Explaining the Challenges in ERP Development
Networks with Triggers, Root Causes, and
Consequences

Tommi Kähkönen
Lappeenranta University of Technology, tommi.kahkonen@live.fi

Aki Alanne
Tampere University of Technology

Samuli Pekkola
Tampere University of Technology

Kari Smolander
Aalto University

Follow this and additional works at: http://aisel.aisnet.org/cais
Explaining the Challenges in ERP Development Networks with Triggers, Root Causes, and Consequences

Tommi Kähkönen
Lappeenranta University of Technology - Innovation and Software
Lappeenranta, Finland
tommi.kahkonen@live.fi

Aki Alanne
Tampere University of Technology - Laboratory of Industrial and Information Management
Finland

Samuli Pekkola
Tampere University of Technology - Laboratory of Industrial and Information Management
Finland

Kari Smolander
Aalto University—Department of Computer Science
Finland

Abstract:

Many organizations still find it painful to implement an enterprise resource planning (ERP) system. Although ERP projects are collaborative efforts that many separate organizations conduct, academic research has not investigated ERPs fully from this perspective. To identify the challenges in ERP development networks (EDNs), we carried out an interpretive empirical study by using grounded theory to analyze data. After identifying 10 EDN challenges and analyzing the associations between the challenges further, we constructed a model that explains the challenges in ERP development networks. Relationship conflicts (root causes) create or reinforce one or more operational problems (consequences). Changes in the EDN structure initiate or reinforce the other two types. Whereas the existing literature has discussed ERP challenges mainly separately, we offer a more profound explanation of how they emerge and interrelate. Our findings aid practitioners in recognizing and focusing on the root causes of challenges rather than firefighting consequences. The findings can provide useful insights into collaborative and dynamic environments where multiple organizations interact.

Keywords: ERP, Information Systems Development, Challenges, Stakeholders, Network, Empirical Study, Grounded Theory.

This manuscript underwent peer review. It was received 02/06/2015 and was with the authors for 9 months for 3 revisions. Petri Hallikainen served as Associate Editor.
1 Introduction

Enterprise resource planning (ERP) systems promise organizations an all-in-one solution for seamlessly integrating information flows across an organization, which increases competitiveness (Davenport, 1998; Momoh, Roy, & Shehab, 2010). Consequently, researchers and practitioners have paid great attention to ERP systems (Dezdar & Sulaiman, 2009). Although ERP research has reached a certain level of maturity (Schlichter & Kraemmergaard, 2010), ERP projects still tend to exceed schedules and costs and, in the worst cases, lead to project cancellations (Amid, Moalagh, & Zare Ravasan, 2012; Patnayakuni, Rai, & Tiwana, 2007; Pekkola, Niemi, Rossi, Ruuskamo, & Salmimaa, 2013). In 2010, sources have estimated that more than 90 percent of ERP implementations were unsuccessful to some extent (Momoh et al., 2010). An annual report on ERP systems’ success in companies conducted in 2015 points out that 57 percent of ERP projects encounter cost and duration overruns, and 46 percent of organizations believe they receive less than half of the expected benefits (Panorama Consulting Solutions, 2016).

Research considers the large numbers of stakeholders in ERP projects as challenging (Momoh et al., 2010) because ERP development projects are socio-technical endeavors in which numerous stakeholders from different levels and organizations, such as customer organizations (i.e., the organization adopting the ERP system), vendors, consultants, and third parties such as database vendors and business partners, work together to implement the system (Dittrich, 2014; Dittrich, Vaucouleur, & Giff, 2009; Patnayakuni et al., 2007; Sammon & Adam, 2002). In addition, these projects tend to cross national boundaries because ERP vendors tend to outsource parts of the projects to low-cost offshore locations (Levina & Vaast, 2008). These stakeholders form an ERP development network (EDN) (Alanne, Pekkola, & Kähkönen, 2014). Sammon and Adam (2002) point out that a solid understanding of the relationships between the organizations involved in ERP development could be a key milestone in ERP research. However, this network aspect of ERP development has not gained much attention from researchers; in contrast, the majority of the literature focuses on the customer and largely neglects the vendor and other organizations (Koch, 2007; Pekkola et al., 2013). Research has also largely overlooked identifying actors, their roles, interactions, decisions, and impacts on ERP implementation (Bintoro, Simatupang, Putro, & Hermawan, 2015).

In this paper, we investigate ERP development network challenges by examining the EDNs of three large manufacturing enterprises. We refer to EDN challenges as issues that can complicate or cause problems in EDNs’ collaborative development efforts. Thus, we address the following research questions (RQ):

**RQ1:** What are the challenges in EDNs?

**RQ2:** How are these challenges interrelated?

To answer these questions, we conducted an interpretive empirical study by collecting data from three EDNs and their most important organizations and stakeholders. All the ERP systems in these EDNs were custom developed for the specific needs of customers. In total, we conducted 45 semi-structured interviews.

