Enhancing functionality in an enterprise software package

Judy E. Scott\textsuperscript{a,}\textsuperscript{*}, Lisa Kaindl\textsuperscript{b,1}

\textsuperscript{a}Graduate School of Business, The University of Texas at Austin, Austin, TX 78712-1175, USA
\textsuperscript{b}Dell Computer Corporation, 2214 West Braker Lane, Suite D, Austin, TX 78758-4053, USA

Received 28 April 1998; received in revised form 12 November 1998; accepted 20 July 1999

Abstract

Although enterprise resource planning (ERP) packages strive to integrate all the major processes of a firm, customers typically discover that some essential functionality is lacking. To address this issue and to complement their capabilities, both ERP vendors and customers increasingly recognize the importance of collaboration. Using a grounded theory approach, this study’s objective is to derive a theoretical understanding of how customers collaborated on enhancements to an ERP module. The main contribution is, therefore, a theoretical model that relates the two processes — selection of participants, and interorganizational collaboration — to share knowledge of a subsystem’s best practices. Important findings are that ‘swift trust’ from the occupational community, conflict resolution, reciprocity, and informal networks impact functionality enhancement. The implication from these findings is that a deeper understanding of the functionality enhancement process benefits not only team members, but also ERP customers and collaborators overall. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Enterprise resource planning; Software packages; Organizational learning; Information requirements determination; User requirements; Development team; Collaboration; Trust; Conflict resolution; Informal networks

1. Introduction

Frustration with incompatible legacy systems, IS department inability to cope with systems integration, the year 2000 problem, and the consolidation of currencies in Europe are driving the demand for enterprise resource planning (ERP) software packages. The market for ERP software packages has grown from $4 billion in 1995 to $10 billion in 1997, making predictions of $15.5 billion in 2000 [13] seem conservative and $52 billion in 2002 feasible [12]. ERP software packages strive to support essentially all the processes in a firm’s value-added chain. For example, SAP R/3 currently stores over 1000 predefined processes that represent financial, logistics and human resources best practices in a repository called ‘business engineer’ [4,55].

In an effort to be comprehensive and to be all things to all people, SAP R/3 offers so many options in 10,000 tables that implementation is often extremely complex, necessitating the services of expensive consultants. Yet despite the scale of offerings, most customers inevitably find that at least 20% of their needed functionality is missing from the package.
Enhancing functionality is very important, since alternatives to cope with unmet needs, including forcing business processes to fit the software and bolting on customized programs, add to the time and cost of implementation. Moreover, some alternatives, such as using work-arounds, and modifying the software, increase the difficulty of upgrading to new releases of the ERP package.

Providing enhancements is based on marketing potential and criteria such as the frequency of requests and the influence of the customer, complicated by the fact that customers are physically dispersed, are in multiple external organizations, and have a variety of needs [10,25,31]. In contrast to custom software, package software needs to appeal to the discretionary customer, and must be generic, parameterized and flexible. Usually, the customer and developer communicate through support lines and user groups with intermediaries such as customer support, sales and marketing, or customer surrogates. Yet direct customer-developer links, such as facilitated teams are more effective at clarifying ambiguous information. Direct links transmit multiple cues and decrease communication filtering and distortion.

To satisfy customers and to keep ahead of the competition, ERP software vendors are constantly adding new features using a variety of strategies. For example, to develop their new Treasury module solution to replace old workstations, Peoplesoft bought Advanced Treasury Technology (ATTL) and rented ATTL’s key people [21]. Similarly, Oracle acquired Treasury Services Corporation [19], and also used its Cooperative Application Initiative (CAI) program to interface through its database to treasury workstation vendors, such as ICMS and XRT [47]. XRT also interfaces to J.D. Edwards and SAP packages. In contrast, SAP’s main strategy was to develop its own rich treasury functionality integrated with its other modules to provide foreign exchange, money market investing, lending, derivatives, and debt management functions [44].

