

EUR Research Information Portal

Implementing Package Enterprise Software in Multi-Site Firms: Intensification of Organizing and Learning

Published in:

European Journal of Information Systems

Publication status and date:

Published: 01/01/2007

DOI (link to publisher):

[10.1057/palgrave.ejis.3000708](https://doi.org/10.1057/palgrave.ejis.3000708)

Document Version

Publisher's PDF, also known as Version of record

Citation for the published version (APA):

van Fenema, PC., Koppius, O., & Baalen, P. (2007). Implementing Package Enterprise Software in Multi-Site Firms: Intensification of Organizing and Learning. *European Journal of Information Systems*, 16(5), 584-598.
<https://doi.org/10.1057/palgrave.ejis.3000708>

[Link to publication on the EUR Research Information Portal](#)

Terms and Conditions of Use

Except as permitted by the applicable copyright law, you may not reproduce or make this material available to any third party without the prior written permission from the copyright holder(s). Copyright law allows the following uses of this material without prior permission:

- you may download, save and print a copy of this material for your personal use only;
- you may share the EUR portal link to this material.

In case the material is published with an open access license (e.g. a Creative Commons (CC) license), other uses may be allowed. Please check the terms and conditions of the specific license.

Take-down policy

If you believe that this material infringes your copyright and/or any other intellectual property rights, you may request its removal by contacting us at the following email address: openaccess.library@eur.nl. Please provide us with all the relevant information, including the reasons why you believe any of your rights have been infringed. In case of a legitimate complaint, we will make the material inaccessible and/or remove it from the website.

Implementing packaged enterprise software in multi-site firms: intensification of organizing and learning

Paul C. van Fenema¹,
Otto R. Koppius² and
Peter J. van Baalen³

¹Faculty of Military Sciences, Netherlands Defence Academy (NLDA), Breda, The Netherlands; ²Department of Decision and Information Sciences, RSM Erasmus University, Rotterdam, The Netherlands; ³Department of Decision and Information Sciences, RSM Erasmus University, Rotterdam, The Netherlands

Correspondence: Paul C. van Fenema,
Faculty of Military Sciences, Netherlands
Defence Academy (NLDA), PO Box 90002,
Breda 4800 PA, The Netherlands.
Tel: +31 76 5273878;
Fax: +31 76 5273255;
E-mail: pc.v.fenema@nlda.nl

Abstract

Packaged enterprise software, in contrast with custom-built software, is a ready-made mass product aimed at generic customer groups in a variety of industries and geographical areas. The implementation of packaged software usually leads to a phase of appropriation and customization. As the associated processes remain ill understood, particularly for multi-site implementations, the objective of this paper is to understand the impact of packaged software in a multi-site organization. Adopting a case study method, this paper reports on a multi-site project that was analyzed at the group, site, and corporate level. Our findings suggest that as organizational units face the unsettling experience of having to implement a single source code across globally distributed sites, packaged software intensifies organizing and learning processes across these levels. The paper identifies specific processes for these levels and concludes with implications for research and practice. Our research extends IS research on packaged software implementation with an emphasis on multi-site firms.

European Journal of Information Systems (2007) 16, 584–598.

doi:10.1057/palgrave.ejis.3000708

Keywords: packaged software; organizing; learning; multi-site firms; global projects

Introduction

The objective of this paper is to understand the organizational impact of packaged software in a multi-site organization. Packaged enterprise software is a ready-made mass product offering users a solution based on design processes aimed at generic customer groups in a variety of industries and geographical areas. Packaged software can be contrasted with custom-built approaches, based on the organization of the development and delivery process. While in other economic sectors, packaged products and services have been a common mode for offering value, packaged enterprise software (further: packaged software) is still relatively new. Examples include workflow software, CRM, ERP, groupware, and e-communication software. Packaged software products are designed by global software companies such as Oracle, SAP, and Microsoft, and implemented by vendors, consultancy companies, or IT staff from the customer organization.

Packaged software is of interest to firms because on the one hand it makes knowledge available that has been elicited from interactions with numerous customers. On the other hand, the costs of packaged software lie substantially below custom-built software. At the same time, the development process aimed at a generic customer base implies that the product will not precisely meet customer requirements (Soh & Sia, 2004). The implementation of packaged software usually leads to a phase of

Received: 31 March 2006

Revised: 10 May 2007

Accepted: 30 August 2007

appropriation, re-development, and customization (Orlikowski, 2000; Boudreau & Robey, 2005; Light & Wagner, 2006). With custom-built software, on the other hand, development involves close interaction with the users' context, substituting for extensive re-development and appropriation.

Thus, it seems that packaged software offers many advantages: cost savings, elimination of requirements definition and development processes, integration and standardization of business processes. However, this appears to come at a price. Implementing the software demands elaborate adaptation processes between the organization and the system (Soh *et al.*, 2000; Boudreau & Robey, 2005), and process integration seems difficult to achieve (Light & Wagner, 2006). Packaged software thus impacts organizations in a complex manner, eliminating some processes and triggering new ones.

This applies in particular to multi-site implementations of packaged software (Markus *et al.*, 2000). Organizations seek to standardize their operations and substitute for elaborate development processes (Davenport, 1998) as part of their efforts to globalize and economize operations. Multi-site implementation processes involve a variety of decision processes, concerning the type of implementation in terms of organizational model, timing and degree of adaptation (Markus *et al.*, 2000). Upon deciding on these issues, organizations face an implementation trajectory with often unknown complexities. That is, depending on the degree of standardization, local sites may resist the system, or insist on unique adaptations geared to their site and national regulatory system that threaten the integrative philosophy of the packaged system.

While much is known on multi-site custom-built new product development (Majchrzak *et al.*, 2000; Malhotra *et al.*, 2001) and software development (Meadows, 1996; Carmel & Tjia, 2005; Kotlarsky & Oshri, 2005), few studies investigate the complexities of multi-site implementation of packaged software. Early work on implementing packaged IS emphasized the organizational impact of ERP, and tended to express concern about the technology's deterministic impact (Davenport, 1998). Subsequent research has contributed significantly to our understanding of how ERP systems get appropriated in practice (e.g. Robey *et al.*, 2002; Scott & Vessey, 2003; Tippins & Sohi, 2003; Soh & Sia, 2004; Boudreau & Robey, 2005; Soh & Kien, 2005). However, literature on the impact of packaged software on organizational practices has remained more fragmented, for example Volkoff *et al.* (2005), and research on multi-site implementation of packaged software is virtually absent.

In this paper, we take the system-in-use as the starting point of our research by conceptualizing it as an occasion for restructuring (Barley, 1986) the organization (in this study across multiple sites). What is missing, in particular, are in-depth studies of implementation trajectories that identify processes triggered by packaged software at different organizational levels characterizing multi-site

firms. Thus, in this paper our objective is to understand the organizational impact of packaged software in a multi-site organization.

This paper is structured as follows. After elaborating on the literature on packaged software implementation and our research method, we introduce the TechCo case, an international electronics manufacturing firm. Next, we present our findings and conclude with a discussion and pointers for further research.

Implementing packaged enterprise software

In this section, we elaborate on the distinctions between packaged and custom-built software. Next, we discuss the implementation of packaged software and in particular multi-site implementations.

Packaged software and time-space discontinuity

The rise of packaged software has revolutionized the world of software development and implementation (Carmel & Sawyer, 1998; Davenport, 1998; Sawyer, 2000). Born out of the 'software crisis' (Howcroft & Light, 2006), packaged software provides dedicated software solutions to particular business functions or processes (Lucas *et al.*, 1988). From a vendor perspective, the main advantage of packaged software lies in its large-scale tradability. Pre-built software can be licensed to a mass market. In order to attain economies of scale software is 'packaged' and standardized. In contrast, custom-built software is made-to-order software, which is specifically developed in cooperation with a specific user community.