This paper proceeds as follows: In Section 2, we discuss related research on EDNs, ERP challenges, and critical factors. In Section 3, we introduce the cases and the research approach we used to collect and analyze data. In Section 4, we present the identified EDN challenges, explain their associations, and introduce three types of EDN challenges derived from the analysis. In Section 5, we discuss the findings’ novelty and practical implications and the study’s limitations study. Finally, in Section 6, we discuss opportunities for future research and conclude the paper.

2 Background

ERP systems are information systems that integrate a company’s core business processes that monolithic legacy applications previously automated (Alshawi, Themistocleous, & Almadani, 2004; Yusuf, Gunasekaran, & Abthorpe, 2004). ERP systems automate the flow of information, materials, and financial resources of these processes in a single storage, which one can access to obtain the enterprise data whenever needed (Somers & Nelson, 2003; Su & Yang, 2010). ERPs originate from inventory control systems, which organizations later transformed into mainframe-based material requirements planning (MRP) systems (Ebert, 2008). MRP systems converted production plans into detailed requirement schedules of raw materials and components (Grant, Hwang, & Tu, 2013). Later, MRP systems optimized the production process of a plant (Cardoso, Bostrom, & Sheth, 2004; Hwang & Grant, 2011). In the 1980s,
organizations began to adopt MRP II systems to provide control of their resources in production, marketing, and finance (Koh, Gunasekaran, & Goodman, 2011). MRP II systems enabled functional areas, such as sales, production, finance, and accounting, to share data with each other (Grant et al., 2013). In the 1990s, organizations introduced functionalities such as human resources, finance, product planning, and logistics into these MRP II systems and renamed them ERP (Cardoso et al., 2004). ERP systems became the de facto standard in providing the backbone for enterprise integration (Umble, Haft, & Umble, 2003).

Researchers have extensively studied ERP systems with different theoretical lenses, including formal business modeling, the connectionist model, the innovation process, organizational sociology, change management, the supply chain theory, object orientation, organizational memory, and the adoption model (Schlichter & Kraemmergaard, 2010). Critical success factors have been the most dominant theoretical perspective in ERP research. Researchers have examined critical success or failure factors in various settings (see e.g., Aloini, Dulmin, & Mininno, 2007; Dezdar & Sulaiman, 2009; Finney & Corbett, 2007; Momoh et al., 2010; Nah, Lau, & Kuang, 2001; Ngai, Law, & Wat, 2008; Ram & Corkindale, 2014; Saade & Nijher, 2016; Shaul & Tauber, 2013). We synthesize their findings in the following sections.

2.1 Characteristics of ERP Projects

The early phases of an ERP project are crucial because the challenges concern selecting the right type of system and implementation strategy (Finney & Corbett, 2007; Ngai et al., 2008) and a suitable vendor to cooperate with (Aloini et al., 2007). Businesswise, the most common pitfalls result from the lack of a clear vision for the project (Shaul & Tauber, 2013), the misalignment of business and IT strategies and processes (Aloini et al., 2007; Momoh et al., 2010), or the underestimation of the costs and needed resources (Finney & Corbett, 2007; Shaul & Tauber, 2013). In addition, senior management has the power to make or break the project (Leyh & Sander, 2015; Ram & Corkindale, 2014). Without their support and commitment, the project will likely run into trouble because it may, for example, not receive enough resources or not obtain proper legitimization (Aloini et al., 2007; Nah et al., 2001).

Organizations need to manage people, especially employees, in ERP projects. Organizational units or individuals can have their own agendas and objectives that can lead to internal conflicts (Momoh et al., 2010). Without a proper change-management program, challenges are bound to appear (Aloini et al., 2007; Dwivedi et al., 2015; Shaul & Tauber, 2013), such as underestimating change resistance (Momoh et al., 2010) or not considering the roles and responsibilities that implementing the system will change (Nah et al., 2001). Uncertainty, loss of power, lack of involvement in the change process, and reluctance to change are the key issues in any new IT implementation (Ali, Zhou, Miller, & Ieromonachou, 2016). Furthermore, not involving users in the change initiative (Ngai et al., 2006) and, thus, not considering (for example) cultural differences can be a mistake (Finney & Corbett, 2007), which applies especially if training is of poor quality or missing completely (Aloini et al., 2007; Leyh & Sander, 2015; Ngai et al., 2008; Shaul & Tauber, 2013). Moreover, the project team has an important role in ERP projects. While the team composition varies by project, to avoid challenges, the project team should always include the best available personnel, some of them full-time, dedicated members, and it should be cross-functional (Finney & Corbett, 2007; Ngai et al., 2008). If the team members are not co-located or have inappropriate skills and knowledge, problems will likely arise (Dezdar & Sulaiman, 2009; Nah et al., 2001). In addition, the project team should be empowered to make decisions, and a project champion should be appointed (Ngai et al., 2008; Shaul & Tauber, 2013).