This objective of this study is to provide a theoretical explanation of how SAP enhanced the functionality of the R/3 treasury module for the US market with customers’ participation in facilitated teams. Using grounded theory, this study interprets the participants’ accounts of the events. Thus, the main contribution is a conceptual model for customer participation in package software functionality enhancement.

2. Contextual background

2.1. The vendor SAP

Established in Germany in 1972, SAP AG, with 33% market share, is the major ERP package vendor for the Fortune 500 companies. With more than 20,000 employees and an estimated revenue of $3.1 billion for 1997, up 30% from 1996 [14], SAP has become one of the largest software companies in the world. To stay ahead of the competition, SAP spends 20% of its annual revenues on R&D.

2.2. The facilitators

Price Waterhouse Coopers (PWC) worked closely with SAP to develop the Treasury module (TR). PWC undertook a gapping analysis based on the technology standards required by major multinational companies. They met with user groups in Europe and the US to confirm these findings. SAP then used the output of this exercise to set priorities for the TR development program. PWC facilitators provided treasury knowledge, system implementation experience, SAP TR experience, and knowledge of the other SAP modules, as well as expertise in benchmarking and best practices [59] gained from SAP implementation experience. The consultants added value by facilitating meetings as open discussions of requirements, preparing agendas, prioritizing issues, writing requirements documentation, improving process efficiency, providing objectivity and avoiding the bias, conflict of interests and possible confusion that SAP would face without an intermediary.

2.3. Customer participants

Seven organizations formed a steering committee to enhance functionality for the US release of the Treasury module. A group of customers in Europe formed another steering committee, and SAP combined the results from both committees in the development plans for R/3 3.1 and 4.0 releases.
Most of the participating organizations were large global firms from a variety of industries — high-tech, consumer goods, agriculture, food and beverage, financial and chemicals. They reflect a wide range of incomes, with $200 million to $18.5 billion in sales revenues, and 200–38,000 employees in 1996 (see Table 1). SAP selected partners with diverse backgrounds so that it could generalize and standardize best practices applicable to as many industries as possible.

2.4. The ERP software package

SAP’s first two products operated on mainframe hardware; R/1 was batch-oriented, but in 1981 was replaced by R/2, an online system. In 1992, SAP introduced R/3, a client/server architecture product, which quickly gained dominant market share. Despite an evolving architecture, SAP has leveraged its knowledge of enterprise business practices with each release. Using real-time integration, linking a company’s business processes, and supporting immediate responses to change throughout the organization on a global scale, R/3 supports multiple currencies simultaneously and automatically handles country-specific import/export, tax, legal and language requirements. Accepted as a standard in key industries such as oil, chemicals, consumer products, and high technology and electronics, R/3 currently contains modules for over 1000 business processes, that may be selected from the SAP library and included within installed SAP applications, tailoring the application solution to the user. However, R/3’s complexity has generated a lucrative consulting support industry and caused controversy over the risks of escalating implementation project costs.

2.5. SAP R/3 treasury module

The purpose of the R/3 Treasury module is to integrate cash flow management with logistics and financial transactions [2]. Currently, the module, comprises cash management (TR-CM), funds management (TR-FM), treasury management (TR-TM), and market risk management (TR-MRM) [54]. The first manages cash flows by integration between the controlling module and the cash forecast, which groups customers and vendors into risk classes and gives information on long-term liquidity. Within TR-CM, check entry generates postings to general ledger accounts determined by standard or user-defined algorithms. TR-FM is used for budget control and allocation, integrated with purchase orders and release orders in the materials management module. TR-TM controls liquidity management, risk assessment, and planning/portfolio management integrated with postings in the financial system. TR-MRM monitors and manages the entire financial risk position of the company. It uploads market data through automated digital feeds and assesses risks with ‘what-if’ scenarios.

