping is traditionally viewed as "a system that works whereas a working system is a system that is used" (Dearnley and Mayhew, 1983; 37-8). Dearnley and Mayhew argue that the prototyping approach may be the only way that users discover exactly what they want from the system – and what is feasible. During analysis and design, the use of prototypes will help to create a feedback process between users and analysts. This technique encourages planned-for "loop backs between analysis and design that is used to gain as much information as possible from users" (40). The authors explicitly encourage early participation and suggest future research focus on the optimum time for creating the first prototype. Thus, prototypes are developed in an iterative fashion until users are satisfied and the production system is installed. Prototyping, however, may look rather different in the packaged software environment.

Prototyping draws our attention to how packaged IS development may be usefully seen as an iterative process— both formal and informal— where developers and users negotiate how the IS might evolve as users learn the system in the context of their daily ongoing activities. Using the IS in their everyday work alerts them to new ideas about how the IS can potentially improve practice, hence in our analysis we consider participation in both the pre- and post-implementation phases. In doing this we draw upon the situated learning perspective (Lave and Wenger, 1991; Wenger, 1998), and its close ally, ethnographic studies of work (see Suchman, 1987; Luff et al., 2000), which see knowledge and learning as emergent properties, rooted in the everyday interactions between people with the artifacts that occupy work spaces (such as IT).

Methodology

Given that existing literature does not consider this issue of participation timing, we analyze data from two case studies in order to identify how timing impacted the ES success in terms of exploiting the desired functionality. Our fieldwork comprised of interviews with key participants regarding their project experiences and systematic review of official project documentation.

Following Walsham (1993), the cases are designed to seek “validity...not [from] the representativeness of the case in a statistical sense, but on the plausibility and cogency of the logical reasoning used in describing the results and in drawing conclusions from them” (p 15). We used interpretive research techniques to gather, interpret, and analyze field data (Walsham, 1993; Klein and Myers, 1999). The paper’s findings are based on the comparative analysis of user participation within each case as well as a cross-case analysis. Each author separately conducted the fieldwork related to one of the two cases. In neither case was the initial research focus specifically on the timing of user participation; rather this issue emerged as the authors discussed insights from their respective cases. This subsequently led us to analyze each case in terms of this important, emergent issue.

One case focused on the process by which a workable ERP system was created within the administration of a prestigious U.S. university— ABC.¹ ABC provided a rich and appropriate research setting for two main reasons. First, the objective of the ERP implementation was to improve work practices for ABC’s user community. Second, ABC leaders and external consultants took seriously the participation of users throughout the project. The second case focused on the implementation of a CRM system across the global business of a large consulting firm, XYZ, which was trying to meet the needs of thousands of geographically and functionally dispersed users. Given the complex structuring of work, this case enabled us to compare the nature of user participation within a differently structured organization. In addition, those interviewed at XYZ had personal and professional experiences with many ES implementations. Not only were the interviewees able to reflect upon their experiences with the XYZ project, they did so in light of their consulting experiences on other ES projects. Furthermore, XYZ provided the ES methodology they were following, which explicitly includes a user participation emphasis.

Collecting multiple forms of data helped us to seek multiple interpretations to improve the “plausibility and cogency of our interpretive accounts” (Klein and Myers, 1999). Following Levina and Vaast (2005), we present a cross-case comparison in the form of a table that summarizes the research methods employed in the field (Table 1).

Table 1: Cross-case comparison of research methods

Methods	ABC	XYZ
Field work	4 visits each lasting 8 weeks over a period of 18 months	Interviews over 6 months
Timing	Implementation and post-implementation	Implementation and post-implementation
Narrative interviews	129 with 53 stakeholders. Recorded and verbatim transcriptions	15 in US, 1 in UK Recorded and verbatim transcriptions
Field journal	Pre and post interview notes and observations, transcribed	Notes taken during interview and site visits, transcribed
Documentation	Yes	Yes
Follow-up contact	Yes	No

Although the two cases were conducted separately, both researchers were informed by an interpretive epistemology, employed the same narrative interview technique (Bauer, 1996), and focused data collection on the same unit of analysis — specifically, an organizational implementation of an ES. Our field work resulted in different but complementary insights: one observed the individual psychology of users who were not fully engaged in a project until the post-implementation phase, while the other found that changing technical requirements leading to budgetary shifts meant that continuous user participation from requirements definition through go-live was not practical or realizable. Our analysis involved fully articulating the dynamics of each case study and then developing a shared perspective based on the themes that were present in either one or both settings. In doing this, we were identifying when participation appeared to be more or less successful in terms of supporting the objectives of the ES in each company. However, in terms of measuring ‘success’ we were assuming a subjective rather than an objective epistemology (Denzin and Lincoln, 1998): we did not aim to compare

¹ Company names have been anonymized as have the details of interviewees in order to protect their privacy.

the cases in terms of “success”, as a fixed dimension, since we assumed that what counted as success was socially constructed by those involved.

To accomplish this each researcher individually reviewed her qualitative data and constructed a case description highlighting issues of user participation that represented our independent interpretations of the data. We shared these write ups, along with a list of emergent themes, in order to compare the cases and consider the principles of dialogical reasoning and suspicion in order to further improve our interpretive accounts (Klein and Myers, 1999). An iterative process of cross-case discussion and fine-tuning of the case descriptions helped us to identify an over-arching theme related to the timing of user participation. We then organized the analytical themes around this issue, which allowed us to develop a coherent sense of the participation across the two cases. After further developing the analytical focus, we chose the theoretical lens of situated learning as being most effective for illuminating the issues that emerged from the analysis of case data.

Case Descriptions

Each case is organized into two episodes based on key issues related to user participation. We focus in both cases on user participation once the IS package was selected, not on the selection phase, which does not typically include end user involvement in the context of ES (Sawyer, 2001).

XYZ Case

XYZ is a global organization manufacturing and retailing both PC and high-end computer systems. It also has global consultancy businesses. In 1999, senior executives decided to implement an external CRM package because of its scalability and ability to support the breadth of XYZ, which had over 50,000 users. The project commenced in 2000, and by 2004 several of the modules had been implemented but not across all regions, and there had been significant setbacks with many of the modules, especially sales and marketing. Moreover, many of the legacy systems that should have been ‘sunset’ once a module went live were still in operation, being used alongside the new CRM. The core team was described as “burnt out.”

Participation during Pre-Implementation

Each CRM module was designed and implemented by a separate project team working with sub-teams at the local geographies for implementation at each site. There was a central project leadership team overseeing the whole initiative. A well-defined project methodology was made explicit in a 70-page manual that included considerable emphasis on user participation:

The user represents the field or end customer. Throughout the process the user ensures that the customer’s requirements do not get lost among the infrastructure, coding, and financial decisions. The user validates the deliverables and benefits, and is active in every phase.