The large number of stakeholders participating in ERP projects makes them difficult endeavors and susceptible to conflicts (Momoh et al., 2010). Customer organization must establish and maintain partnerships with external organizations (Aloini et al., 2007; Nah et al., 2001; Ngai et al., 2008; Saade & Nijher, 2016; Shaul & Tauber, 2013). For instance, an unstable or underperforming ERP vendor, lack of vendor support, and vendor lock-in can hinder development (Aloini et al., 2007; Shaul & Tauber, 2013). Moreover, because ERP projects often need consultants, one needs to carefully select and use them (Dezdar & Sulaiman, 2009; Shaul & Tauber, 2013). However, low participation or an inability to transfer the consultants' knowledge introduces challenges (Finney & Corbett, 2007).

ERP development, with multiple stakeholders, is a knowledge-intensive endeavor that requires communication, cooperation, and knowledge management (Leyh & Sander, 2015; Ngai et al., 2008; Patnayakuni et al., 2007; Shaul & Tauber, 2013). Interaction between stakeholders is prone to errors and misunderstandings (Sarker & Lee, 2003). Organizations need to support communication and knowledge integration with proper tools and formal and informal practices (Aloini et al., 2007; Patnayakuni et al., 2007; Shaul & Tauber, 2013).
In addition to exchanging information internally, interorganizational communication must occur (Finney & Corbett, 2007; Ngai et al., 2008). Because ERP projects cross national boundaries, organizations need to coordinate and integrate multiple knowledge sources (Desouza, Awazu, & Baloh, 2006). Offshoring increases the challenges in collaboration and knowledge transfer between the parties (Koltarsky & Oshri, 2005; Levine & Vaast, 2008; Nidhra, Yanamadala, Afzal, & Torkar, 2013).

From a project management point of view, ERP projects are difficult due to their organizational complexity (Leyh & Sander, 2015; Ngai et al., 2008; Ram & Corkindale, 2014; Shaul & Tauber, 2013). Compared to onsite projects, globally distributed projects encounter additional risks and are more difficult to manage (Betz, Oberweis, & Stephan, 2010; Chauhan, Dwivedi, & Sherry, 2012; da Silva, Costa, Franca, & Prikladinicki, 2010). An unclear project plan with no milestones can cause project management problems (Nah et al., 2001; Ngai et al., 2008; Shaul & Tauber, 2013). ERP project managers must be able to handle crises and conflicts (Finney & Corbett, 2007; Ngai et al., 2008). The projects also often need to deal with scope creep and changes in intended schedules (Momoh et al., 2010; Nah et al., 2001). In addition, not having dedicated resources (Momoh et al., 2010; Ngai et al., 2008) and not allocating project responsibilities clearly (Finney & Corbett, 2007; Nah et al., 2001) can hinder ERP project management further.

In a more technical sense, inappropriate IT infrastructure and complications in integrating ERP systems with legacy systems (Dezdar & Sulaiman, 2009; Leyh & Sander, 2015; Shaul & Tauber, 2013), along with poor data-quality management, may hinder ERP systems’ development (Momoh et al., 2010). Similarly, a very complex system architecture or design and excessive customization can cause quality problems (Aloini et al., 2007; Momoh et al., 2010; Ngai et al., 2008). In particular, defining an ERP system’s architectural and functional requirements is difficult (Ngai et al., 2008; Shaul & Tauber, 2013) partly because the requirements tend to change over time (Momoh et al., 2010). Further, improper development tools and techniques further hinder ERP software development (Momoh et al., 2010; Shaul & Tauber, 2013). In addition, poor testing practices (Dezdar & Sulaiman, 2009; Ngai et al., 2008), problems with configurations (Finney & Corbett, 2007; Ngai et al., 2008), lack of software support and maintenance, and inadequate troubleshooting are common causes of ERP software development problems (Aloini et al., 2007; Shaul & Tauber, 2013). Although outsourcing and offshoring software development have promising advantages (Gupta, 2009; Lacity, Khan, Yan, & Willcocks, 2010), these processes still come with many challenges (Lacity, Khan, & Willcocks, 2009; Nidhra et al., 2013). Stakeholders in remote locations may be separated by multiple and overlapping organizational, cultural, national, and professional boundaries (Levina & Vaast, 2008). Consequently, cultural differences, separate time zones, varying working methods, such as processes and approaches to software development, and incongruent levels of common understanding of the end-user environment can cause problems (Al-Salti & Hackney, 2011; Betz et al., 2010), which may slow down the development.