3. Meetings to develop enhancements for the treasury module

The collaborative effort on the US specific Treasury module began with a meeting in January 1996 at Foster City, CA. Fifteen customer organizations were invited to a demonstration of the existing module, on the basis of their sophisticated Treasury knowledge, and asked to discuss what features they liked and disliked. The functionality was rudimentary and geared to Europe. Certain features, such as mortgage
obligation, cash management parsing and polling, check handling and reporting required in the US, were lacking. During this meeting, attendees exchanged information and knowledge on Treasury processes, and planned a Steering Committee with SAP, which also compiled a prioritized list of requirements from the discussions, brainstorming and documentation of functional requirements.

The Steering Committee, developers, and consultants met in subsequent years (see Table 2). The second meeting focused on sharing knowledge of best practices for requirements determination. In addition, participants defined the initiatives, addressed the status of action items, gave feedback, and discussed SAP progress on development plans. Prioritized lists from brainstorming formed the basis for workteam assignments, and the team leader allocated tasks for completion by team members. Furthermore, from the list of requests for functionality the facilitator developed a rough draft of a document to which the participants added more detail. In addition, participants ranked the items as high, medium or low priority. For example, direct download of data from banks had high priority, since that functionality is fundamental to cash management. Although the document resembled a ‘Request For Proposal’ (RFP), it generalized processes from multiple organizations into best practices for the enterprise.

The third meeting involved a mix of SAP training and marketing, and feedback on the prototype it had developed. The prototype allowed users and developers to discuss improvements and problems due to gaps in functionality. The lead developers for each of the Treasury sub-modules gave an intense training session, which was attended by 75 people. Thirty-five SAP consultants, including five from the US, participated. However, there were not as many ways to provide feedback as customer participants expected.

The fourth meeting was held with the annual SAP Sapphire Users’ Conference. It focused on discussion of the prototype demonstration and testing, the documentation of final requirements, and a demonstration of Citibank’s data feed to R/3 on a Treasury workstation. Although it was not feasible for SAP to pull information from banks all over the world, by partnering with Citibank, SAP developed this enhancement and service in cash management. The focus of the fourth meeting was on design approaches.

A fifth meeting was planned but canceled since several of the participants realized they were not ready to implement the Treasury module at that time. One participating organization decided to confine R/3 implementation to the Human Resource component and others postponed the Treasury module since core Financials and Logistics took precedence. In June 1997, two US SAP consultants traveled to Germany to continue the effort to develop the Treasury module.

4. Research methods

This study used insider/outside team research [5,22,61]. One of the co-authors was a participant in the information collection as a member of the Steering Committee for the development of the US Treasury module in the software package SAP R/3. The other co-author, an outsider-researcher, treated all the participants as informants and data sources in the study, this allowed a multiple perspective viewpoint and reduced bias.

Since there is a paucity of research literature on the development of software packages, this study uses grounded theory methodology [58,66]. This is appropriate in the early stages of research on a topic, because it is inductive and does not rely on previous literature or prior empirical evidence [17]. Moreover, this method explains process, ‘how’ research questions, and context, and provides detailed data for deducing constructs for theory generation and elaboration. Grounded theory places emphasis on conceptualizing participants’ accounts of experiences and events to explain a process [49].