Given the perceived importance of user participation, the project methodology specified that each CRM module had a project development lead whose job was to focus on the user. The way that this participation was managed was to bring user representatives together to actually look at the functionality of the CRM system and compare it to existing practice and to challenge why this practice could not be changed. For example, with the first call center module:

The very first one we did, we literally brought 50 people in from around the world, who were real users, and sat them in a room in [location] for two weeks. We said to them here’s how your process works, and here’s how people might handle that process. Tell me how it works and tell me why not, and we did a whole lot of mapping of the processes...the question we always asked was ‘why can’t you use this?’ Don’t tell me you don’t like; why can’t you use it? [Core project member E]

In this instance the users identified over 600 modifications that they wanted, but only eight of these were actually enabled in the first release, and these related to terminology and nomenclature, since they “elected to change the people rather than the tool.”

Users involved in the project struggled because they could not fully understand how the new practice would work while it was still at the concept stage. It was not until the users could get a firm idea of the new system that they began to take notice and show concern about the changes that it might force:

In terms of, ‘now this is what it’s going to look like’...[users] start thinking through a little bit more carefully in terms of ‘this is what I’m actually going to get versus what I have today.’ Then they start to worry about whether or not what they’re getting is in fact necessary to them or not. And they do more reflection, more detail, and more survival to make sure that they’re not getting anything less than what they have today. [Core project member A]

Even when users had been involved in the configuration/customization phase, it did not necessarily translate into a system they found useful:

Roughly one year ago we did a walk-through of the application with the same audience after the requirements had gone in from the fit-gap and the developers had actually developed the tool. This was enlightening for them – in some ways it worked better and in some ways worse. And we also got some additional requirements out of this process. [Core project member C]

In the case of this first module, the project implementation team was able to ignore the customization requests of the user representatives and force the system on the users. They were not able to do this with later modules. This was partly because users involved in the later modules were not present at the outset, meaning that they missed out on the momentum that built around the initiative and failed to hear much rhetoric about the project mission. As one senior member of the core project team said:

If I were to run a project like this, the lessons I have learned about people on the project is we need to be re-educating them on how you want the project to work; otherwise, they will reinvent it.

Thus, users in sales and marketing resisted the vanilla model. These users were more powerful and, while some of their resistance was recognized as warranted because of the need for local customizations to support differentiating processes, some of it was considered unnecessary:

The sales team basically stuck their tongue out and said screw you; we ain't going to do this unless you do it our way, which has led to a number of compromises [customizations] in how we actually implemented the package, some of which are good, some of which aren't good. [Core project member A]

Both the marketing and sales modules were stalled because they could not get agreement on what customizations were essential. In this period they spent time discussing these issues, but trying to accommodate all the needs of different user groups simply led to a stalemate. On the marketing module, the problem eventually led to the appointment of a new project manager. He believed that trying to accommodate all the diverse needs was impossible:

Piloting and working forward I think are the absolute key ingredients because otherwise you can get a committee working and discussing it for years and they'll come up with something they think is absolutely perfect and it will fall apart within two weeks of going live because there's so much stuff they didn't know, or the world moved on, or you know it wasn't supposed to be like that, you told me this field was unique, it's not. I thought it was. And suddenly you have all these other issues. So I think getting on and doing it is absolutely important.

He saw this as inevitable because of the way in which people learn to use their information systems over time in ways that were not originally envisioned or intended:

So you end up with a huge amount of different data formats and different legacy databases... the people are doing the best they can and also using the fields for something in which it was never intended, either because it was never really closely defined or because tactically they had to do something so they did that.

As such, he took a different tactic to move the stalled project forward. He explicitly attempted to make the new system look as similar to the old as possible in order to gain user support. This meant building things into the new system that were not strictly necessary,

And so they are bringing with them the legacy thinking. And some of that legacy thinking you're just going to have to accept even though it's wrong. Customer numbers being an example. Even though it's wrong. You shouldn't do it [try and attempt to remove them in the new system]. Basically it's not going to get in the way long term so let them keep [it] and then you can just move [the project] along. Otherwise you can just get bogged down in some really big debates and a lot of teams do that.

Participation during Post-Implementation

With this revised approach to configuration/customization, a new marketing module was eventually rolled out. By explicitly focusing on accommodating users' traditional practices, the power of the CRM was not fully exploited at installation. However, over time, and with practice users learned to utilize what they had been given, sharing this knowledge across their group:

People are very good at exploiting; that's people's nature when they discover that, for example, there's a button there and they press it and it actually does something which they couldn't do before, and they say, 'You know this button works'. And you go, 'Yes, yes it does work. It's been there all along and you've

now discovered something'...And you don't have to educate people. It just spreads like wildfire. [Marketing module project manager]

Through this experiential learning, users also came to understand what could and could not be done with the software, leading them to make requests for customizations that would improve their practice:

Sometimes people ask for a particular feature... 'if we had this, we could do a lot'. And they wait for it...and eventually six months later they finally get it and they test it and...zoom straight afterward it makes the [work] change. So I think the deploying of the enterprise system really is the enabler to massive change...What happens is you go through this cyclic thing.... [Marketing module project manager]

This exploitation process in the post-implementation environment, however, was not done systematically – the methodology did not cover this phase but rather assumed that the process was ended once the IS was installed. Thus, there was no formal process to engage users systematically in this period; the learning through practice was ad-hoc. This led to some of the project team stressing the importance of getting the configuration/customization 'right' initially:

You know I compare it to having a house built...no matter how much planning you do and all the right intentions you go in with a limited dollar amount and a limited amount of patience. There are many things that you say I'm going to go back after I build a house because either I can't afford it or it's not the right time or I haven't made a decision properly...most of the time you don't go back and you don't make those changes...Very, very similar with SAP implementations. Which is why we try as much as possible to stress the importance of 'you need to do it now'. [Core project member B].

ABC Case

ABC is a prestigious U.S. university comprising an undergraduate college, graduate and professional schools. The university as a whole is administered centrally by a large staff housed in functional units. These staff gather information from academic departments through various legacy systems and then reconcile the results in order to report data on a quarterly basis. In 1996, the university leadership chose to develop and implement an ERP package in partnership with a multinational software vendor in order to integrate their loosely coupled structure by reducing the administrative autonomy that existed in the departments. The project commenced in late 1996 and by summer 2000 the entire suite had been implemented across the university; however, the academic constituencies were refusing to work with the financial management module.

Participation during Pre-Implementation

A great deal of care was taken to select a representative project team. Guided by the ERP vendor's methodology and with expert guidance from a multinational management consulting firm, a matrix project structure was employed so that each ERP module was associated with both a functional and a technical team. Members of these teams were staffed largely with ABC central administrative management, whose positions were to varying degrees back-filled for the duration of the initiative. Vendor representatives, software contractors, and management consultants were also assigned to the larger project team. User participation was sought from diverse populations on campus and representatives from departmental units were involved including several 'super-users', line level employees, and low- to middle-management. Yet in spite of the project team's attempt to involve faculty, they failed to show interest in the administrative IT project.