### 2.2 ERP Development Network

Many specialists and stakeholders from different organizations interact and influence one another in the development of strategic and enterprise-wide ERP systems (Damsgaard & Karlsbjerg, 2010; Doolin & McLeod, 2012; Levina, 2005). The term “stakeholder” refers to any individual, group, or organization that affects the system or is affected by it. There can be internal or external stakeholders (Davenport, 1998; Sathish, Pan, & Raman, 2014; Soh, Kien, & Tay-Yap, 2000). The stakeholders form EDNs, which are also called ERP communities (Koch, 2007; Sammon & Adam, 2002) or ERP ecosystems (Dittrich, 2014; Dittrich et al., 2009). In the widest definition, the network includes every actor from the flagship organization, such as Microsoft or SAP, to the users of the customer organization (Dittrich et al., 2009; Doolin & McLeod, 2012; Ernst & Kim, 2002; Sarker, Sarker, Sahaym, & Bjørn-Andersen, 2012). Often, the most important stakeholders are the customer, the vendor, and the consultant (Dittrich et al., 2009; Sammon & Adam, 2002; Soh et al., 2000). In addition, there may be various others from different organizations providing expertise from specific areas (e.g., specialized implementation consultants and local experts and designers) (Dittrich et al., 2009; Ernst & Kim, 2002). To reduce the development costs and to focus on core capabilities, an organization may also partly source development to low-cost offshore locations (Chua & Pan, 2008; Lacity et al., 2009; Levine & Vaast, 2008). Outsourcing development work obviously adds new partners and relationships to the network. The EDN may also have subnetworks and different types of nodes. In addition to working together to develop and implement the ERP system, the actors exchange experiences and innovations in the EDN.
2.3 The Knowledge Gap

Although scholars have commonly used critical success factors as a lens to study ERP projects, critical success factors have not been studied widely from the perspective of different stakeholders and organizations (Tarhini, Ammar, Tarhini, & Masa'deh, 2015). Research has examined the stakeholders but mostly at very high levels (Koch, 2007). In general, IS research has not sufficiently studied ERP development networks, including their problems and interactions (Bintoro et al., 2015; Koch, 2007; Pekkola et al., 2013). For example, research has not studied ERP software development, project management, and communication issues from the EDN perspective; that is, it has not elaborated on how different organizations contribute to these problems.

We address this gap by answering RQ1. One can approach EDNs through the challenges that they encounter. However, instead of limiting our study only to RQ1 only and producing yet another list of critical success factors or challenges in the context of ERP systems, we sought to deepen the understanding of EDNs. Therefore, by answering RQ2, we explain how the challenges and the associations between the challenges emerge in EDNs. We believe that challenges are interrelated as the earlier findings on the interrelated nature of critical success factors or ERP challenges also suggest (Akkermans & van Helden, 2002; Bansal & Agarwal, 2015; King & Burgess, 2006).

Traditionally, researchers have approached EDNs from the perspectives of general businesses (Ernst, 2010; Ernst & Kim, 2002), global vendors and their partners (Sarker et al., 2012), or human communications research in a single company and have separated the network from the development activities (Isomäki & Pekkola, 2010). In contrast, we takes a holistic view to investigate the full breadth of EDNs.

3 Research Approach

Qualitative research methods are especially useful when one seeks to understand a phenomenon from the point of view of practitioners and when a rich social and institutional context is important for the understanding (Kaplan & Maxwell, 2005). Data analysis in qualitative research can create rich descriptions and understandings of social life (Walker, 2006). Because we sought to understand the challenges in ERP development networks and their interrelated nature, we considered the qualitative approach to be suitable for addressing our research problem. Qualitative research can take a positivist, a critical realist, or an interpretive approach (Orlikowski & Baroudi, 1991). We consider our approach interpretive because we sought to make sense of the full complexity of the phenomenon in its social and organizational context (Walsham, 1993).

In line with this approach, we used grounded theory (GT), originally developed by Glaser and Strauss in 1967, as the research method (Glaser & Strauss, 2009). As we stress in Section 2, ERP development projects are complex socio-technical problems, which make the role of stakeholders and human interactions apparent (de Albuquerque & Simon, 2007). They create social, organizational, and technical challenges and require various stakeholders to collaborate and share knowledge to overcome them (Welker, van der Vaart, & Pieter van Donk, 2008). As an inductive research method based on rich real-world research data, GT is especially suitable for approaching this kind of complex organizational phenomenon (Charmaz, 2006). Our specific focus on the ERP development network required in-depth knowledge of the different stakeholders involved in ERP development. Therefore, we approached the subject with an iterative inquiry into the practice and an investigation of the challenges presented from different viewpoints. Without having a predefined theoretical model in mind as grounded theory emphasizes (Kendall, 1999), we investigated the EDN challenges from the viewpoint of one stakeholder at a time and iteratively collected and analyzed the data, which GT supported well.

3.1 Grounded Theory

Grounded theory is useful for creating context-based and process-oriented descriptions of organizational phenomena, and Corbin and Strauss’s (1990) version provides clear guidelines and techniques for analyzing data. One of the benefits of GT is that it allows the researcher to trace back to the original sources of data to observe how the theory has been developed and how different instances of data have emerged into concepts and relationships between them (Corbin, 2008).