The objective of this research is to generate theory to explain how SAP enhanced the functionality of its US module. Customers collaborated to determine the processes that developers embedded in SAP R/3. Two videotapes (60 and 40 min, duration respectively) were transcribed from presentations on the SAP project by the participant to two MBA classes at a large US university during the fall semester of 1996. Further details were communicated to the ‘outside co-author’ in an unstructured informal interview of approximately 11/2 h plus three telephone conversations lasting about 10 min each and four lengthy question-and-answer e-mail messages.
<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Purpose</th>
<th>Events</th>
<th>Meeting outcomes</th>
<th>Shifts in emphasis over time</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 1996</td>
<td>Foster City, California</td>
<td>Knowledge acquisition; Functional requirements</td>
<td>Demonstration of rudimentary prototype; discussions, brainstorming, documentation</td>
<td>Invitations to join development team; Documentation of prioritized functional requirements</td>
<td>Knowledge</td>
</tr>
<tr>
<td>May 1996</td>
<td>Atlanta, Georgia</td>
<td>Functional requirements</td>
<td>Discussions, brainstorming, documentation Status, development plans (SAP progress); feedback</td>
<td>Documentation of prioritized functional requirements; workteam assignments;</td>
<td>Requirements</td>
</tr>
<tr>
<td>June 1996</td>
<td>Walldorf, Germany</td>
<td>Training; marketing; Feedback on prototype</td>
<td>Status, development plans (SAP progress); Training on prototype; feedback</td>
<td>User manual for the prototype is distributed</td>
<td>Design approaches</td>
</tr>
<tr>
<td>August 1996</td>
<td>Philadelphia, Pennsylvania</td>
<td>Marketing; Feedback on prototype</td>
<td>Discussions, brainstorming, documentation Status, development plans (SAP progress); discussion of prototype; Citibank demo</td>
<td>Documentation of final requirements is distributed</td>
<td>Design approaches</td>
</tr>
<tr>
<td>Jan/Feb 1997</td>
<td></td>
<td></td>
<td>Meeting was planned in August 1996 but interest waned and the meeting was not held</td>
<td></td>
<td></td>
</tr>
<tr>
<td>June 1997</td>
<td>Walldorf, Germany</td>
<td></td>
<td>Workshop attended by two US SAP consultants</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In addition, the outsider-co-author interviewed:
two other Steering Committee members on the telephone for about 30 min; a partner from the participating consulting firm by phone for 1 h; a Vice President and three managers from SAP in two face-to-face interviews of 30–40 min duration; an SAP Treasury consultant for 45 min on the phone; and two managers from Oracle Applications in a face-to-face interview for 20–30 min. Background information was supplemented with SAP’s web site and publications on SAP R/3. Published articles on SAP Treasury reinforce the validity of this study. Furthermore, including more than one insider perspective, and incorporating multiple theoretical perspectives at multiple levels of analysis into our discussion, strengthens the generalizability of our findings.

The data collection and analysis were iterative and guided by literature sources on collaboration that functioned as secondary data, increasing theoretical sensitivity. Grounded theory analyzes phenomena by examining causal and intervening conditions and consequences of an action/interaction strategy. Table 3 below lists the causal and intervening conditions, and consequences of the two strategies — the selection and interorganizational collaboration processes. The seven constructs (mutual goals, occupational community, mutual trust, conflict, informal networks, reciprocity and shared knowledge) emerged from open coding, a technique of categorizing data; the processes emerged from axial coding, a technique that links the constructs. Analysis continued until no further concepts emerged — the point at which theoretical saturation was reached.

The grounded theory approach culminated in a conceptual model (Fig. 1) that sheds light on a fresh theoretical perspective of the customer interaction for

### Table 3

<table>
<thead>
<tr>
<th>Grounded theory</th>
<th>Selection process</th>
<th>Interorganization collaboration process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Stakeholders: SAP, customers, facilitators</td>
<td>Software package: Treasury module Steering Committee meetings</td>
</tr>
<tr>
<td>Causal conditions</td>
<td>(1) Mutual goals</td>
<td>(3) Mutual trust</td>
</tr>
<tr>
<td>Intervening conditions</td>
<td>(2) Occupational community</td>
<td>(4) Conflict (5) Informal networks (6) Reciprocity</td>
</tr>
<tr>
<td>Consequences</td>
<td>(3) Mutual trust</td>
<td>(7) Shared knowledge</td>
</tr>
</tbody>
</table>

**Fig. 1.** A conceptual model for ERP package enhancement.
enterprise software development. The conceptual model is a synthesis of two process models that provide explanations for necessary but not sufficient conditions [56]; for example the selection process is necessary but not a sufficient condition for interorganizational collaboration during ERP functionality enhancement.