Traditionally, there had been a lack of mutual understanding between academic departments and those working in central administration. Central staff complained that, while their work was intended to help the University as a whole, the academic constituencies did not value rigorous administrative practices. Faculty argued that administration charged them close to 30 percent in indirect costs for every research dollar they brought into the organization, and for that price Central should take the brunt of the administrative effort. With the inception of the ERP initiative, the newly formed project team was aware that academic constituencies would be affected because the design of the software would require faculty and their staff to work differently. In particular, the ERP supported an integrated financial management and budgeting approach that would assist in rolling-up departmental information into a University-wide corporate budget. This approach—time-phased budgeting—was expected to professionalize accounting and budgeting by moving the academic enterprise out of its ad-hoc legacy 'commitment accounting' environment. The project team tried to garner faculty support from the beginning of the project and worked hard to keep them informed of the changes that were being made to administrative practice:

...Particularly working with the faculty and trying to figure out how to keep them up to speed and interested in what we were doing, was decidedly less successful. That was, super challenging—maybe it was us—I mean maybe we just didn't focus enough time and attention in trying to figure out exactly what they would want and need and how to keep them excited throughout the process. I'm not sure. But we did have a couple of attempts at committees, to bring people on committees and it really...was not...a



very successful process...they weren't interested during the project but they certainly are interested now. [ABC Leader]

Members of the project team, on the other hand, were becoming deeply invested in configuring the ERP in a manner that was commensurate with their centralized approach to administration.

Those super-users from departments whose perspectives were at odds with the centralized approach found themselves swayed by the conceptual ideas that underpinned the proposed tighter coupling of practices:

Everyone was talking about how [the vendor] was going to allow us to be more professional in terms of finances and the concepts sounded great and made sense to me [Super-user project member]

The project team felt as though everyone was in agreement with the proposed configuration because no one stated otherwise:

We are doing great, everyone is on-board with the proposed changes. I mean, they know it's going to be a lot of work in the beginning but the community is on the same page. [Change manager]

The change manager held weekly sessions with the user community in an attempt to elicit feedback and communicate project news.

In addition, ABC's VP for Administration, who was the project's champion, advocated user participation throughout the duration of the initiative in order to improve the likelihood of user acceptance at installation. He ensured adequate funding for back filling positions and encouraged input from the wider community. These efforts were only partially successful:

...Most people were not particularly interested in the details of the implementation and that's what you end up spending most of your time and effort worrying about...I mean, some people were very nice and very co-operative and came to all the meetings and sat there and would occasionally ask questions and tried to act interested. But I really don't think, even the few people who we had involved in committee efforts and the project team part time, my sense was that they were not really heavily engaged in the process...as long as they got the system they wanted at the end, that's all they much cared about it. [ABC Leader]

As a result, the project team configured and implemented the modules to reflect their preferred central administrative perspective to financial management.

Participation during Post-Implementation

At installation, however, the wider ABC community refused to work with the financial management module. The unfamiliarity with the ERP technology itself and the necessary learning curve that is a part of all post-implementation phases of development only partially explained user resistance. The user community, who had expected a 'reduction in hassle factor' as a result of the new system, felt that their needs had not been met.

Why has central management become invested in changing, for an extremely large revenue stream, from a tried-and-true system, proven to be both efficient and well-accepted by the faculty, to one that has not been tested with these types of accounts, and is less efficient and useful for the faculty? My guess is that there is too much water under the bridge to go back. [Academic user]

During the post implementation phase faculty formed a coalition and demanded a series of meetings with the provost, where, along with their staff and key project leaders, they expressed their feeling that the project had failed to meet the needs of the academic community. The provost mandated the project team to work with the academic constituencies on modifying the ERP in order to stave off their rejection of the system. This, coupled with academic interest in the post implementation phase, meant that rather than winding down the project infrastructure, it remained fully ramped up and staffed for the first year.

The project team negotiated with faculty to find a workable solution for everyone that involved making sufficient modifications to the ERP and the accompanying business processes. One project member who was seen to have strong skills communicating between central administration and the academic constituencies was assigned to work closely with faculty one-on-one in the post implementation environment:

[Faculty] are receptive to the new reports which have more of the commitment accounting concepts that they are familiar with...The faculty are never going to use time-phased budgeting, why are we working so hard to change the culture?...[A leader on the finance team] says we need to either change the culture to fit the approach or change the approach to fit the culture and now she thinks that maybe we shouldn't really work to change the culture. One way or the other we need to get faculty on board.

Interestingly, post-implementation interviews with several project leaders indicated that they believed these negotiations with faculty existed only because the project team was inexperienced and had somehow failed to do it right:

I'm devastated, we tried so hard and everyone was all smiles and going about their business and now they are out with the daggers, it's as if the past three years of my life are being challenged and I'm told that I'm a failure because the departments all of a sudden hate what they seemed to be ok with before. [Project Team Member]

They argued that if they were to embark on another ERP project user resistance would not happen:

[Project leader] says she feels as though they struggled so much because it was new to them and that the upgrade to the [next version of vendor's product] will go a lot more smoothly because of the critical success factors they learned about from the first implementation. She felt strongly that they had learned a lot and would get users to be more readily involved in the future so that they would avoid post-implementation problems. [Field researcher's pre-interview notes]

Furthermore, ABC lost a significant amount of project expertise during the post-implementation phase. Several university leaders, including the VP, left within the first year after installation, and project members were hired away by the external groups who were consulting on the project. Yet despite these human resource changes, the project team succeeded in creating customizations to the ERP and modifying business processes enough to bring the academic enterprise into the ERP environment.

Analysis

While we recognize the benefits of user participation as advocated by Enid Mumford, we argue that the increasing popularity of buying rather than building systems today requires an elaboration of the concept, as we attempt in this analysis. Our main point is to stress the reality of ES implementation projects as inevitably highly iterative, with organizations (and groups) gradually learning to exploit the functionality of this powerful type of integrative software package through situated learning in use. During cycles of configuration/customization and implementation/use, organizations are gradually able to exploit ES functionality. Companies should not see the first cycle as the end-point because, during post-implementation use, individuals are much more likely to be able to learn how to subsequently configure/customize to maximize future use. Our analysis indicates that user participation is as important, if not more, in the post-implementation period as in preceding periods, a point that seems to be ignored in much of the literature on user participation as well as the resource allocation model used by organizations in practice, both of which stress the importance of user participation during the pre- rather than post-implementation phase.

Limits to User Participation during the Initial Phase

Our case data indicate that there are limits to user participation during the initial configuration/customization phase.

Legacy thinking

Evidence from both cases suggests that users found it difficult during the early stages of requirements definition to see beyond their current practices and anticipate how things could be done differently if they had new tools to enable more integration within and across functions. This is because much work practice is rooted in everyday interactions (Suchman, 1987) so that trying to appreciate a new way of working by just looking at the technical system is difficult. Add this to the users' limited technical knowledge (Beath and Orlikowski, 1994) and it becomes understandable why, when users are involved in the configuration/customization phase, their main concern is that the new system will enable them to do what they did before— with 'as little change as possible'. Arguably, given such legacy thinking, user involvement during the design stage can actually restrict the potential for the new system to be used in transforming ways. In the XYZ case, this legacy thinking significantly delayed the implementation of the marketing and sales modules because the user representatives would not compromise on their existing practices. Here, these users had significant power in the organization and so were able to stall the project; a situation that many user groups will not enjoy (Howcroft and Wilson, 2003).