Software packaging calls for reconsidering what Orlikowski (1992) has called the time-space discontinuity of software development and implementation. She distinguishes between the design mode and the use mode. The time-space discontinuity is caused by the fact that actions aimed at constituting the technology (e.g. vendor organizations) are separated from actions in which technology is applied (user organization) (Orlikowski, 1992). First, in the design mode, developers build into technology interpretative schemes, facilities and norms (e.g. so-called reference models for various industries). Second, in the use mode, users appropriate technology by assigning shared meaning to it and instantiating the technology in an idiosyncratic manner.

Actor network theorists use the concepts of inscription and translation to analyze the distinction between the two modes. First, inscription refers to the way technical artifacts embody patterns of use and anticipate to the way technology is used (Monteiro, 2001). Inscriptions may vary in strength: strong inscriptions leave little room for interpretative flexibility, which means that adapting the technology to the local context is accompanied with high costs. Second, translation refers to the process in which multiple actors re-interpret and appropriate each other's interest to their own. The meaning of the technology is negotiated in accordance with the specific needs of the context (Bijker *et al.*, 1994; Scott & Wagner, 2003).

While the time-space discontinuity is inevitable as long as developer and user are not the same actor, in the case of packaged software the discontinuity becomes more manifest during the implementation process. Packaged software developers are physically and organizationally separated from product users and rely on numerous intermediaries like customers support lines and sales staff. They are not held responsible for implementation trajectories (Carmel & Sawyer, 1998). This means that successful use of packaged software becomes primarily dependent on the adequate appropriation (translation) of the package to the local context.

To a certain extent appropriation has been pre-built into the product templates by the vendor. These options allow customers to configure the software package to the needs of their organization. Configuration of packaged software refers to the process by which organizations are expected to create local information systems that will be used to meet particular processing and business goals (Wagner *et al.*, 2006). However, numerous recent studies on implementation of large packaged information systems, such as ERP, suggest that end users are still facing many problems because of strong inscriptions (Soh & Sia, 2004; Soh & Kien, 2005) that differ from the user context. While intense efforts and conflicts surface early in the development-use cycle for custom-built software, packaged software seems to postpone this intensity to the 'translation' phase, that is, implementation.

Implementing packaged software

Integrated information systems aim to provide standard solutions for heterogeneous, idiosyncratic demands at local sites. Implementing these standardized solutions by definition disrupts local routines and practices, and creates uncertainty among IT departments and users. In his discussion on the impact of information technology on the coordination between the companies that designed the B-2 Stealth Bomber, Argyres (1999) emphasizes the importance of developing a technical grammar between the cooperating firms. Technical grammar can be seen as a 'set of social conventions' around which the people involved coordinate their activities while acting relatively autonomously. Through the development of new social conventions, the technology becomes embedded and routinized in the context of use. Similarly, with packaged software, users experience complex adaptation processes such as sensemaking, reinterpreting existing routines, and developing a grammar. Fleck (1994) proposes that users innovate within their local context *during* implementation. He distinguishes this learning-by-trying from learning-by-using/doing. While the latter mode focuses on using a system that is already working (for instance, custom-built software that is introduced in an organization), learning-by-trying refers to 'the struggle to get the system to work as an integrated entity' (Fleck, 1994, p. 638). This struggle tends to come later in the development-use cycle for packaged software as compared with custom-built software.

The adaptation process characterizing learning-by-trying seems a particularly apt description for the implementation of enterprise systems, which need to work as an integrated entity.

However, how these adaptation processes unfold within the context of packaged software implementation remains unclear. Current research on ERP systems tends to emphasize different dimensions. First, researchers have explored factors influencing the 'success' of ERP systems. This common perspective in IS research proposes a range of general and ERP-specific independent variables that would explain the failure or success ERP system implementation (Esteves-Sousa & Pastor-Collado, 2001; Motwani & Mirchandani, 2002; Motwani *et al.*, 2002). Second, implementation of ERP systems is considered a complex process due to a variety of changes to business processes (Robey *et al.*, 2002; Soh *et al.*, 2003). Many studies have focused on the difficulties and dynamics of technology-organization adaptation (Cadili & Whitley, 2005; Nandhakumar *et al.*, 2005; Pozzebon & Pinsonneault, 2005; Wang *et al.*, 2006). A third emerging line of inquiry takes implementation as a given and subsequently explores the broader impact of ERP systems. Hitt *et al.* (2002) showed a positive impact of investments in ERP systems at the business level, using financial and stock market data (Hitt *et al.*, 2002). Volkoff *et al.* (2005) showed that the introduction of ERP led to different types of organizational integration (Volkoff *et al.*, 2005). The type and degree of integration would depend largely on the interdependence between business units (a point also made by Gattiker & Goodhue, 2004, 2005).

These three streams of research lead to a number of observations. First, while this emerging stream of ERP impact literature explores performance and business process integration, it does not yet pay attention to organizational implications in terms of organizing processes, learning and capability development. Second, implementing packaged software is supposedly different from developing custom-built software. The abundance of research on the latter topic is not relevant for a situation when customer organizations are confronted with a complex out-of-the-box system. And finally, although many enterprise packages are inherently designed to function across multiple sites, extant research tends to focus on single-site implementations. The next paragraph reviews the implementation studies that focus specifically on the multi-site aspect.

Multi-site implementation of packaged software

The limited number of publications on multi-site implementation of packaged software pay significant attention to choices and trade-offs in the development and implementation process (Markus *et al.*, 2000). Research emphasizes the strong tension between the benefits of global integration and the difficulty of local adaptation (e.g. King & Sethi, 2001; Lai, 2001; Chae & Poole, 2005; Kirsch & Haney, 2006). While these are important problems, our focus in this paper is on the

implementation process and in particular its consequences for organizational processes. Recent research shows a variety of possible outcomes of implementation. Codification enabled by standardized IT can help to build intra- and interorganizational ties (Gattiker & Goodhue, 2005; Rai *et al.*, 2006), but it can also deteriorate these (Levina & Vaast, 2005, 2006). Even if the intended integration across multiple sites is achieved, this may have both positive and negative consequences (Volkoff *et al.*, 2005).

Part of these seemingly paradoxical findings is no doubt due to the inherently emergent nature of software development and implementation (Chae & Poole, 2005). However, analogous to what happened in the literature on the effects of Advanced Manufacturing Technologies (Dean *et al.*, 1992; Aiman-Smith & Green, 2002), we also believe that making sense of these paradoxical findings requires a more careful distinguishing of various processes and levels of analysis in implementation trajectories (Klein & Sorra, 1996).

In general, multi-site implementations demand significant investment in cross-site communication and learning (Cramton, 2001; Kotlarsky & Oshri, 2005). In distributed organizations, locally held perspectives tend to remain disconnected (Vaughan, 1997; Nemiro, 2000). While the same technology could thus lead to different local implementations, with packaged software; however, companies attempt to impose a common template on the organization to move beyond this differentiated legacy environment. These partly opposing forces initiate adaptation processes that remain ill understood (Chae & Poole, 2005).

Summarizing, we would argue that the time-space discontinuity plays out differently for custom-built vs packaged software, and for single vs multi-site projects. Packaged software introduces an extensive discontinuity between R&D units of software multinationals and globally distributed customer bases. In addition, multi-site projects may introduce the complexity and uncertainty of international collaboration. Compared with multi-site projects within the same country (e.g. Majchrzak *et al.*, 2000; Malhotra *et al.*, 2001), the international dimension introduces and exacerbates organizational challenges such as cultural diversity, language problems, and time zone differences (e.g. Carmel & Tjia, 2005). Taking these notions as our starting point, the next objective is to understand the organizational impact of packaged software (in our case an ERP system) in a multi-site organization called TechCo.

Research methods

In the TechCo case study, we analyze the impact of packaged software on organizational processes. The case study research focused on the final phase of TechCo's multi-site ERP implementation during which these effects are likely to become noticeable. The regional project in Southeast Asia was part of a global roll out of

Oracle ERP. The main sites involved in the ERP implementation were Singapore, Malaysia, and China.