Furthermore, process models provide ‘the story,. . . , and sequence of events that occur over time’ and focus ‘on the dynamics of social change, explaining how and why the results [occur]’ [42,45]. Although process models can become analytically complex, they do have a specific form and should not be discounted as unscientific or less rigorous than [the more commonly used] factor models. The conceptual model represents the interaction of context with the participant selection and the interorganizational collaboration processes. The process models illustrate relationships among constructs developed with grounded theory.

5. Analyses

5.1. The participant selection process

SAP narrowed the number of organizations participating down to seven, from the 15 attending the original meeting. It selected organizations with state-of-the-art treasury knowledge, such as foreign exchange, or statutory requirements, like GAAP reporting. To ensure mutual goal alignment, SAP chose organizations that were willing to look at best practices and to change their processes, if necessary. Following such discussions, applicants submitted resumes. Due to the rigorous selection process, the chosen individuals felt honored as members of an elite group and were excited to influence the design and be among the first customers to have the module.

Although collaboration among rivals is possible, firms prefer to trade know-how with non-rivals [8,63]. Along these lines, SAP avoided assigning competing customers to the Treasury team. It also exercised judgment in selecting meeting facilitators. These people had a key role in building collaboration by preventing distrust and unbalanced influence, as well as managing conflict resolution and determining best practice. While the accuracy of professionals’ judgment in selecting participants for a collaborative effort varies, when the collaboration has clear goals, the selection of high quality performers should focus on mutual goals.

5.2. Mutual goals

Mutual goals are objectives in common to all participants. In general, cooperative R&D enables firms to share the costs and benefits of projects.

Prior to the steering committee meetings, the functionality of the SAP R/3 treasury module was minimal. Since then, treasury functions have been extended in the areas of foreign exchange, security management and market risk management. Customers were motivated to comply because they realized the advantages of Treasury functionality. A participant explained how integration in the treasury module offered huge benefits.

The hardest thing is to get information out of all departments. Companies don’t communicate between departments. The attitude is ‘my information is my information; what do you want it for?’ In the treasury module, I need to determine cash flow, when I am going to need money, and how long I should invest my money. With SAP, there is a structure created, such that in five minutes I have access to all the accounts payable, all the accounts receivable, all projected sales for the next six months, all materials management expenses, and eventually when cost accounting is implemented — all budget information.

While mutual goals are necessary they are not sufficient to gain mutual trust [16] for collaboration. Other conditions that impacted mutual trust were SAP’s judgment and customer rivalries, and the occupational community.

5.3. Occupational community

Interdependence among collaborators with common professional interests and social similarities, for example R&D engineers, medical technologists or biotechnologists, promotes community level mutualism [50,62]; a shared culture, a community of intent; and a ‘community of fate.’ Participants from the same occupational community, such as academia, share
values, norms and perspectives [48], reinforcing mutual trust. The steering committee customers had in common their occupational domain. Their collective identity, creation of a joint product and commonly shared values, helped develop identification-based trust [34]. Identification-based trust is an emotional connection between parties that have mutual understanding, agree, empathize with each other and take on the other’s values.

5.4. Mutual trust

Mutual trust, the expectation shared by [participants] that they will meet their commitments to one another [43], was reinforced by the legitimacy conferred by SAP on the participants, and the impact of the occupational community. It is surprising that the respondents could not recall any overt signs of mistrust at any of the meetings. Previous studies propose that trust develops slowly and needs long-term prospects [37,60]. Cognitive trust is a rational view of trust and encompasses competence, and reliability [41]. In contrast, affect-based trust is the social view of trust encompassing care and concern [3].