The data analysis in this version of GT comprises three coding procedures: open, axial, and selective coding (Corbin, 2008). In open coding, one first labels the transcribed interview data with codes that
capture the data’s meaning. Most importantly, in open coding, one needs to constantly compare pieces of data with each other to find similarities and differences (Walker, 2006). In axial coding, one forms associations between the codes and categories (Kendall, 1999). Basically, this process involves interpreting the codes, categories, and properties developed in open coding to refine the constructs and make them more abstract and theoretical (Urquhart, Lehmann, & Myers, 2010). In selective coding, one chooses a core category, interpret its relationships with the other categories, and explain it as a theory (Corbin, 2008). With selective coding, one seeks to systematically relate the core category (i.e., the central phenomenon of concern that connects all the other categories) and refine the categories into theoretical constructs (Kendall, 1999). When iteratively collecting and analyzing the data, one also needs to determine when to stop. As a theory emerges, one needs to focus more on some particular aspects of the theory. At the same time, collecting more data allows one to better refine the categories, dimensions, and properties. One reaches theoretical saturation when one determines that more data will not further bring about significant new codes, categories, and/or relationships (Corbin, 2008).

3.2 The Case EDNs

We gathered data from three case EDNs to identify the characteristics and challenges related specifically to ERP development networks. We refer to the organizations as case A, case B, and case C.

The customer organization in case A was a large, global manufacturing organization with almost 30,000 employees. In 1996, the organization decided to implement a customized ERP system for sales and logistics because the existing ERP products in the market did not support the company’s specialized business processes and the varying nature of the end product. During the development, the company encountered many challenges (e.g., a company that provided middleware had to architecturally redesign the ERP system in the middle of the project). In addition, the customer organization merged with another company during the busiest implementation phase, which complicated the project. Eventually, the project exceeded its intended budget and schedule. When we interviewed members from the organization, the customer organization was still using the system as intended in all facilities worldwide. Although the system was 20 years’ old, the case company continued to develop it in cooperation with the same vendor and some of the same people as in the beginning. However, the customer organization mostly outsourced technical development to low-cost countries.

The customer organization in case B was a global service provider in retail business with more than 1,000 employees. During the interviews, the company was renewing its ERP system with a customized solution. The old system no longer supported storing the business data and company’s critical business processes. The company and the vendor have had a joint history of more than 15 years because the same vendor also provided the previous ERP system. Now, the vendor aimed to build a general product based on the dedicated system. The project began 2008. By the time we finished collecting data (2013), initial rollouts were about to occur. Other stakeholders included an offshore department, corporation-level IT management, the vendor’s subcontractors and their other customers, and various administrative entities.

The customer organization in case C was a globally operating manufacturing organization with more than 20,000 employees. In 2003, case C decided to implement a customized ERP system for the raw material procurement business with a vendor. The project began 2006 with the first live version in 2008. At the time we finished collecting data, case C was still investing substantially in the project and in the maintenance work to improve the system. In 2011, unlike originally intended, the customer organization rolled out the system in new geographic areas where the core business processes differed dramatically from those in the main country. This rollout introduced several novel challenges. The vendor in this EDN was likewise a large multinational organization, and the cooperation with the vendor lasted several years. Other important stakeholders included offshored units, various administrative entities, and competitors participating in a national standardization initiative.

Our unit of analysis was the EDN. Consequently, although the cases were in different phases of development, were initiated at different times, and were from different contexts, we are not concerned with the use location or context. The cases share some commonalities. For instance, in all three cases, the system under development was built and customized as a result of long-term customer-vendor relationships to fit customers’ particular needs. The organizations all also offshored development to remote locations.
3.3 Research Process

Figure 1 presents the research process we used in this study. The process started with collecting data from the case EDNs. The first author collected and analyzed data for case A and the second author collected and analyzed data for cases B and C. In the second phase, we identified the main categories (the EDN challenges) from the data with open coding. Because both authors performed this process individually, we arranged several brainstorming sessions to harmonize the codes and reach consensus on the EDN challenges. In the third phase, the first and the second authors collaboratively applied the principles of axial coding to discover the associations between the challenges. The final phase of the process comprised selective coding, during which we grouped the EDN challenges into three main categories. In addition, we identified the associations between these categories by relying on the findings in axial coding by which we distinguished three types: triggers, root causes, and consequences of EDN challenges. As a result of our selective coding, we constructed a model that explains EDN challenges. The following sections offer detailed information about each phase of the process.

![Figure 1. The Research Process](image)

3.3.1 Phase 1: Data Collection

In total, we interviewed 45 persons with varying roles in ERP development. We conducted the interviews from January to June in 2013 except for two in February in 2014. One can describe our role as that of a “neutral observer” because none of us was aligned with the case organizations or the interviewees (Walsham, 2006). In each case, we began collecting data via an interview with the main contact person (e.g., the CIO). We chose the following interviewees by snowball or “chain referral” sampling (Pan & Tan, 2011) so we could investigate each EDN by moving from one node to another, pinpoint the key stakeholders, and obtain multiple viewpoints on the same topics. With this process, we could also test single sources of information against others (Myers & Newman, 2007; Pan & Tan, 2011). The
interviewees had different positions that ranged from upper management to mid-level management and developer, and included people from the customer organization, an ERP vendor, and third-party companies. Table 1 presents the interviewees and their organizations.