This problem is not traditionally experienced in software development because bespoke software is being developed specifically to support organizational processes (even if these have been previously reengineered). This highlights a key difference between the traditional and packaged IS development lifecycle and the implications for user participation.

Vanilla Implementation

Since the option to purchase software was first recognized in the literature so too was the question raised as to whether one should implement an unmodified (vanilla) package, or enhance and customize the purchased package (Rands, 1992).

While the new generation ES packages do allow configuration options, the general advice from software companies and consultants is to avoid, as far as possible, customizations to the package, and so go for a vanilla implementation. This was certainly the strategy followed by XYZ, where the user participation was consultative (Mumford, 1983b) in nature and met the intelligence function of project management, who sought to learn about the local environment and minimize changes to the software (Kawalek and Wood-Harper, 2002). At ABC, they were combining modules from existing ES products to create a university-friendly suite. To this end, ABC did employ consultative and representative participation, seeking to gather intelligence regarding higher education work practices from University representatives in order to determine the essential modifications to their existing, public sector software package. However, the vendor made it clear that the relatively small market size of the university sector meant that a return on investment required as few alterations to their existing product as possible.

Given this context, one wonders what the point of user involvement is if users are not listened to because of the desire for a vanilla implementation. For example, in the XYZ case, the call center users requested 600 customizations to the standard package but were only allowed eight of these in the first release of the software. One can anticipate that involving users but then ignoring their requests is unlikely to foster trust and confidence, ingredients recognized as essential for effective participation by the situated learning school (Wenger, 1998). Instead, it may actually create frustration and resentment, thus eliminating one of the primary goals of participation, to reduce user resistance (Wong and Tate, 1994). In the instances where vanilla implementation is a project goal, user participation beyond the consultative level during the configuration/customization phase might in some circumstances be counter-productive. Of course, such a problem would not surface during traditional IS development.

Motivation

A final problem observed in the cases during this pre-implementation phase was actually getting users to be interested in participation throughout the initiatives. This was particularly the case in ABC, where representative participation existed through the involvement of super-users whose primary positions were back-filled. Notwithstanding, faculty and the wider user community did not take much notice of the ES being developed, despite the fact that they were regularly invited to attend update meetings. Furthermore, super-users were unable to recognize how the ES financial module's configuration would adversely affect academic constituencies, a fact that would be anticipated from a situated learning perspective (Lave and Wenger, 1991). Similarly, in the XYZ case, the core team commented on how users were not really interested until they could actually see a working version of the new system and then really appreciate how it was going to affect their work practice.

We argue that, rather than being a shortcoming of the participative approach, the lack of engagement is a reflection of human nature, as depicted by the situated learning school, where we only become interested in something when it is salient to us and when we can actually begin to learn about the technology through practice and participation (Wenger, 1998). Even genuine attempts made by the users to be involved will fall short because of more pressing matters that are in front of them at the time (Wagner and Piccoli, forthcoming). As our case evidence indicates, even when the precept of early user participation is followed, as was the case at both ABC and XYZ, users who are busy at work will have their attention captured by immediate responsibilities and the project will remain low in salience, failing to fully engage user attention. This was the case with ABC faculty who became committed to the ES project only when their sphere of work was directly impacted. This is in part due to legacy thinking (see above); it takes significant cognitive effort to envision how the final system will operate and change work practice. In addition, a temporal element needs to be considered, involving examining how closely the information given impacts daily life. According to elaboration likelihood (Petty and Cacioppo, 1986), individuals have to be both motivated and able to process information in order for it to become salient and move them into action. Thus, when project completion is imminent and the reality of new work practices becomes apparent, users begin to evaluate the new system more closely and raise significant issues, often leading to user resistance and the need for post-implementation modifications.

Importantly, project staff at ABC interpreted the lack of engagement and dissent as indicative of user support. The change manager felt that the team was "doing great". We come to see later that users were in fact not on-board and were merely acting "as-if" they trusted the project team to deliver an appropriate system because of the project's lack of salience for them. According to Wynne (1996) "as-if" trust exists in risky environments because there seems to be no other plausible choice.

On the other hand, in XYZ, the problem was that call-center users were motivated and suggested changes to the standard configuration, but these were mostly ignored because of the desire to install a vanilla system. Both the as-if trust in ABC and the ignoring of users in XYZ can be seen as examples of pseudo participation (Avison and Fitzgerald, 1995). The latter can be referred to as **top-down pseudo participation**, where managers attempt to involve users while controlling all the decisions

because of their power (Howcroft and Wilson, 2003). However, our findings suggest that users may also be described as showing only pseudo participation, albeit for different reasons. We can refer to this as **bottom-up pseudo participation**. Interestingly, the former type of pseudo participation was more evident in XYZ, while the latter was more pronounced in ABC. This is probably a reflection of the difference between organizational forms where the loosely coupled university environment retains distributed power and autonomy with faculty who do not feel compelled to participate until their domain and work practice is directly impacted. Conversely, the structure of global corporations tends to require more traditional reporting and communication structures that enable management directives. Moreover, the bottom-up pseudo participation in ABC meant that resistance did not occur until the post-implementation phase, while in XYZ the top-down pseudo participation led to resistance during the configuration/customization phase among those groups that had the power to resist. For this reason, the project implementation was stalled in certain modules at XYZ until customizations were made that users demanded. In ABC, the implementation was not stalled, but users refused to use the implemented system until customizations had been made. In both cases, therefore, this pseudo participation led to project delays.

Opportunities for User Participation in the Post-Implementation Environment

Given these difficulties of user participation in this initial configuration/customization phase, we suggest that organizations should consider carefully how users may be encouraged to participate in the post-implementation environment. If we take the notions of salience during situated learning as our point of departure with regard to the engagement of human resources during ES projects, then the framework for effective user participation changes. Take, for example, a key issue that was apparent from the two cases the iterative nature of an ES implementation. Modifications occurred in the post-implementation period, when users finally became engaged in making the software work for them in the context of their everyday situated learning from practice. Thus, in ABC, it was during the post-implementation phase of the ES project that previously silent users expressed their dissatisfaction with the system. This is demonstrated most effectively with the example of faculty concerns about the financial management of their grant accounts. At this point, faculty and their staff became angry with the project team's choice to exclude legacy functionality in a manner that they interpreted as underhanded. While evidence does suggest that the project team was motivated to shift the ways in which faculty and their staff report on grant dollars, the quotations illustrate the emergent and situated nature of engagement where faculty and their staff became concerned about the ES design when previously they had not been engaged. Similarly, quotations from the XYZ case **emphasize how**, when users actually started to work with the system, they came to appreciate the advantages and the limitations of the system as provided and so demanded changes and/or learned to exploit its functionality as they participated in and learned from their everyday routines.