In fact, swift trust was observed. This is cognitive perception in temporary groups from cues such as role clarity rather than interpersonal relations [40]. Swift trust is characteristic of temporary groups; they have less opportunity for dysfunctional interpersonal relationships, because they are fast paced, time constrained and focused on the task at hand [30]. Furthermore, although temporary teams lack history, members assume that the contractor has ‘checked them out’ (a proxy for observing reliability and competence) and inclusion implies selectivity. SAP’s rigorous selection process assured customers that SAP had been sure of the high ability and integrity of its participant firms: antecedents of trust [36].

5.5. The interorganizational collaboration process

Team spirit, friendliness and confidence in each other are measures of trustworthiness. Collaboration is exemplified by software teams that ‘gel’ when the members work together effectively, have a strong sense of identity, a sense of eliteness, a feeling of joint ownership of the product and an attitude of enjoyment [29,67].

Collaboration at the interorganizational level is associated with organizational learning [23,24,26,46]. Interorganizational collaboration catalyzes the learning process by introducing a diversity of new knowledge. In the Treasury meetings, there was a wide diversity of team members since the collaboration spanned organizational, geographical, and cultural boundaries, and a variety of industries and organizational sizes. Brainstorming was the key technique for capturing and exchanging the diversity of knowledge.

Team members learned from each other by openly sharing experiences.

[I]t was easier to learn and share experiences with people from different organizations than within one’s own organization. There seems to be more willingness to share between organizations than between [intraorganizational] teams.

For example, problems developed when the individuals representing the development partner organizations exhausted their personal expertise and had to ask their co-workers for help. Some co-workers were not willing to share, either from lack of motivation or fear that their jobs would later become obsolete [15].

Interorganizational collaboration also catalyzes the learning process by stimulating reconsideration of current practices. By challenging assumptions, shared knowledge is transformed into best practices. Participants explained how the team benefited from the synergy of controversial group discussions. Best practices are acquired from challenging the status quo with double-loop learning. The ‘free flowing ideas’ in the Treasury meetings were characteristic of Model II or double-loop organizational learning behavior [1,18,27,57].

5.6. Conflict

Interorganizational partnerships carry the risk of discord from cultural differences causing different interpretations and misunderstandings [33]. Moreover, the a-conflictual fallacy of systems development [11] posits that the interaction of actors having diverse viewpoints, positions, goals, values, knowledge and interests that need to be reconciled results in ‘a mixed, conflict-cooperation game’ [35]. Conflict and tension are inevitable in these new working relationships.

Explicitly managing conflict is a way to facilitate learning and establish consensus during software
development [20,64]. Conflict is ‘not a debilitating factor needing to be suppressed in the software design team’. Effective conflict resolution increases the accuracy of requirements and the likelihood of project success [51,53]. Dealing with conflict helps error detection, which is the primary stimulus for organizational learning. Organizational conflict is associated with challenging the status quo and double-loop learning [38,39]. Since systems development improves with conflicting views and an attempt to understand the differences, a conscious effort should be made to discuss thoughts, feelings and assumptions relevant to a problem. Yet as assumptions are challenged conflict is likely [28,52]. Furthermore, confronting others by challenging the validity of their positions tends to elicit a defensive response.

There...There were some small tense moments when team members argued over what is important to be developed, and when some companies feel their needs are higher priority. Sometimes someone got defensive when challenged. There are a lot of emotions when it comes to implementing new systems and getting the functionality.

Nevertheless, offering opportunities to suggest alternatives can generate more valid information. Conflict resolution is an organizational learning process and team development process enabling convergence of perspectives to establish standards. However establishing the standard interface to the Treasury module generated arguments since no member wanted to have to maintain the interface.

There was a lot of arguing about different companies’ needs, mostly involving generalized interfaces between third party software companies and SAP. Obviously SAP cannot write an interface with every other system that a company uses. The standard was to choose the most common outside systems, and the users also could place a request with their software vendors to work with SAP to become partners.