Design

Case study design IS researchers increasingly adopt qualitative methods such as case study research to explore the role of ICT in today's complex, technology-intensive organizations, and society. We studied a multi-site implementation of Oracle ERP software at TechCo. For the study reported in this paper, we took inspiration from the literature on packaged software introduced in the previous section yet without developing an explicit conceptual model. Such a model would have shaped our analysis more than we wanted in this research. As we proceeded with our analysis, we introduced more concepts that were considered relevant. From a methodological point of view, we followed the interpretive tradition of information systems research within an exploratory research setup. We were interested in organizational processes that were triggered and affected by the introduction of the ERP system. The principles developed for qualitative IS research (Klein & Myers, 1999) shaped the design of the research, and data collection and analysis. Details on the use of these principles are provided in Appendix A.

Site and focus TechCo offers data storage solutions to customers across the world. The company, headquartered in the US and with regional headquarters in the Far East and Europe, assembles products in Singapore; components are sourced from other Asian countries. The ERP implementation project commenced in the US and gradually spread out to the Far East and Europe. For this study, we had access to some of the Far East sites that were part of a global roll out of Oracle ERP. Local sites in the Far East had thus far customized their own systems according to local routines, and were only loosely tied to the regional headquarters in Singapore. With the new system, people were expected to commit and conform to a single global source code. Only minimal changes were allowed to this code. Throughout the project, standardization and commonality were emphasized because the company sites would have to work from a single source code. Focusing on the final phase of the Far East regional implementation of the ERP system where most of the large manufacturing sites were located (Singapore, Malaysia, and China), we were interested in organizational work processes that are common to most organizations, namely coordination, control, and learning (Kogut & Zander, 1996).

Sampling We gained primary access to the IT group at TechCo's Far East regional headquarters. From there, most sites in the region were coordinated. Interviews were arranged with professionals from the IT organization and multiple user groups, at both the regional headquarters and other Asian sites. Interviewees were

purposefully selected to vary geographical sites, functional areas, and corporate level of our data sources, thus strengthening the adoption of principle 7 (Appendix A).

Research quality A number of steps were taken to ensure the quality of the research, including the trustworthiness of data. First, data collection relied on research and interview protocols. Second, data were carefully ordered, and stored to ensure accessibility, transparency, and traceability. Third, the multiple types of data sources, and different roles of interviewees contribute to the trustworthiness of our findings. Fourth, the initial results and interpretations of our study were checked with IT executives at TechCo. And finally, the authors discussed the findings and possible interpretations at length to clarify points of view and check these against data.

Data collection

Data collection involved TechCo staff from various sites reflecting on-site and cross-site organizational processes when most sites in the Far East were in the final phase of implementing the ERP system. In this phase, teams can reflect on organizational processes they have experienced as part of the ERP implementation project. Following common practice in qualitative research, data from multiple sources were collected to enable triangulation (Eisenhardt, 1989; Klein & Myers, 1999). Data were collected using the following techniques in parallel: semi-structured interviews and impromptu conversations, written and electronic documentation, non-participant observation (Yin, 2002), software demonstration, and feedback on initial findings. In total, 16 in-depth, semi-structured face-to-face interviews were conducted face-to-face or through teleconferencing by the first author with 12 informants from Singapore and Malaysia (Appendix B). These multi-site interviews across different hierarchical levels of the organization enabled cross-checking of data. Interviews lasted 60–90 min on average. The interview protocol was to some extent adjusted over time to reflect evolving insights, and it was customized for hierarchical level and functional area. Furthermore, we collected organization charts, handbooks for the implementation process, project plans, contact lists, and other strategic and operational documentation that the company could make available to us and that appeared relevant to our research. After 2 weeks of on-site research, we processed most of the data and delivered a first cut of the case description and analysis to TechCo. We received feedback and engaged in a few additional exchanges to clarify certain issues. These post-data collection exchanges were used to enhance the completeness and accuracy of our research findings.

Data analysis

In our analysis, we focused on the impact of packaged software on organizational processes. Data from various sources were triangulated to increase confidence in the findings. Data analysis was structured inductively, in

accordance with common principles of interpretive research (Walsham, 1995a,b; Klein & Myers, 1999) and qualitative data analysis (Miles & Huberman, 1994). We distinguish the following phases:

- In the first-order analysis (Corley & Gioia, 2004), data were coded inductively, sticking closely to respondents' (descriptive) language. We took the research goal as guiding principle for identifying group, site, and corporate-level processes that would substantiate organizational effects of the ERP system. For this phase of the data analysis, initially a split between site and corporate-level processes was adopted as a device for structuring the analysis, also to bring the multi-site aspects more into focus. In a subsequent iteration of the analysis, we discovered several processes at the group level within a site, hence the group level was added to further clarify the levels of analysis.
- The next phase of interpretation concerns the translation of the categorized initial concepts into second-order themes (Van Maanen, 1979; Corley & Gioia, 2004). This implies further transitioning to a higher level of abstraction and moving closer to addressing the research goal. We structured our data according to group, site, and corporate level.
- And finally, these themes were reinterpreted in the light of our research goal with the intention of building a theory of organizational impact of packaged software. This resulted in Figure 1, which will be explained in the next section. The arrows do not represent causal mechanisms but reflect modeled temporal phases.

Findings: ERP and organizational processes at TechCo

In contrast with custom-built software, packaged software presents organizations with a generic, ready-made solution. Factors such as lack of user involvement in the production phase make for a complex implementation trajectory to customize the generic – often industry level – solution to the requirements of a particular organization. Our objective with the TechCo case was to unpack this complexity in order to understand the organizational impact of packaged software in a multi-site organization. We distinguish three levels for presenting our findings: group, site, and corporate (Figure 1).

Group-level processes

We found two categories of group-level processes: ownership transition and group organizing (Table 1).

Ownership transition At the start of local implementation projects, TechCo organized kick-off sessions to explain the rationale of the system to local groups. While the packaged system was acquired at the corporate level to reap benefits of mass licenses, implementation processes were pushed down the hierarchy. The vice

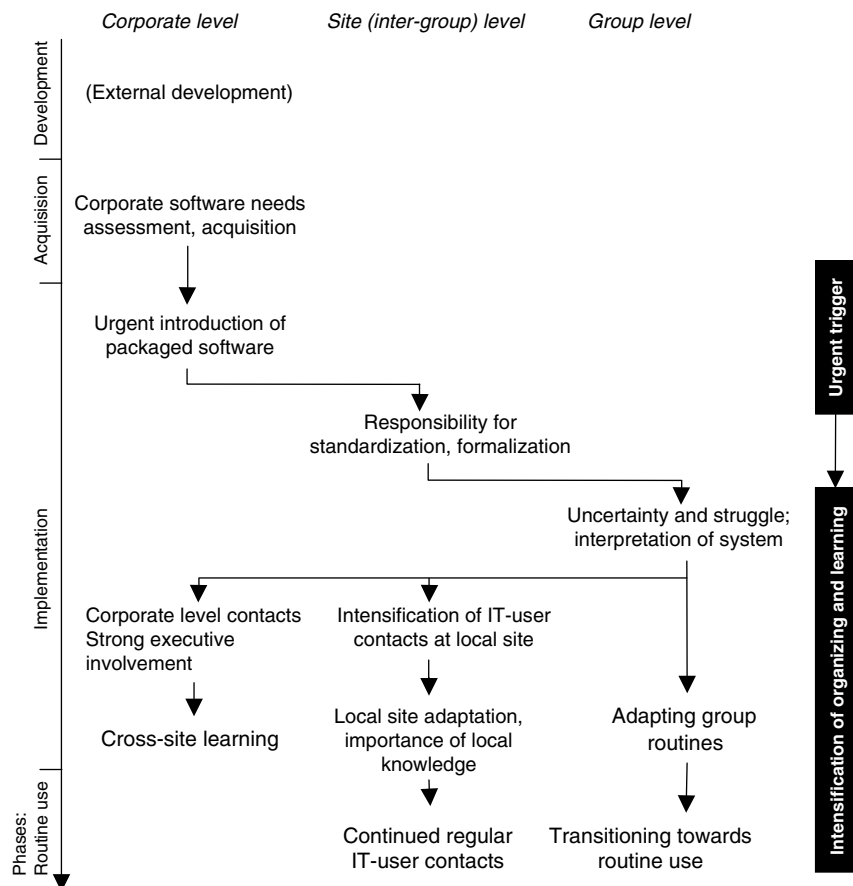


Figure 1 Implementing packaged enterprise software.