This leads us to question high levels of continuous user participation throughout the duration of the initiative, especially when consensus is so difficult to achieve, given the political nature of organizations (Howcroft and Wilson, 2003). Furthermore, the 'whole house' scope of ES implementations means that consensus participation (Mumford, 1983b) would require an unwieldy amount of user participation during pre-implementation. Given the integrated nature of an ES, involving users from all the different areas is likely to lead to a situation where agreement is difficult, if not impossible, to achieve. Different groups will want different things included/excluded and if consensus is sought, this is likely to lead to a stalemate (Wagner and Newell, 2006), as in the marketing and sales deployments at XYZ. Or, it may lead to the development of a configuration that is initially seen to be perfect and yet falls apart in use. The issue here is that users (and designers) are attempting to create an IS that suits current demands, but ignore the fact that demands change rapidly today and/or that once a system is in use, users will in any case adapt/modify and develop work arounds so that, through a process of situated learning, any system will change from its intended use pattern (Orlikowski, 1996; Suchman, 1987). Therefore, it may be more advantageous both in terms of resource time and eventual buy-in to find out how the basic system is used before deciding what customizations are going to add real value, beyond those that are initially agreed to be essential to accommodate unique organizational nomenclature.

Prototyping and iterative approaches to software development were developed for bespoke software design. We now can ask what prototyping means in the context of packaged software implementation and whether the distinction between a prototype and a working system (Dearnley and Mayhew, 1983) continues to be helpful. Quite relevant for today's packaged software environment is Dearnley and Mayhew's statement that the prototype should be "simple and relatively quick to create, amend and rebuild" (41), reminding us of the role for vanilla ES implementations, with the vanilla version being seen as a prototype that might subsequently require modification. Further, it has been pointed out that prototyping might be more than just the basis for the analysis/requirements phase of IS development: it might evolve into the operational system itself. From this perspective, the prototype is a system 'in use' (Avison and Fitzgerald, 1995) that will continue to evolve as the organization does. This is in keeping with the idea of growth and emergence as opposed to design (Truex, Baskerville and Klein, 1999), where user acceptance will never be complete because organizations are constantly changing. As such, IS development needs to be seen as a continuous process of evolution with no final design being possible or warranted. User participation from this perspective may not be constrained to the pre-implementation phase but may also

be considered in the context of use, both in terms of informal learning and improvisation (Orlowski, 1996) but also in terms of formal processes of continuous and iterative IS development. Consultancy providers have recognized the potential of such post implementation adaptation as evidenced by the substantial and growing consultancy business focused on this. The effectiveness of such services, however, has not received research attention, and while our research highlights why such services are necessary, further research is needed to identify whether and how such consultancy provision can be effective.

At the other end of the iterative life-cycle of development, Dearnley and Mayhew (1985) posed a call for research investigating the optimal time to build the first prototype and our study seeks to contribute to this question. During the cycles of configuration/customization and use, the reality of the new system certainly becomes salient and users will inevitably become more interested as they begin to learn from their situated practice. Indeed, we would argue that it is at this point that users might move from being involved to engaged in the ES implementation project because once their work practice is being impacted, prototyping will be most effective. Most importantly, it is during these iterations that they will really look at what the new system offers and be concerned about whether it makes their job easier or more difficult. Where the new system is seen to make their job difficult, as in the ABC case, there will be significant user resistance. Attempting to get users to appreciate and use all the potential functionality of the new system appears to be a mistaken or unrealistic target. Rather, a more realistic goal appears to be to provide users with a system that makes their jobs not significantly more difficult and at the same time provides them with the prospect that they will be able to do even more in the future. This incremental or piecemeal approach to ES implementation (Robey et al., 2002) has been criticized because, in the post-implementation environment, resources are not typically provided to source the effort that is required to continue the development process. However, we argue that this is a matter of planning for resources to be available in the post-implementation environment, rather than a fundamental problem with the piecemeal approach. Indeed, we would argue, based on our cases, that such an approach is likely to be more realistic and more successful, provided that these resources can be made available. Thus, our two themes for post-implementation user participation are that organizations can think selectively about when to engage users and that they can set aside resources for this phase of the ES lifecycle, not expecting the configuration/customization phase to be a one-off, initial phase.

Selective Engagement

We argue that in the post implementation environment, one is better equipped to engage users effectively. This does not necessarily imply that users should **not be engaged in consultative participation during** the initial configuration/customization phase of the implementation since this is in any case likely to be both politically important and necessary for the intelligence function of the project team (Kawalek and Wood-Harper, 2002). What it does mean is that there are limits to user participation in the early stages of an ES implementation, as discussed above, while once they have to actually use the system and begin to learn about its advantages and limitations from their situated practice, they are much more likely to want and be able to have their voices heard. At first consideration, thinking about involving users after installation of the software seems unrealistic and expensive, but we introduce the notion of **selective engagement** as an additional form of participation that would augment the existing levels of participation (Dearnley and Mayhew, 1983; Avison and Fitzgerald, 1995; Mumford, 1983b). We argue that engaging users selectively in the post-implementation use phase is likely to be particularly important and productive in terms of creating a viable IS.

Others have shown that analysts don't do a good job of hearing users during requirements definition (Alvarez and Urla, 2002). This is in part due to the storytelling approach users adopt when talking about their requirements, where they share stories about their current activities and point out limitations, challenges, and highlights of their business practices. This is a reflection of the situated embeddedness of everyday work practices (Lave and Wenger, 1991; Wenger, 1998). However, analysts discount this information as superfluous and become frustrated when users cannot list their requirements in a more linear, rule-based manner. Analysts fail to recognize the extent to which rules of action do not proscribe practice, since each episode of action is unique and in some way dependent on the meaning that actors determine, rather than on pre-defined causal connections (Garfinkel, 1986).

Despite the wisdom of conceptual modeling such as soft systems methodology (Checkland, 1981) as an aid in information requirements determination (c.f. Galliers, 1987), many analysts still discount these stories during pre-implementation. We argue that it is easier to hear the users when there are tangible products/processes/functions to refer back to and users are working in the new environment, as in the ABC case when the problems associated with the lack of legacy accounting functionality became clearly communicated. Once faculty were actually working with the new ES they could better describe how it was limiting (or enhancing) the ways in which they conducted their activities. If users have migrated to the ES environment, they will likely tell stories about their experiences in that environment and highlight what they feel is lacking both in terms of what they used to have and new functionality that might be incomplete. This is not to argue that requirements definition should be avoided during the initial configuration/customization phase. Indeed, this is necessary, both in terms of facilitating users to move away from legacy thinking and the political necessities associated with system

ownership. Rather, this is to say that sticking more closely to the vanilla implementation may be preferable—as long as key actors are willing to engage selectively in the post implementation engagement. Doing so will highlight that which is non-negotiable from the users' perspective and that which is not.

Determining non-negotiable requests is much easier in the post-implementation environment when users will highlight that which is most likely to lead to a rejection of the system, as in the ABC case. The power of an implemented system is not to be underestimated in terms of persuading users to make it work. As has been indicated elsewhere (Wagner and Newell, 2006), attempts to achieve consensus about development on an integrated software project can consume considerable resources without knowing for sure which issues are necessary for success and which are negotiable. This leads us to the theme of resource allocation.