Table 1. The Interviewees and Their Affiliations

<table>
<thead>
<tr>
<th>Case</th>
<th>Customer business</th>
<th>Customer IT</th>
<th>ERP vendor</th>
<th>Offshore department</th>
<th>Third parties</th>
<th>Total</th>
</tr>
</thead>
</table>
| A    | • Business-IT negotiator  
|      | • Logistics representative | • IT manager of business area  
|      |                           | • Program manager  
|      |                           | • Enterprise architect  
|      |                           | • Service manager of sales  
|      |                           | • IT support manager  
|      |                           | • Project manager  
|      | • Service manager  
|      | • Middleware manager  
|      | • Technical consultant | 17 |
| B    | • Consumer business manager  
|      | • Controller  
|      | • Concept manager  
|      | • Sales office manager  
|      | • Sales person  
|      | • CEO  
|      | • Business area manager | • ICT manager  
|      |                           | • IT support  
|      |                           | • Technical support  
|      |                           | • Former project manager  
|      | • CEO  
|      | • Customer interface  
|      | • Lead designer  
|      | • Product development leader  
|      | • Team leader  
|      | • Developer  
|      | • Corporate CIO | 18 |
| C    | • Development manager 1  
|      | • Development manager 2  
|      | • Solution owner 1  
|      | • Solution owner 2  
|      | • Solution owner 3  
|      | • Project manager | • Team manager  
|      |                           | • Service owner 1  
|      |                           | • Service owner 2  
|      |                           | • System specialist  
|      |                           | -  
|      |                           | -  
|      |                           | -  
|      | 10 |
| Total| 15  
|      | 14  
|      | 10  
|      | 3  
|      | 3  
|      | 45 |

The themes of the semi-structured interviews (Myers & Newman, 2007) covered 1) stakeholders in ERP development, 2) the interviewee’s experiences in the ERP project, 3) issues considered successful, and 4) what should have been done otherwise. This type of interview falls into the category of appreciative interview (Schultze & Avital, 2011) in which we encouraged the interviewee to tell stories about concrete, lived experiences related to their ERP developments’ successes and challenges and explain their strategies for dealing with the challenges. We did not use a strict interview protocol, but we asked additional, detailed questions based on the answers. We believe that this type of interview protocol supported the open-minded nature of grounded theory well in which no existing framework or model determines the analysis process in the beginning of research (Kendall, 1999). We conducted each interview on site at the case organizations, and they lasted from 11 to 111 minutes; the average was about one hour per interview. We recorded and later transcribed the interviews to retain the factual accuracy of the data and to tackle the threat of descriptive validity (Maxwell, 1992). Similarly, to address the threat of misinterpretation, two researchers were present at most of the interviews as Maxwell (1992) suggests.

In addition to interview data, we also gathered secondary research material as is common in qualitative research (Urquhart et al., 2010). In cases A and C, we had access to organizational charts. In case B, we collected requirement specification documents, meeting minutes, and project plans. These materials were not very significant, but, from them, we better understood the organizations and their structures and governance models. We stopped collecting data when the interviews began to repeat the same issues and the interviewees began to recommend persons who we had been interviewed already. We stopped analyzing the data when no new codes emerged from the data, which Corbin (2008) describes as theoretical saturation. Theoretical saturation means that one can move from analyzing to describing and articulating the findings.
3.3.2 Phase 2: Open Coding

According to Corbin and Strauss (1990, p. 61), “open coding is the process of breaking down, examining, comparing, conceptualizing and categorizing the data”. The first and second authors coded the data by using Atlas.ti (Scientific Software Development GmBH, 2016) to identify and code the challenges in the EDN. Open coding started right after we conducted the first interview (Pan & Tan, 2011). Because the researchers each had their own interpretation of coding (Heath & Cowley, 2004) and because all grounded theorists acknowledge that how investigators interpret data determines the codes and categories they select (Kendall, 1999), we needed to have means for harmonizing the open codes so we could compare them.

To do so, we compared and discussed the individually identified EDN challenges in several sessions. During the brainstorming sessions, we harmonized all EDN challenges (see Table 2 for examples). Furthermore, we agreed on common definitions for the categories. We revised the categories continuously to form consistent definitions without overlapping. For example, we further divided the initial challenge “IT service management” into “project management problems” and “software development problems”. During the sessions, we excluded no EDN challenges, but we gave some more descriptive names. At the end of open coding, we had established a list of EDN challenges and their definitions (see Table 3).

We stopped creating new codes and categories after the open coding phase as we noticed that no new EDN challenges emerged from the data and already observed phenomena and patterns began to repeat.