Table 1 Group-level processes

Second-order themes	First-order concepts
Ownership transition	<ul style="list-style-type: none"> Interaction at site level initiates group ownership Examination of group level needs and gaps with new system Interfacing at site and corporate level to negotiate group level needs
Group organizing	<ul style="list-style-type: none"> Assignment of key users Key users liaise with local and remote colleagues Key users acquire expertise on the system Key users work with colleagues to share their expertise

president IT from the regional headquarters stressed would be owned at the site and ultimately group level:

When we first started off ... we always have a project kick-off meeting. Not only to communicate to the IT people

about the project. We also communicate with the management there and tell them: *OK this is how we are going to implement the project, and why you want to convert to Oracle, and what is the role of the local key users and IT persons* (vice president IT, Singapore).

Local groups, however, mentioned the considerable challenge they faced. They lacked any knowledge of the new system and were presented with a tight implementation schedule. These groups were involved in examining their current processes and comparing these with the new system. The groups also started to work with local and remote counterparts in order to absorb relevant expertise. Initially, vendor consultants supported some groups. They were phased out, however, once the groups felt confident to proceed on their own.

Group organizing The transition of ownership was followed by intensified efforts to organize and change local groups. A new role of key user was developed. This person would represent the group externally and, vice versa, update colleagues on project progress and solve minor problems within the group. Externally, key users explained current business processes and group expecta-

tions, and at times escalated major problems identified by the group.

When we do the Conference Room Pilot (assessment of gaps between current IT system and new packaged system – authors) they (key users – authors) are the people who are there to do the things. Then when we go into the detailed study and design of the individual gaps, the respective key users involved will have to give input, coordinate, and get input for us from the various users to finalize the design (director of IT, Singapore).

Key users spearheaded their group's learning process by following training on the new system and working with colleagues at other sites. Pushing the responsibility for organizing and learning down the hierarchy to local sites reduced pressure on the central organization. As a side effect, local implementations became somewhat fragmented and lacked structure even though common templates were provided.

Site-level processes

At the site level, TechCo intensified local contacts between IT and users, and cross-site communication (Table 2).

Intensification of IT-user contacts at local site The implementation of the new system intensified the relationship between local IT and supporting user groups. Individuals from the user groups were selected as key user to liaise with local IT and key users at other sites. The key users went through intensive training. They would pass on their expertise to the user group and work with them through the implementation process, as the quote in the previous paragraph illustrated. Key users also had another role. Towards the IT group, they would provide access to business expertise as the same director mentions:

They (key users – authors) help us to understand the business process better. ... And they need to subsequently do the user training for their own user department before the turn on. (director of IT, Singapore)

Key users and IT – forming the so-called core team – would meet very regularly during implementations. Afterwards, they continue their exchanges to handle local IT-user problems, and to implement common standards and changes. IT-user communication thus became more structured (through the key user role) and intense.

Corporate-level contacts As local sites encountered the deadlines for their implementation, individuals started connecting to other sites for training, acquiring hands-on experience, and support (Figure 1). Key users would liaise between sites, as a key user from the regional headquarters told us:

Most of my counterparts (key users at other sites – authors) understand me and I can understand them very well. Because they are also the core people that I communicate with.

Intersite communication processes were formally set up and encouraged by the regional headquarters. The standardization that came with the ERP system promoted cross-site contacts to ensure a common approach:

When the US started their conversion we sent people from here to there to go through their conversion process. ... This is one part of the transfer of knowledge: go there, work with them and learn from them (vice president IT, Singapore).

Because of standardization, experience from sites implementing the technology in an initial phase of the global project became relevant to sites scheduled at later stages. In addition to training programs, local key users and IT

Table 2 Site-level processes

<i>Second-order themes</i>	<i>First-order concepts</i>
Intensification of IT-user contacts at local site	<ul style="list-style-type: none"> • Key user role invented – knowledge and issue solving bridge between users and IT • Frequent meetings of IT-user teams • 'Core team' concept invented (local IT staff and key users), responsible for implementation and conference room pilot (CRP) • Close cooperation users and IT
Corporate-level contacts	<ul style="list-style-type: none"> • Documenting of experiences for other sites to be used • Local staff becomes more internationally oriented, working with remote peers with similar roles, bypassing hierarchy • Complex customizations require external help • Lack of local knowledge of the system and implementation process and method; need for acquiring knowledge and learning new routines • Ambiguity on how the system works and interfaces with remaining systems. This requires external help • Documenting lessons learnt and sharing with other sites • Checking groupware and people from other sites for problem solving • Helpful attitudes towards other sites – 'we're all in the same boat'

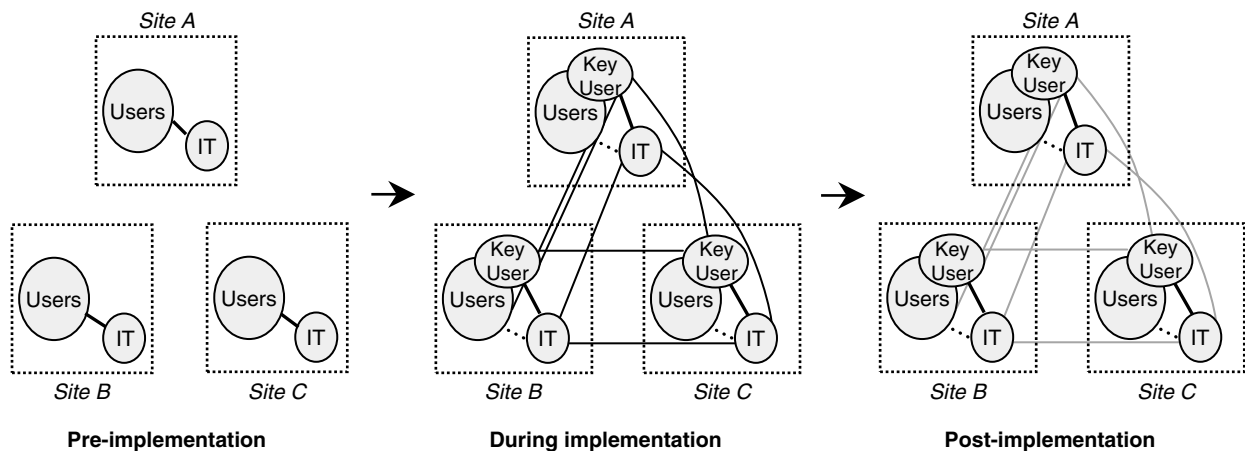


Figure 2 Intensification of organizing and learning.

contacted remote counterparts for support whenever they encountered novel issues that could not be resolved locally. Local key users and IT staff thus initiated corporate-level contacts to resolve their issues, and calibrate their understanding with experts elsewhere. Figure 2 provides a stylized illustration of these two processes. Before the implementation, contacts were concentrated at local sites. During the implementation, contacts intensified; our impression is – even though we did not collect much data on this phase – that after the implementation intersite contacts were routinely maintained to share experiences and synchronize system updates (light gray lines, Figure 2).

Corporate-level processes

Corporate-level processes at TechCo include (1) standardization, formalization, and centralization, and (2) cross-site learning and support (Table 3).

Standardization, formalization, centralization The ERP system changed local operations drastically. Locally adapted systems were replaced by a system with a single source code that would support global sites.