Resource Allocation

User acceptance is seen to be the number one critical success factor for software projects, and user participation can help to achieve this (Davis and Olsen, 1985; Mumford and Weir, 1979; Mumford, 1983b, 1995, 2000). Moreover, user participation continues to be seen as critical in relation to ES implementation success (Holland and Light, 1999; Markus et al., 2000; Nah et al., 2001; Nah et al., 2003; Parr et al., 1999; Rosario, 2000; Sumner, 2000). However, with the high rate of ES failure, more must be done. While considerable resources are being spent, pitfalls continue to be encountered. Engagement emerges over time as users learn from their situated practice, and the needs of different groups must be addressed *at that time*—even where a concerted effort has been made to encourage user participation from the outset. In XYZ, marketing users demanded changes prior to implementation, stalling the project, but in all modules staff made customization requests in the post-implementation environment once they recognized the limits of the system. In ABC, demands from faculty in the post-implementation phase were particularly forceful, upsetting interpersonal relationships, increasing the project budget and delaying acceptance of the ES. Project leaders were resentful of faculty and accused them of an 11th hour interest in the project. Faculty and their staff were shocked, on the other hand, to find that the software did not meet their needs, and trust was damaged. As discussed above, we argue that 'as if' trust is a regular feature of project work and silence should not be mistaken for compliance or relinquishment of ownership by lay persons to the perceived expert (Wynne, 1996). Rather, we suggest participatory ES development requires project leaders to acknowledge the dormant politics of those exhibiting as-if trust and intuit the appropriate timing for soliciting their perspectives and garnering support.

Exploiting the functionality of an ES, or at least the functionality that is considered appropriate for the particular organization, is likely to require both modifications to the software and to organizational structures and processes (Wagner and Newell, 2006). This suggests the need to distinguish between immediate ES installation success and long-term implementation/use success, and to observe the influence of participation on both (Markus Tanis, 2000). Complex ES have enormous potential functionality, which is often under-utilized in the immediate or even medium-term aftermath of a system implementation. Wagner and Newell found that only 28 percent of organizations that had implemented an ES had fully deployed the package. Yet, only a small minority (6.5 percent) had failed to deploy, leaving the majority who had deployed with limited exploitation of the functionality. This suggests that efforts in the post-implementation period would be useful in gradually increasing the amount of functionality deployed. The importance of acknowledging timing with regard to ES initiatives is that resources need to be made available in the post-implementation environment to support user participation in a more structured and strategic way than is common currently.

Conclusions

ES are sold by vendors as 'best practice solutions' so one would expect that these packages would be relatively easy to implement and use. Our cases confirm that this is, in reality, far from being the case. Ironically, it could be argued that vendors already recognize this, albeit covertly, in the sense that they support the need for adopting organizations to use consultancy services; indeed they themselves often sell their own consultancy services. In other words, vendors recognize that most organizations are going to require considerable external consultancy support to tailor or customize this supposedly ready-made solution for the particular organization, or to adapt their organization to fit the vanilla system. In some ways this is like selling a ready-made meal that still requires a professional cook to assemble the dish. Our conclusions are therefore based on what is actually happening in practice; despite the rhetoric about "best-practice" our results support the idea that what is best will always be contingent on particular organizational circumstances. Trying to figure out what is best from the packaged software and best from existing practice is going to be an iterative process that is unlikely to be easily worked through in advance of actual implementation and use. This is because users themselves need to participate in this decision process and for a number of reasons – legacy thinking, bottom-up or top-down pseudo participation, motivation – this participation is going to be limited prior to actual implementation and situated learning through use. We therefore advocate seeing an ES in terms of an iterative experience lifecycle where phases of configuration/customization and implementation/use will alternate cyclically, gradually helping to exploit the functionality of the software. Indeed, this might

require a wholesale change in our thinking about organizations and systems development, where the former is seen as an emergent form (Truex, Baskerville and Klein, 1999) and the prototype is the means for accommodating the evolution, with the prototype installed as a system 'in use' (Dearnley and Mayhew, 1983; Avison and Fitzgerald, 1995).

Getting users to buy in to the final product and empowering them to take an active role in their work lives are goals of participation efforts (Mumford, 1983b, c). We argue that post implementation engagement is an effective way of garnering user interest and assistance, at least with respect to an ES. Given that participation is more beneficial in the post-implementation phases of the project, we argue for resources to be provided to enable this to take place. In addition, budgeting for post implementation work efforts can aid in resource-intensive activities like data cleaning. Of course, prior to go-live, it is important to try and clean up the data from legacy systems in order, for example, to avoid duplication of records. However, it is likely to be impossible or at least extremely time-consuming to get the data completely clean. As users actually use the system, they will be more focused and able, iteratively, to pick it up and rectify any problems with the data in order to gradually improve the database that everyone is working with.

Given this iterative cycle of ES exploitation in the post-implementation period, finding ways to speed up and more effectively facilitate this iteration through user participation would seem to be a significant opportunity for organizations. The problem is that if the importance of this post-implementation user participation is not legitimized, it is likely to be less effective and exploitation will be slower (just as post-build additions to one's house often get postponed) although adaptations will eventually happen. This post-implementation is particularly problematic in an ES implementation because members of the core project team who could usefully facilitate user participation during this period may not be available. This may be for a number of reasons – because the team has been disbanded, or because team members have left to take advantage of their ES implementation experience, or simply because, given the long duration of many ES projects, the project team is burnt out. This indicates the importance of developing a strategic plan to encourage user participation during this post-implementation period.

In sum, we call for a reorientation with regard to the role and timing of user participation on ES projects. Post-implementation engagement should be part of the project plan and adequate resources reserved and retained. We argue that ES initiative projects do not, and should not, wind down at go-live. No longer is it appropriate to have organizations interpreting post-implementation changes as an indication of a failure. As shown in the ABC data, project members were devastated by the seeming change of heart of the academic community and felt they could do better the next time. This sense of not having done well enough but being able to do better next time is, we argue, like seeing the pot of gold at the end of a rainbow. The notion of situated learning teaches us that meaningful engagement prior to go-live is limited. Rather than perpetuating this belief, we call for an extension of user participation activities to reflect post-implementation modifications as necessary and important. This is, after all, what is happening in practice.

Acknowledgements: We would like to thank the editors of this special issue for their support and guidance in developing this manuscript. In addition, the comments of the anonymous reviewers were invaluable.