Table 2. Examples of Coding and Harmonization in Open Coding

<table>
<thead>
<tr>
<th>Extracts from the data</th>
<th>Codes used</th>
<th>Interpretation and harmonization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A: “It has been challenging to transfer that knowledge to outsiders who have only technical IT understanding and no understanding of the business.” (Vendor, service owner)</td>
<td>“Challenge: knowledge transfer” “Challenge: homogenous group”</td>
<td>These extracts relate to communication problems. The domain knowledge of each stakeholder group is difficult to transfer outside the group and further throughout the EDN.</td>
</tr>
<tr>
<td>Case B: “The understanding in India is not always as deep as it is here regarding [domain knowledge].” (Vendor, product development leader)</td>
<td>“Business knowledge” “Network”</td>
<td></td>
</tr>
<tr>
<td>Case C: “It takes about two to three weeks [after if I create a service ticket]. …[Someone from India] starts calling, and then it is messed up.” (Development manager 1)</td>
<td>“Business knowledge” “Multinational operations”</td>
<td>These extracts relate to organizational rearrangements. Key individuals play a significant role in the project; their absence disrupts the whole project.</td>
</tr>
<tr>
<td>Case A: “We are training new developers and making sure that deliveries work there, but if a key person decides to leave it's gonna have a big impact.” (Continuous service manager)</td>
<td>“Challenge: people changing”</td>
<td></td>
</tr>
<tr>
<td>Case B: “There was a clear dip in performance when the [project manager] left; there was no single person who had the 13 years of experience with the system.” (Customer IT, IT support)</td>
<td>“Key person leaves” “Communication channel”</td>
<td></td>
</tr>
</tbody>
</table>

3.3.3 Phase 3: Axial Coding

In axial coding, one identifies associations between the codes and categories and may form new codes and categories based on the associations (Corbin, 2008). Although one can perform open and axial coding concurrently as Corbin (2008) suggests, we applied them sequentially. After reaching consensus on the 10 EDN challenges identified in open coding, we performed another analysis round to detect the associations between them. For example, a new partner organization that joined the EDN tried to achieve only its own goals, which differed from the EDN’s overall goal and hampered the project’s management. The axial coding revealed the consecutive nature of the EDN challenges. We learned that some of the EDN challenges, such as project-management and software development problems, caused other challenges, while others (e.g., organizational rearrangements) could put forward a chain of challenges. This finding suggested that we could further abstract the EDN challenges. Figure 2 presents the associations between the EDN challenges that we identified as a result of axial coding. We codified each association for further referral.
### Table 3. The Main Categories Identified in Open Coding

<table>
<thead>
<tr>
<th>EDN challenge</th>
<th>Definition</th>
<th>Example quote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inefficient customer–vendor collaboration</td>
<td>Problems caused by an inefficient method of collaboratively organizing the ERP development between the customer and vendor.</td>
<td>Case B: “I always try to keep clients happy. When there’s a wish from AO…, I’m happy to add the little feature there to keep them happy even if it’s not part of our processes and it slows down the rest of the development. You have to balance quite a bit with that.” (Vendor, Lead Designer)</td>
</tr>
<tr>
<td>Power imbalance</td>
<td>Issues related to power due to the size and impact of an organization that can be capable of steering the project in biased ways.</td>
<td>Case C: “When the budget was cut, it was challenging to employ the good guys [from the vendor]. …If a capable guy leaves the company, it will require years of learning to reach the same level.” (Customer, Solution Owner 1)</td>
</tr>
<tr>
<td>Blind trust</td>
<td>Problems caused when taking one party’s competence for granted or assuming that the other party has the certain knowledge, which it actually does not have.</td>
<td>Case A: “We trusted too much that [the vendor] knows what it’s doing but they didn’t. They just [relied on the same technology] as before and didn’t confirm the functionality.” (Customer, IT Manager of A Business Area)</td>
</tr>
<tr>
<td>Difficulties with third parties</td>
<td>Conflicts and difficulties in partnerships and relationships and between organizations in EDN.</td>
<td>Case A: “Things ran smoothly with [the original database vendor] for a couple of years, but then they became a little greedy at some point, and the license fees starting increasing a little too much. They weren’t as flexible anymore so we decided to [switch vendors].” (Customer, IT Manager of a Business Area)</td>
</tr>
<tr>
<td>Differing objectives</td>
<td>Challenges related to any additional, differing objective that is advantageous for only one organization, group, or individual in EDN.</td>
<td>Case A: “We should have thought more clearly about whether we are making a product or a customized system…. That’s one of the basic things that disrupted the project.” (Vendor, Software Manager)</td>
</tr>
<tr>
<td>Joining and departing organizations</td>
<td>Problems caused when new organizations become involved in EDN or existing organizations leave the network.</td>
<td>Case A: “I was talking about the fence pole, and [the vendor] was talking about the fence. We had agreed on completely different things, and neither of us understood anything.” (Customer, Project Manager)</td>
</tr>
<tr>
<td>Organizational rearrangements</td>
<td>Issues due to changes in team structures or job positions in any organization of EDN.</td>
<td>“The biggest risk is that if critical experts [leave the company]. They have such knowledge that cannot be documented. You can write hundreds of pages that are not useful for anyone. You just have to know it.” (Service Owner 1)</td>
</tr>
<tr>
<td>Project management problems</td>
<td>Issues related to planning and managing the ERP project, including activities such as road mapping, task prioritization, and resource allocation.</td>
<td>Case B: “By roadmap, I mean the vendor’s ability to clearly state when certain stages are finished and what they will include…. That has been a challenge. If there were a clear roadmap to put on the wall and slice into smaller pieces, it would be easier to inform the stakeholders about the status. Also, it would be easier for the project leader to manage the overall thing.” (Customer IT, Project Manager)</td>
</tr>
<tr>
<td>Software development problems</td>
<td>Problems related to software development practices, methods, and tasks such as requirements specification and testing.</td>
<td>Case A: “Practically, [the vendor] didn’t have a clue about how to make it work….. They developed it in a vacuum, and when we looked at it, it seemed that the method of implementing the system and the use of the object model were completely wrong.” (Middleware Consultant)</td>
</tr>
<tr>
<td>Communication problems</td>
<td>Problems related to communication, interaction, and domain knowledge articulation between stakeholders and organizations.</td>
<td>Case C: “For three years, we have had this model with service owners, and it was very personalized to me and [my colleague]. …If we tell [the business people] to create a ticket, …they are faceless resources, but we are the only ones they know by name. This adds a lot to the workload every now and then.” (Service Owner 2)</td>
</tr>
</tbody>
</table>
3.3.4 Phase 4: Selective Coding