Last time (with the legacy system – authors) we have different sources for different sites ... So we can do whatever things we want, relying on the source itself. But now we cannot. When we go to Oracle we emphasize one common source worldwide. So for any changes that we make we have to consider other sites also (IT member regional headquarters).

The global and common character implied that executives got more strongly involved in IT operations. The VP IT told us his experience:

You can't just sit in your office, waiting for something to happen. Because the impact is so great, when you turn on and it fails, there is no way to go back. (...) Since the consequences are a lot more serious, I cannot just sit there

and wait for people to tell me all the good news. I have to be in there to see it.

Executive-level involvement with site-level IT and business operations thus intensified (Figure 1). Request for changes to the source code had to go through a multi-tier evaluation to ensure their relevance to local operations:

Now, because of Oracle, everything is centralized, everything is one single source code. Any request that comes from the individual organization has to go through a committee to review the request ... we have the objective to establish common processes, common procedures, commonality, standardization (VP IT).

Change requests had to go through an official, hierarchical chain of approvals as they could affect other sites. The IT director at regional headquarters explains how she handles change requests (which TechCo refers to as 'projects'):

After I receive a project and before I approve it I will include more approval into the list to make the US also look at it and make sure they have no issues on the changes. We need to add the requirement specs (specifications – authors), the review, design specs, review, code test, review code, third party test, user test, and installation and assign each task. (...) once they sign off the first task, the mail will send out immediately to the next person to say *OK I have finished this one, now it's your turn to do (for example) a review*.

A global master plan detailed when which sites were expected to convert to the new system. Subsequent documentation spelled out steps to be taken towards the conversion. Progress was frequently briefed and monitored through centralized hierarchical channels, leaving little room for deviations from the plans:

You have to have a command hierarchy. (...) If you leave it open to anybody to make the core decisions, then there is no way that you can really meet that kind of schedule. (...) And in the US (corporate headquarters – authors) the big

Table 3 Corporate-level processes

<i>Second-order themes</i>	<i>First-order concepts</i>
1. Standardization, formalization, centralization	<ul style="list-style-type: none"> • Increased management control and coordination; involvement in local site details • Centralization of communications • Commonality, standardization stressed with respect to the ERP system, business process, and handling the project and conversion problems • Explicit formulation of local responsibilities and roles • Introducing additional formalization: ERP vendor and TechCo method, templates, documentation • Formal controls on status of the project, progress • Stringent restriction on changes to the source code; discouraging local adaptations to the ERP system • Use of hierarchical position and power to enforce standards and planning • Tight planning, integrated master plan (global) • Keeping up pressure on local sites for sticking to plans • Centralized control of source code • Elaborate checklists, plans, standards for handling data • Frequent monitoring, briefings, conference calls for checking progress • Clarify expectations to local sites • Strongly discouraging changes proposed by local sites • Development of interfaces often centralized • Global issues (local site enhancement requests) database: tightly monitored problem solving of (multiple levels in the hierarchy); collective thinking about local problems (anyone can contribute)
2. Cross-site learning and support	<ul style="list-style-type: none"> • Need for learning across sites to understand customizations • Discussions between sites on customizations, and improvement of templates and documentation • Intense cross-site training • Building new relationships between sites and headquarters so that people can learn from peers with similar roles • Use of groupware to support communities around ERP functionality; people 'jump in' with advice • Cross-site assistance, remote and through visits, flexible networks, problem and expertise driven • Intense communications with local sites during key events (kick off, problems, launch) • Intensification of remote contacts (prior more separate islands), sharing of contact lists, building relationships across sites • Headquarter support for local sites • Documentation and sharing of problems and solutions for world wide communities – 'anyone can jump in with an idea' • Extensive corporate communication infrastructure for facilitating inter-site communications (e.g., unlimited phone calls) • Dynamic role change: once local sites gain experience, they support other sites • Problem-solving communications directly between persons, not through hierarchy or liaisons (because of efficiency). Issue and knowledge driven, not organizational structure • Promoting help-oriented, proactive attitudes: informal knowledge sharing, willing to help, mutual support for problem solving • Global issues (local site enhancement requests) database: tightly monitored problem solving of (multiple levels in the hierarchy); collective thinking about local problems (anyone can contribute) • Informal communications between sites, problem solving • Support with development of interfaces between ERP and some local systems • Diversity of operations, complexity, and unique local issues fuel extensive communications and on-site visits • Global knowledge database: evolving representation of past problem solving and experience accumulation across sites • Key users and IT from headquarters support local sites with local conference room pilot meeting • Intra functional communications between sites ('same wave length') • Same role contacts across sites promoted

bosses are all following the schedule very closely. (...) So I think hierarchy is very important. You have a strong support there and it is very clear that that must be done. And it's a *don't ask any questions, just get it done* kind of thing.

And then we will draw all the plans nicely to help you to meet that plan. But don't change the schedule. We are the ones who can change or don't change (vice president IT, Singapore).

Apparently, corporate-level influence on site-level practices was strong and tended to promote rapid implementations.

In line with Adler and Borys (1996), we found that availability of detailed prescriptions was often considered in positive terms. Individuals, having been used to outdated technology, realized that they could benefit from prescriptive documentation that would communicate best practices. About the ERP implementation method, the director of IT mentions:

This methodology helps a lot in the sense that because we had never done such an ERP implementation before, not for my self. (...) I think it is quite a structured way, a method, and it really fits very well.

At the same time, the new method introduces a more complex approach than people were used to in the era of legacy systems:

Using [the old system], you write a very simple report in just probably half a day. If you want to compare this report to Oracle it may take sometimes a week. The tool is not easy. Because now we have to follow [method]. Everything must go step by step. We cannot skip the design specs before you do the coding. It takes longer also (IT member, regional headquarters).

In other words, the corporate level was imposing rigidity and standardization as part of the ERP implementation, which had mixed effects.

Cross-site learning and support Contacts between sites intensify due to the ERP system. Individuals look for knowledge about the new system, and they contribute their experiences to global TechCo databases using Lotus Notes groupware. The VP IT observes this for his regional IT staff:

This whole ERP project makes all the IT people work much closer than before. Before this project, they hardly see each other; they were very independent in the sense that they take care of all local issues, they have the right to change the program. But because of the ERP system, they work as a whole IT group, as a total resource.

The IT director echoes this view and points at the processes induced by the new system:

... we worked as (IT) groups quite independently. But because of the Oracle implementation, we try to emphasize knowledge sharing, communication, and commonality.

Problem solving exchanges take place within functionally similar communities:

[Interactions between finance people] are on the same wave length. As finance people we talk to finance easier. If we talk with IT, we may not really understand some of the technical things that they mention. So it will be better that the patterns of communication go by IT to IT people, and finance to finance (key user, Malaysia).

An issue database environment was built to facilitate global participation in local problem solving. The group-

ware environment invites people to contribute their experiences and support colleagues elsewhere, as an IT member from Malaysia remarked:

Every time when a new site is coming in, we make sure that we paste every issue in the Lotus Notes database. We have an IT enterprise issue database where we posted all kinds of issues so that every site will have the knowledge of what's going on, even if it doesn't happen to their organization. So eventually it is up to the individuals to read that.

Cross-site exchanges were not limited to implementation issues but continue as an ongoing dialogue to ensure consistency. The director of IT explained that she would ensure commonality by working with key users at her sites, who in turn would contact their remote counterparts, also key users:

Sometimes on certain things we want commonality, so I will tell the regional headquarters key users: 'OK please sign up with the other key user sites whether they agree on the way we want to do things.' So I let the key user here do the necessary communication. ... Because they know the business, they have a common language of the things they do.

Cross-site exchanges for problem solving and updating do not follow hierarchical lines but become part of distributed networks of functionally similar professionals:

We don't go through layers. It's important, otherwise it would take a lot of time. If let's say the manager never attends to his emails, managers normally have a lot of emails, you would be delaying things. (...) We talk to the person who is in charge of the module directly. We never go up to their boss (IT member, regional headquarters).