References

- Alvarez, R., and Urla, J. 'Tell me a good story: using narrative analysis to examine information requirements interviews during an ERP implementation,' *SIGMIS Database*, 33(1), 2002, pp. 38 – 52.
- Argyris, C. 'Some inner contradictions in Management Information Systems', in H Lucas, et al., (eds.), *The Information Systems Environment*, Amsterdam: North Holland, Reprinted in R Galliers (ed), *Information Analysis: Selected Readings*, Wokingham: Addison-Wesley, 1987, op cit. pp. 99-111.
- Avison, D., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools* 2nd ed. London: McGraw-Hill, 1995.
- Bauer, M. *The Narrative interview: Comments on a technique for qualitative data collection*. London: London School of Economics and Political Science, 1996.
- Beath, C., and Orlikowski, W. 'The contradictory structure of systems development methodologies: Deconstructing the IS-user relationships in information engineering,' *Information Systems Research* 5(4), 1994, pp. 350-377.
- Checkland, P.B. *Systems thinking, systems practice*, Chichester: Wiley.
- Davis, G. and Olsen, M. *Management of Information Systems*. McGraw-Hill, 1995.
- Dearnley, P.A., and Mayhew, P. J. 'In favor of system prototypes and their integration into the system development life cycle', *Computer Journal* 26(1), 1983.
- Denzin, N.K., and Lincoln, Y.S. 'Introduction: Entering The Field of Qualitative Research,' in *Strategies of Qualitative Research*, N. K. Denzin and Y. S. Lincoln (Eds.), Sage, London, 1998, pp. 1-34.
- Galliers, R.D. *Information Analysis: Selected Readings*, Wokingham: Addison-Wesley, 1987.
- Garfinkel, H. (ed.) *Ethnomethodological Studies of Work*. London: Routledge and Kegan Paul, 1986.

- Holland, C., and Light, B. 'A Critical Success Factors Model for ERP Implementation,' *IEEE Software*, 16(3), 1999, pp. 30-36.
- Howcroft D., and Light B., 'Reflections on issues of power in packaged software selection', *Information Systems Journal*, 16(3), 2006, pp. 215-236.
- Howcroft, D., and Wilson, M. 'Participation: 'Bounded freedom' or hidden constraints on user involvement,' *New Technology, Work and Employment*, 18(1), 2003, pp 2-19.
- Kawalek P., and Wood-Harper, A.T. 'The finding of thorns: user participation in enterprise system implementation, *ACM SIGMIS DataBase*, 33(1), 2002, pp. 13-22.
- Klein, H. K., and Myers, M. D. 'A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems', *MIS Quarterly*, 23(1), 1999, pp. 67-92.
- Land, F., 'Adapting to changing user requirements', *Information & Management*, 5, 1982, pp. 59-75. Reprinted in Galliers (ed.) 1987 op cit.: pp. 203-29.
- Lave, J., and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, Cambridge, 1991.
- Levina, N., and Vaast, E. 'The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems,' *MIS Quarterly*, 29(2), 2005, pp. 335-363.
- Luff, P., Hindmarsh, J., and Heath, C. *WorkPlace Studies: Recovering Work Practice and Informing System Design*, Cambridge: Cambridge University Press, 2000.
- Markus, M.L., Axline, S., Petrie, D. and Tanis, C. 'Learning From Adopters' Experiences with ERP: Problems Encountered and Success Achieved,' *Journal of Information Technology*, 15, 2000, pp. 245-265.
- Markus, L., and Mao, J-Y. 'Participation in development and implementation – updating an old, tired concept for today's IS contexts,' *Journal of the AIS*, 5(11-12), 2004, pp. 514-544.
- Markus, L., and Tanis, C. 'The Enterprise Systems Experience–From Adoption to Success', in R.W. Zmud (ed.) *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Cincinnati, OH, Pinnaflex Educational Resources, Inc., 2000, pp. 173-207.
- Mumford, E., and Weir, D. *Computer systems in work design – the ETHICS method*, New York: Wiley, 1979.
- Mumford, E. *Designing human systems*, Manchester Business School, Manchester, 1983a.
- Mumford, E. *Designing Participatively*, Manchester Business School, Manchester, 1983b.
- Mumford, E. 'Participative systems design: Practice and theory,' *Journal of Occupational Psychology*, 4(1), 1983c, pp. 47-57.
- Mumford, E. *Effective systems design and requirements analysis*, Basingstoke: MacMillan, 1995.
- Mumford, E. 'Socio-technical design: An unfulfilled promise or a future opportunity?' in R. Baskerville, J. Stage and J. DeGross (eds.), *Organizational and Social Perspectives on Information Technology*, London: Kluwer, 2000, pp. 33-46.
- Nah, F. F., Lau J. L., Kuang, J. 'Critical factors for successful implementation of enterprise systems,' *Business Process Management*, 7(3), 2001, pp. 285-296.
- Nah, F., Zuckweller, K. and Lau, J. 'ES implementation: Chief information officers' perceptions of critical success factors,' *International Journal of Human-Computer Interactions*, 16(1), 2003, pp. 5-22.
- Oliver, I., and Langford, H. 'Myths of demons and users: evidence and analysis of negative perceptions of users' in *Proceedings: Australian Computer Conference, Sydney, NSW, November 4-9, Australian Computer Society, 1984*, pp. 453-463. Reprinted in Galliers (ed.) 1987 op cit.: 113-23.
- Orlikowski, W. 'Improvising organizational transformation over time: a situated change perspective,' *Information Systems Research*, 7(1), 1996, pp. 63-92.
- Parr, A., Shanks, G. and Darke, P. 'Identification of necessary factors for successful implementation of ERP systems,' in: O. Ngwenyama., L. Introna., M. Myers. And J. DeCross (eds.), *New Information Technologies in Organizational Process*. Boston, MA: Kluwer Academic Publishers, 1999, pp. 99-119.
- Petty, R. E. and Cacioppo, J. T. *Communication and persuasion: Central and peripheral routes to attitude change*, New York: Springer-Verlag, 1986.
- Rands, T. 'The key role of applications software make-or-buy decisions', *Journal of Strategic Information Systems*, 1(4), 1992, pp. 215-23.
- Robey, D., Ross, J. and Boudreau, M-C. 'Learning to implement enterprise systems: An exploratory study of the dialectics of change,' *Journal of Management Information Systems*, 19(1), 2002, pp. 17-46.
- Rosario, J. 'On the leading edge: Critical success factors in ERP implementation projects'. *Business World*, 2000, Philippines
- Sawyer, S. 'A market-based perspective on information systems development', *Communications of the ACM*, 44(11), 2001, pp. 97-102
- Sauer, C., Liu, L. and Johnston, K. 'Where project managers are kings,' *Project Management Journal*, 32(4), 2001, pp. 39-49.