5.7. Reciprocity

Reciprocity is payback for contributions to an exchange. Self-amplifying reciprocity can become the emerging norm as member companies take turns in making contributions. There is an obligation to return a favor, resulting in reciprocal learning.

In exchange for our output to them, we get a direct link to the developers. If I have a problem, I don’t have to go through a consultant, or through the hotline support desk. I can email or call a developer directly in Germany and he will either tell me you can do this, or try this or he will go find a solution for it or he will put it on the list of requirements that need to be looked at.

5.8. Informal networks

Collaboration emerges and transactions costs are lower from informal relationships. Moreover, managing the risks in collaboration relies on informal people-based mechanisms, such as interpersonal integration: a network of interpersonal ties between members of separate companies.

Developers often exchange information through informal networks [6]. Because of informal networks and close relationships, SAP software developers responded rapidly to problems and queries from steering committee members and readily communicated by telephone or email. This is contrary to experiences of regular customers, who, because of the SAP expertise shortage, at times experience frustrations making contact with knowledgeable personnel.

Germany was a major information gathering session. I made connections with people I would never [have had] the opportunity to meet with just being an average customer in the States. Our company was a big customer, so I got responses on requests to meet directly with developers. Most SAP consultants don’t even get that benefit. I came home with a much greater understanding of SAP and its methodology.

Trust is more likely to develop during a socialization process in an informal atmosphere [7]. Previous research associates trust and informal networks of relationships, essential for effective collaborations for software development and new product development [32].

5.9. Shared knowledge

Shared knowledge is important. However, organizations need to understand their processes before they share [65]. Knowledge acquisition is an element of both organizational learning and requirements analy-
sis processes [9]. The Treasury participants exchanged information and exhibited reciprocity as they helped each other. The emotional connection enabled the formation of informal networks that increased communications.

The Treasury team members shared firm specific information that in other circumstances might have been considered proprietary. Certainly ‘top secret’ information would not be shared, but organizational knowledge was discussed in far more detail than customary with people external to the organization.

6. Discussion

Our model is new in explaining interorganizational collaboration for enhancing functionality in ERP packaged software. In contrast, traditional software development is not an interorganizational collaboration and does not explicitly incorporate generalized standard processes that SAP calls best practices into requirements analysis.

The grounded theory developed in this study meets the criteria of applicability. First, it fits the substantive area of study. Second, it is understandable to the practitioner. Third, it has generality. Finally, it provides potential control for the action and conditions to which it applies.

6.1. Implications for professionals

While the majority of large US corporations have committed to using ERP packaged software in the last few years, academics have neglected to address practitioners’ concerns. This research contributes by addressing these concerns through understanding the process of enhancing functionality in a vendor-supported team.

The main implication for ERP customers in general is an improved understanding of the functionality enhancement process that will contribute to better decision making. For example, the importance of close collaboration highlights the limitations of user groups and other indirect links to the vendor. Organizing a consortium is a possible way to get a direct link. Setting realistic expectations facilitates corporate planning for missing functionality and consideration of alternatives. Finally, to reduce dependency on one vendor, organizations might consider third party solutions that target this issue. In conclusion, a deeper understanding of the functionality enhancement process benefits not only team members, but also ERP customers and collaborators overall.

References


Judy E. Scott is an assistant professor in the Management Science and Information Systems department at the University of Texas at Austin. She completed her MBA and Ph.D. degrees at the Graduate School of Management, University of California, Irvine. Her research interests include enterprise software packages, organizational learning and the impact of IT on high-tech organizations. Her research has been published and is forthcoming in Communications of the ACM, Data Base, Decision Support Systems, Proceedings of The International Conference on Information Systems, Proceedings of The Hawaii International Conference on System Sciences and several other conference proceedings.

Lisa Kaindl is a project manager at Dell Computer Corporation in Austin, Texas. Although she now works on new product projects, she has been involved as a subject matter expert in the implementation of several software packages. Her MBA degree focused on Finance.