Problem solving thus bypassed hierarchical channels, and occurred more in line with the communication structure depicted in the middle panel of Figure 2.

Discussion and conclusion

Our objective was to understand how the standardized, comparatively rigid nature of packaged software impacts organizational processes. We studied this in a multi-site organization that implemented an ERP system in a standardized, 'plain vanilla' manner. We presented a model to summarize our findings (Figure 1). The main finding of this case study is that the rigid role structures and workflows enforced by the ERP system triggered the creation and intensification of social networks at sites and across sites. In these networks, organizational learning increased compared to the pre-ERP era at TechCo. The standardized nature of packaged software (a single source code) induced new organizing and it necessitated organizational learning. At the same time, it enhanced opportunities for cross-site learning (individuals 'speak the same language') and made learning from others more relevant to the local situations since sites were using the same system.

Packaged software, we argue, thus initiates not only complex and difficult enactment processes as predicted in current literature (Soh *et al.*, 2000; Boudreau & Robey,

2005), but it also triggers social networking and knowledge transfer processes that enable organizational learning. While some of the more detailed findings are due to the specifics of the ERP system, our main results are not so much driven by the fact that we studied an ERP system, but rather by the enforced standardization of activities that are characteristic of third-party packaged software and its inherently generic nature (note that in our case the ERP system was implemented with very little customization). We thus believe that our main results extend more broadly beyond just ERP systems to other packaged enterprise software that has a significant collaborative component for coordination and control of work processes.

Implications for research

Linking back to other types of software Our findings have several implications for current literature on multi-site technology implementation and packaged software in particular. Implementation research has often examined the organizational process of (re)structuring routines and technology, contributing to insight into intra-site dynamics (e.g. Barley, 1986; Orlikowski, 2000). Our study extends this literature by considering the processes constituting the relationships between sites involved in the implementation of the same technology.

While packaged software such as ERP is often considered as rigid and constraining (Davenport, 1998), we found with Soh *et al.* (2000, 2003) and Boudreau & Robey (2005) that organizations adapt the technology to their local practices. ERP systems thus do not impact organizations in a deterministic, linear fashion, even with 'vanilla' implementations. To the contrary, we have highlighted new organizational processes at corporate and site level that were initiated by the technology. These practices promote social integration in the sense of increased cross-site contacts. In a way, ERP made the organization talk and reflect on its practices.

We can reinterpret this finding using Giddens' (1986) notion of reflexivity and extending it to a reflexive capability. Giddens (1986) defines reflexivity commonly at the individual level. He proposes that agents monitor and theorize about their behavior in relationship to intentions, drawing upon their knowledge of a task context (Giddens, 1986). In this study, we found that individuals reflect on and communicate about the new technology. This local-level reflexivity is embedded in organization-wide reflexive practices about standards, adaptations, and changes. Collective reflexivity is thus defined as the organizational level monitoring and discourse about a particular topic aimed at understanding and regulating agency in relation to that topic and organizational objectives.

With packaged software, different sites can still use the same technology in slightly different ways, for instance, for TechCo, legal regulations in Thailand forced adaptations to the ERP system there. Yet at the same time, the homogeneity and interconnectivity of ERP systems

(as well as other collaborative/coordinative enterprise software such as workflow, CRM systems) demand standardization. Reflexive capability could be considered a buffer between these conflicting demands. It involves reinterpretation of the same technology in the light of a corporate wide intention to use the ERP system in a particular mode. From an organizational point of view, reflexivity circumvents to a large extent the organizational hierarchy as our data suggest. Individuals engage in dispersed communities of like-minded professionals to support each other, solve problems, and show-off their expertise. These communities, supported by desktop-accessible global groupware infrastructures, do not replace hierarchy but supplement it.

The capability generated by collective reflexivity may be more important in the long term. We suggest that as the character of technology becomes more complex, integrated, and constraining (i.e. less malleable), reflexivity at local sites and organization wide will increase (Hanseth & Braa, 2001). This effect, contradicting deterministic views, can be explained when one considers the unique nature of local practices (von Hippel, 1994), and the codified nature of technology in a multi-site context. Our research suggests that just focusing on the rigidity and specific adjustments at a single site would ignore these broader effects that we have found. In short, standardization and rigidity initiates collective reflexivity that builds organizational capabilities, and over time increases flexibility (Teece *et al.*, 1997). This also highlights the necessity of taking the multi-site nature of organizations into account when studying the implementation of packaged software.

Implementation and use Our results show a clear support for what Fleck (1994) calls 'learning-by-trying'. Significant adaptation occurred during the implementation process, while the organization struggled with the system in order to get it to work (Fleck, 1994). It is important to note that most of the adaptation did not occur in terms of technological modifications to the system (although some of this did occur), but mostly in terms of the organizational processes. This resulted in emergent outcomes, such as the cross-site learning depicted in Figure 1 (corporate level), which would have been difficult to predict from the technology angle alone.

Implications for practice

Practitioners can learn a number of lessons from this study, with respect to (1) ERP implementations and (2) organizations and IT. First, if they decide to pursue a 'vanilla' implementation, they must invest accordingly in enabling IT and (cross-site) user communities. Vanilla implementation has the advantage of reducing inter-site operability problems on a technology level, but at the same time it increases the need for local and cross-site reflexivity. Neglecting to facilitate this linking up will have adverse effects on the organization. A related implication is that, given the pervasive character of

much packaged software, a CIO may not want to delegate implementation to local IT units. They must consider it as an organizational effort that remains to a large extent tied to their role as executive, while at the same time ensuring local commitment and resource usage. Second, the organization and IT relationship has often been considered from a human agency point of view, with an emphasis on user enactment (Orlikowski, 2000). This leaves room for additional perspectives, namely organization-level phenomenon, IT organization, and strategic perspectives. At the organizational level, reflexivity is promoted by community building around technologies. By being aware of and facilitating this need, organizations are likely to reap benefits beyond those deriving directly from implementing packaged software, such as new organizational structures, and intensified social contacts and learning. Interestingly, this intensification co-exists with notion that packaged software is intended to substitute for and thus reduce the need for operational interaction required for coordinating operational work processes.

Limitations and future research

Any study has its limitations. First, this study was part of a larger research project with a broader focus and the data analysis was grounded. This approach is justified for the type of research question we were pursuing. However, future studies can build more formal *ex ante* models to guide their empirical work. Second, our empirical research investigated only the implementation of packaged software, and covered only the final phase of the implementation project. It was not a longitudinal study and covered only parts of the Far East region of this global

project because of time constraints. A full-scale global study – possibly with globally dispersed research units – would enable a longitudinal study on a global scale. Moreover, future studies may want to pay attention to more specific types of organizational processes and (dynamic) capabilities than the ones we considered here. For instance, at first glance it might seem that there is a fundamental tension between the tacit, amorphous nature of exploration activities of organizations, and the standardized nature of packaged software that is more naturally suited to exploitative activities. At the same time, our study showed that these processes play out differently depending on the level of analysis at which they are examined, so a multi-level study on how organizations manage distributed innovation with packaged software would be very interesting. Furthermore, researchers may want to examine multi-site implementation of a single custom-built application, and compare this process with those found for packaged software. Although the customization work needed for a custom-built application would be less than with packaged software, some of the consequences for the organizing and learning processes across sites may be similar. Future research in this area will help researchers and practitioners to deal with the complexities of managing IT across multiple sites.

Acknowledgements

The authors gratefully acknowledge support from Nanyang Technological University, Singapore (in particular Christina Soh and K. Pelly Periasamy), Erasmus Research Institute of Management, and Netherlands Defence Academy.

About the authors

Paul C. van Fenema is an Associate Professor at Netherlands Defence Academy, The Netherlands. He holds a Ph.D. in Information Systems and Management from RSM Erasmus University, and held positions at this school and Florida International University. His research focuses on coordination, value, and knowledge management in global IS projects and High Reliability Organizations. His work has been widely published in books and journals.