- Scott, J. and Vessey, I. 'Managing Risks in Enterprise Systems Implementations,' *Communications of the ACM*, 45(4), 2002, pp. 74-81.
- Suchman, L. *Plans and Situated Actions: The Problem of Human/Machine Communication*, Cambridge, UK: Cambridge University Press, 1987.
- Sumner, M. 'Risk Factors in Enterprise-wide/ERP Projects,' *Journal of Information Technology*, 15, 2000, pp. 317-327.
- Truex, D., Baskerville, R., Klein, H., 'Growing systems in emergent organizations', *Communications of the ACM*, 42(8), pp. 117-123.
- Valusek, J. R., and Fryback, D. G. 'Information requirements determination: obstacles within, among, and between participants', in *Proceedings: End-User Computing Conference, Minnesota, ACM Inc., 1985*, Reprinted in Galliers (ed.) 1987 op cit.: 139-51.
- Wenger, E. *Communities of Practice: Learning, meaning and identity*. Cambridge: Cambridge University Press, 1998.
- Wagner, E. and Newell, S. 'Best' for Whom?: The tension between best practice ERP packages and the epistemic cultures of an ivy league university,' *Journal of Strategic Information Systems*, 13(4), 2004, pp. 305-328.
- Wagner, E. and Newell, S. 'Repairing ERP: Producing Social Order to create a working information system,' *Journal of Applied Behavioral Research*, 42, 1, 2006, pp. 40-57.
- Wagner, E., Piccoli, G. 'A Call to Engagement: Moving Beyond User Participation in Order to Achieve Successful Information Systems Design,' *The Communications of the ACM*, forthcoming.
- Walsham, G. *Interpreting Information Systems in Organization*. Chichester: Wiley, 1993.
- Westrup, C. 'Knowledge, legitimacy and Progress? Requirements as Inscriptions in Information Systems Development,' *Information Systems Journal*, 7, pp. 35-54.
- Wong, E. and Tate, G. 'A study of user participation in information systems development' *Journal of Information Technology*, 9, 1994, pp. 51-60.
- Wynne, B. 'May the sheep safely graze? A reflexive view of the expert-lay knowledge divide,' in S. Lash, B. Szerszynski and B. Wynne, *Risk, environment and modernity: Towards a new ecology*, London: Sage Publications, 1996, pp. 44-83.

About the Authors

Sue Newell is the Cammarata Professor of Management, Bentley College and also holds an appointment at Warwick University. She has a BSc in psychology and a PhD from Cardiff University. Dr. Newell is currently the PhD Director at Bentley. She has worked previously at Aston, Birmingham, Nottingham Trent, and Warwick Universities, all in the UK. Her research focuses on understanding the relationships between innovation, knowledge, and organizational networking - primarily from an organizational theory perspective. At Warwick, Dr. Newell was a founding member of ikon, and she continues to focus on research that explores innovation processes using knowledge and organizational networking perspectives. She is also involved in research that explores the implementation and use of packaged information systems. Her research emphasizes a critical, practice-based understanding of the social aspects of innovation, change, knowledge management, and interfirm networked relations. Dr. Newell has published over 60 journal articles in the areas of organization studies, management, and information systems as well as numerous books and book chapters.

Erica Wagner is an Assistant Professor of Information Systems at Cornell University's School of Hotel Administration. She earned her PhD from the London School of Economics and has significant practitioner experience in the fields of accounting and budgeting. Dr. Wagner's research interests focus on the ways in which complex software is "made to work" within different organizational contexts. Her research has appeared in numerous academic and applied journals including Information and Organization, Communications of the ACM, and the Journal of Strategic Information Systems. For more information on Dr. Wagner, visit www.people.cornell.edu/pages/elw32/index.htm.

Copyright © 2007, by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers for commercial use, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints, or via e-mail from ais@gsu.edu.



Journal of the Association for Information Systems

ISSN: 1536-9323

Editor
Kalle Lyytinen
Case Western Reserve University, USA

Senior Editors			
Izak Benbasat	University of British Columbia, Canada	Robert Fichman	Boston College, USA
Varun Grover	Clemson University, USA	Rudy Hirschheim	Louisiana State University, USA
Juhani Iivari	University of Oulu, Finland	Robert Kauffman	University of Minnesota, USA
Frank Land	London School of Economics, UK	Jeffrey Parsons	Memorial University of Newfoundland, Canada
Suzanne Rivard	Ecole des Hautes Etudes Commerciales, Canada	Bernard C.Y. Tan	National University of Singapore, Singapore
Yair Wand	University of British Columbia, Canada		
Editorial Board			
Steve Alter	University of San Francisco, USA	Michael Barrett	University of Cambridge, UK
Cynthia Beath	University of Texas at Austin, USA	Anandhi S. Bharadwaj	Emory University, USA
Francois Bodart	University of Namur, Belgium	Marie-Claude Boudreau	University of Georgia, USA
Susan A. Brown	University of Arizona, USA	Tung Bui	University of Hawaii, USA
Dave Chatterjee	University of Georgia, USA	Patrick Y.K. Chau	University of Hong Kong, China
Wynne Chin	University of Houston, USA	Ellen Christiaanse	University of Amsterdam, Nederland
Mary J. Culnan	Bentley College, USA	Jan Damsgaard	Copenhagen Business School, Denmark
Samer Faraj	University of Maryland, College Park, USA	Chris Forman	Carnegie Mellon University, USA
Guy G. Gable	Queensland University of Technology, Australia	Dennis Galletta	University of Pittsburg, USA
Hitotora Higashikuni	Tokyo University of Science, Japan	Kai Lung Hui	National University of Singapore, Singapore
Bill Kettinger	University of South Carolina, USA	Rajiv Kohli	College of William and Mary, USA
Chidambaram Laku	University of Oklahoma, USA	Ho Geun Lee	Yonsei University, Korea
Jae-Nam Lee	Korea University	Kai H. Lim	City University of Hong Kong, Hong Kong
Mats Lundeberg	Stockholm School of Economics, Sweden	Ann Majchrzak	University of Southern California, USA
Ji-Ye Mao	Remnin University, China	Anne Massey	Indiana University, USA
Emmanuel Monod	Dauphine University, France	Eric Monteiro	Norwegian University of Science and Technology, Norway
Mike Newman	University of Manchester, UK	Jonathan Palmer	College of William and Mary, USA
Paul Palou	University of California, Riverside, USA	Yves Pigneur	HEC, Lausanne, Switzerland
Dewan Rajiv	University of Rochester, USA	Sudha Ram	University of Arizona, USA
Balasubramaniam Ramesh	Georgia State University, USA	Timo Saarinen	Helsinki School of Economics, Finland
Rajiv Sabherwal	University of Missouri, St. Louis, USA	Raghu Santanam	Arizona State University, USA
Susan Scott	The London School of Economics and Political Science, UK	Olivia Sheng	University of Utah, USA
Carsten Sorensen	The London School of Economics and Political Science, UK	Ananth Srinivasan	University of Auckland, New Zealand
Katherine Stewart	University of Maryland, USA	Mani Subramani	University of Minnesota, USA
Dov Te'eni	Tel Aviv University, Israel	Viswanath Venkatesh	University of Arkansas, USA
Richard T. Watson	University of Georgia, USA	Bruce Weber	London Business School, UK
Richard Welke	Georgia State University, USA	George Westerman	Massachusetts Institute of Technology, USA
Youngjin Yoo	Temple University, USA	Kevin Zhu	University of California at Irvine, USA
Administrator			
Eph McLean	AIS, Executive Director		Georgia State University, USA
J. Peter Tinsley	Deputy Executive Director		Association for Information Systems, USA
Reagan Ramsower	Publisher		Baylor University