Otto Koppius is an Assistant Professor of Decision and Information Sciences at RSM Erasmus University, where he also received his Ph.D. His main research interests revolve around the dynamics of social networks, the role of IT in enabling intra- and interorganizational coordination, electronic auctions, and predictive modeling.

Peter van Baalen is an Associate Professor of Knowledge Management and e-Learning at the Department of Decision and Information Sciences, and director of the Centre of e-Learning and Knowledge Management (CELK) of RSM Erasmus University. He lectures in the fields of knowledge management, e-organizations, new media and communication in business, and open innovation in the knowledge society. His recent research focuses on knowledge exchange, IT-adoption, open source software development, the emergence of e-communities, and the evolution of formal knowledge networks. Peter J. van Baalen published seven books and about 75 articles in national and international journals, chapters in books, research papers/reports.

References

- ADLER P S and BORYS B (1996) Two types of bureaucracies: enabling and coercive. *Administrative Science Quarterly* **41**, 61–89.
- AIMAN-SMITH L and GREEN S G (2002) Implementing new manufacturing technology: the related effects of technology characteristics and user learning activities. *Academy of Management Journal* **45**(2), 421–430.
- ARGYRES N S (1999) The impact of information technology on coordination: evidence from the B-2 “stealth” bomber. *Organization Science* **10**(2), 162–180.
- BARLEY S R (1986) Technology as an occasion for structuring: evidence from observations of CT scanners and the social order of radiology departments. *Administrative Science Quarterly* **31**, 78–108.

- BIJKER W E, PINCH T J and HUGHES T P (Eds) (1994) *The Social Construction of Technological Systems*. The MIT Press, Cambridge.
- BOUDREAU M-C and ROBEY D (2005) Enacting integrated information technology: a human agency perspective. *Organization Science* **16**(1), 3–18.
- CADILLI S and WHITLEY E A (2005) On the interpretative flexibility of hosted ERP systems. *Journal of Strategic Information Systems* **14**(2), 167–195.
- CARMEL E and SAWYER S (1998) Packaged software development teams: what makes them different? *Information Technology & People* **11**(1), 6–17.
- CARMEL E and TJIA P (2005) *Offshoring Information Technology Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, Cambridge.
- CHAE B and POOLE M S (2005) The surface of emergence in systems development: agency, institutions, and large-scale information systems. *European Journal of Information Systems* **14**(1), 19–36.
- CORLEY K G and GIOIA D A (2004) Identity ambiguity and change in the wake of a corporate spin-off. *Administrative Science Quarterly* **49**(3), 174–208.
- CRAMTON C D (2001) The mutual knowledge problem and its consequences for dispersed collaboration. *Organization Science* **12**(3), 346–371.
- DAVENPORT T H (1998) Putting the enterprise into the enterprise system. *Harvard Business Review* **76**, 121–131.
- DEAN J W, YOON S J and SUSMAN G I (1992) Advanced manufacturing technology and organization structure – empowerment or subordination. *Organization Science* **3**(2), 203–229.
- EISENHARDT K M (1989) Building theories from case study research. *Academy of Management Review* **14**(4), 532–550.
- ESTEVEZ-SOUSA J and PASTOR-COLLADO J (2001) Enterprise resources planning systems research: an annotated bibliography. *Communications of the Association for Information Systems* **7**(8), 1–51.
- FLECK J (1994) Learning by trying – the implementation of configurational technology. *Research Policy* **23**(6), 637–652.
- GATTIKER T F and GOODHUE D L (2004) Understanding the local-level costs and benefits of ERP through organizational. *Information Processing Theory. Information & Management* **41**(4), 431–443.
- GATTIKER T F and GOODHUE D L (2005) What happens after ERP implementation: understanding the impact of interdependence and differentiation on plant-level outcomes. *MIS Quarterly* **29**(3), 559–585.
- GIDDENS A (1986) *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press, Berkeley, CA.
- HANSETH O and BRAA K (2001) Who's in Control: Designers, Managers – or Technology? Infrastructures and Norsk Hydro. In *From Control to Drift. The Dynamics of Corporate Information Infrastructure* (CIBORRA CU, BRAA K, CORDELLA A, DAHLBOM B, FAILLA A, HANSETH O, HEPSON V, LJUNGBERG J, MONTEIRO E and SIMON KA, Eds), pp. 125–147, Oxford University Press, Oxford.
- HITT L M, WU D J and ZHOU X (2002) Investment in enterprise resource planning: business impact and productivity measures. *Journal of Management Information Systems* **19**(1), 71–98.
- HOWCROFT D and LIGHT B (2006) Reflections on issues of power in packaged software selection. *Information Systems Journal* **16**(3), 215–235.
- KING W R and SETHI V (2001) Patterns in the organization of transnational information systems. *Information & Management* **38**(4), 201–215.
- KIRSCH L J and HANEY M H (2006) Requirements determination for common systems: turning a global vision into a local reality. *Journal of Strategic Information Systems* **15**(2), 79–104.
- KLEIN H K and MYERS M D (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* **23**(1), 67–94.
- KLEIN K J and SORRA J S (1996) The challenge of innovation implementation. *Academy of Management Review* **21**(4), 1055–1080.
- KOGUT B and ZANDER U (1996) What firms do? Coordination, identity, and learning. *Organization Science* **7**(5), 502–518.
- KOTLARSKY J and OSHRI I (2005) Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems* **14**, 37–48.
- LAI V S (2001) Issues of international information systems management: a perspective of affiliates. *Information & Management* **38**(4), 253–264.
- LEVINA N and VAAST E (2005) The emergence of boundary spanning competence in practice: implications for implementation and use of information systems. *MIS Quarterly* **29**(2), 335–363.
- LEVINA N and VAAST E (2006) Turning a community into a market: a practice perspective on information technology use in boundary spanning. *Journal of Management Information Systems* **22**(4), 13–37.
- LIGHT B and WAGNER E (2006) Integration in ERP environments: rhetoric, realities and organisational possibilities. *New Technology, Work and Employment* **21**(3), 215–228.
- LUCAS H C, WALTON E J and GINZBERG M J (1988) Implementing packaged software. *MIS Quarterly* **12**(4), 537–549.
- MAJCHRAK A, RICE R E, KING N, MALHOTRA A and BA S (2000) Technology adaptation: the case of a computer-supported inter-organizational virtual team. *MIS Quarterly* **24**(4), 569–600.
- MALHOTRA A, MAJCHRAK A, CARMAN R and LOTT V (2001) Radical innovation without collocation: a case study at boeing-rocketdyne. *MIS Quarterly* **25**(2), 229–249.
- MARKUS M L, TANIS C and VAN FENEMA P C (2000) Enterprise resource planning: multisite ERP implementations. *Communications of the ACM* **43**(4), 42–46.
- MEADOWS C J (1996) Globalizing software development. *Journal of Global Information Management* **4**(1), 5–14.
- MILES M B and HUBERMAN A M (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd edn, Sage, Thousand Oaks, CA.
- MONTEIRO E (2001) Actor-network theory and information infrastructure. In *From Control to Drift. The Dynamics of Corporate Information Infrastructure* (CIBORRA CU, BRAA K, CORDELLA A, DAHLBOM B, FAILLA A, HANSETH O, HEPSON V, LJUNGBERG J, MONTEIRO E and SIMON KA, Eds), pp. 71–83, Oxford University Press, Oxford.
- MOTWANI J and MIRCHANDANI D (2002) Successful implementation of ERP projects: evidence from two case studies. *International Journal of Production Economics* **75**, 83–96.
- MOTWANI J, MIRCHANDANI D, MADAN M and GUNASEKARAN A (2002) Successful implementation of ERP projects: evidence from two case studies. *International Journal of Production Economics* **75**(1–2), 83–96.
- NANDHAKUMAR J, ROSSI M and TALVINEN J (2005) The dynamics of contextual forces of ERP implementation. *Journal of Strategic Information Systems* **14**(2), 221–242.
- NEMIRO J E (2000) The Glue That Binds Creative Virtual Teams. In *Knowledge Management and Virtual Organizations* (MALHOTRA Y, Ed), pp. 101–123, Idea Group Publishing, Hershey, PA.
- ORLIKOWSKI W J (1992) The duality of technology: rethinking the concept of technology in organizations. *Organization Science* **3**(3), 398–427.
- ORLIKOWSKI W J (2000) Using technology and constituting structures: a practical lens for studying technology in organizations. *Organization Science* **11**(4), 404–428.
- POZZEBON M and PINSONNEAULT A (2005) Global-local negotiations for implementing configurable packages: the power of initial organizational decisions. *Journal of Strategic Information Systems* **14**(2), 121–145.
- RAI A, PATNAYAKUNI R and SETH N (2006) Firm performance impacts of digitally enabled supply chain integration capabilities. *MIS Quarterly* **30**(2), 225–246.
- ROBEY D, ROSS J W and BOUDREAU M-C (2002) Learning to implement enterprise systems: an exploratory study of the dialectics of change. *Journal of MIS* **19**(1), 17–46.
- SAWYER R K (2000) The etiquette of improvisation. *Mind, Culture, and Activity* **7**(3), 171–176.
- SCOTT J and VESSEY I (2003) Implementing enterprise resource planning systems: the role of learning from failure. In *Second-Wave Enterprise Resource Planning Systems: Implementing for Effectiveness* (SHANKS G, SEDDON P B and WILLCOCKS L P, Eds), pp. 241–274, Cambridge University Press, Cambridge, UK.
- SCOTT S V and WAGNER E L (2003) Networks, negotiations and new times: the implementation of enterprise resource planning into an academic administration. *Information & Organization* **13**(4), 285–313.
- SOH C and KIEN S S (2005) The challenges of implementing “Vanilla” versions of enterprise systems. *MIS Quarterly Executive* **4**(3), 373–384.
- SOH C, KIEN S S, BOH W F and TANG M (2003) Cultural fits and misfits: is ERP a universal solution? *International Journal of Human-Computer Interaction* **16**(1), 81–100.

- SOH C, KIEN S S and TAY-YAP J (2000) Cultural fits and misfits: is ERP a universal solution? *Communications of the ACM* **43**(4), 47–51.
- SOH C and SIA S K (2004) An institutional perspective on sources of ERP package–organization misalignments. *Journal of Strategic Information Systems* **13**, 375–379.
- TEECE D J, PISANO G and SHUEN A (1997) Dynamic capabilities and strategic management. *Strategic Management Journal* **18**(7), 509–533.
- TIPPINS M J and SOHI R S (2003) IT competency and firm performance: is organizational learning a missing link? *Strategic Management Journal* **24**, 745–761.
- VAN MAANEN J (1979) The fact of fiction in organizational ethnography. *Administrative Science Quarterly* **24**, 539–550.
- VAUGHAN D (1997) The trickle-down effect: policy decisions, risky work, and the challenger tragedy. *California Management Review* **39**(2), 80–102.
- VOLKOFF O, STRONG D M and ELMES M B (2005) Understanding enterprise systems-enabled integration. *European Journal of Information Systems* **14**(2), 110–120.
- VON HIPPEL E (1994) “Sticky information” and the locus of problem solving: implications for innovation. *Management Science* **40**(4), 429–439.
- WAGNER E L, SCOTT S V and GALLIERS R D (2006) The creation of ‘Best Practice’ software: myth, reality and ethics. *Information & Organization* **16**, 251–275.
- WALSHAM G (1995a) The emergence of interpretivism in IS research. *Information Systems Research* **6**(4), 376–394.
- WALSHAM G (1995b) Interpretive case studies in IS research: nature and method. *European Journal of Information Systems* **4**(2), 74–81.
- WANG E T G, KLEIN G and JIANG J J (2006) ERP misfit: country of origin and organizational factors. *Journal of Management Information Systems* **23**(1), 263–292.
- YIN R K (2002) *Case Study Research: Design and Methods*. Sage, Newbury Park, CA.

Appendix A

See Table A1.

Table A1 Principles of interpretive field research

<i>Seven principles</i>	<i>Description</i>	<i>Methods</i>	<i>Application of principles in TechCo case study</i>
1. The fundamental principle of the hermeneutic cycle (overarching principle)	Improving understanding of a complex whole by moving back and forth between the parts and the whole	Repeat cycles of interpretation that keep confronting the larger whole (theoretical concepts, overall meaning of empirical data) with empirical data (the parts)	<ul style="list-style-type: none"> • Data analysis involved multiple researchers having ongoing discussions about the findings
2. The principle of contextualization (emphasis on uniqueness of situation studied)	Difference between interpreter (the present) and author of text (past)	Place subject matter in its social and historical context to clarify for current audiences how situation emerged	<ul style="list-style-type: none"> • Case study description
3. The principle of interaction between the researcher(s) and the subjects (emphasis on uniqueness of situation studied)	Researchers and subjects are placed in historical perspective, i.e., researchers interact with subjects	Interact with an open mind with subjects in their context and history	<ul style="list-style-type: none"> • Direct interviews with multiple company representatives • Observations
4. The principle of abstraction and generalization	Relationship between particulars of a specific study to abstract categories, theories, ideas, concepts (differentiates from anecdotes)	Relate data to abstract theoretical assumptions	<ul style="list-style-type: none"> • Case study findings are analyzed and related to literature
5. The principle of dialogical reasoning	Confrontation of researcher’s prejudices (<i>ex ante</i> lens) with emerging data	Make historical philosophical assumptions guiding the research explicit	<ul style="list-style-type: none"> • Researchers’ assumptions are based on the following literatures: packaged software, ERP, technology adaptation, organizational learning • Researchers’ social background is European; they have been trained as management scholars in Europe and the US
6. The principle of multiple interpretations	Importance of collecting multiple – possibly contradictory – interpretations and incorporate their impact (probing beneath the surface)	Collect multiple interpretations and use them to reinterpret findings	<ul style="list-style-type: none"> • Data have been collected from multiple data sources • The authors have interacted as a team and with multiple external researchers and company representatives • The case study narrative and findings have been presented to TechCo and led to feedback

Table A1 Continued

<i>Seven principles</i>	<i>Description</i>	<i>Methods</i>	<i>Application of principles in TechCo case study</i>
7. The principle of suspicion (critical perspective)	Discovery of false preconceptions, reading social world behind the words of actors	Question surface meaning of data, attention to small cues	<ul style="list-style-type: none"> This questioning of meaning was enabled by: (1) interviewees with different roles at different levels and sites shared their views anonymously. (2) Multiple interviews with some interviewees. This increased the chance of unveiling possibly incorrect or incomplete meanings

Source: Adapted from Klein & Myers (1999).

Appendix B

See Table B1.

Table B1 Interviewee details

<i>Name^a</i>	<i>Role</i>	<i>Location</i>	<i>Number of interviews</i>
1. CPW	Vice President, IT	Singapore	3
2. HHT	Director, Applications Development IT	Singapore	3
3. OBT	Member, Oracle Conversion Team	Singapore	1
4. GP	Member, Oracle Conversion Team	Singapore	1
5. JPL	Member, Oracle Conversion Team	Singapore	1
6. SCC	Member, Oracle Conversion Team	Singapore	1
7. JLL	Key User, Material & System	Malaysia	1
8. ST	Key User, Finance	Malaysia	1
9. ET and on-site colleagues	Key User, Inventory Control	Malaysia	1
10. SKL	Key User, Finance	Malaysia	1
11. JNL	Project Manager, Data Conversion Team	Malaysia	1
12. MC	Member, Data Conversion Team	Malaysia	1
Total			16

^aInitials to preserve anonymity.