Selective coding is similar to axial coding because one develops categories and their dimensions, properties, and relationships but at a higher level of abstraction (Walker, 2006). Strauss and Corbin (1990) consider selective coding as a process of integrating and refining theory. We could have chosen any of the categories as the core category and explained how the corresponding category related to the others. However, we treated all EDN challenges as the core category because we sought to explain how and why the EDN challenges emerge instead of focusing on a single aspect, such as project management. The axial coding results pointed out that we need to further abstract the EDN challenges. For example, we observed that some categories were mainly the causes, not the effects, of the others, which hinted that the categories could have common characteristics. Thus, in selective coding we categorized the EDN challenges further. Table 4 shows the categorization that we made during selective coding. First, we formed three main categories based on the commonalities. For example, inefficient customer-vendor collaboration, power imbalance, and blind trust are all issues related to organizational affairs, so we grouped these challenges under the main category relationship conflicts. Additionally, by examining the directions of the associations identified in axial coding, we determined the type of the main category. For example, we identified project management, software development, and communication problems as consequences because these challenges were the effects of other EDN challenges rather than their causes (though they can affect each other). Similarly, we identified changes in EDN structure as a trigger because the EDN challenges in this category were not the effects of any other challenge.
Table 4. Examples of Coding and Harmonization in Open Coding

<table>
<thead>
<tr>
<th>EDN challenges</th>
<th>Main category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inefficient customer-vendor collaboration</td>
<td>Relationship conflicts</td>
<td>Root cause</td>
</tr>
<tr>
<td>Power imbalance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blind trust</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulties with third parties</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differing objectives</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joining and departing organizations</td>
<td>Changes in EDN structure</td>
<td>Trigger</td>
</tr>
<tr>
<td>Organizational rearrangements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project management problems</td>
<td>Operational problems</td>
<td>Consequence</td>
</tr>
<tr>
<td>Software development problems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication problems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As the result of selective coding, we could construct a model that explains EDN challenges. According to our model, many operational problems occur (consequences) due to relationship conflicts (root causes) or changes in EDN structure (trigger) that generate operational problems directly.

4 Findings

We divide this section into three sub-sections. In Sections 4.1 and 4.2, we explain how the EDN challenges (relationship conflicts and changes in EDN structure) influenced other challenges by breaking Figure 2 down into subfigures for individual EDN challenges and associations. In Section 4.3, we present the outcome of our data analysis—a model that provides an overall explanation for EDN challenges. Subsequent codes in the subfigures refer to associations found in the data.

4.1 Relationship Conflicts

ERP development in the case EDNs turned out to be a cooperative endeavor in which the relationships between the different organizations were critical. The customers and vendors formed long-term relationships in each EDN. During the years of cooperation, the vendors acquired knowledge about each customer’s business, which made the cooperation’s termination or vendor changes not feasible. For instance, the customer considered sticking to the same vendor as self-evident in case B because a new vendor, according to the chief information officer (CIO), “would have had to spend a couple of years learning the domain issues”. The customer sales representative of case A considered the vendor relationship a “forced marriage” and even considered buying the ownership of the source code from the vendor. In addition to the customer’s relationships with the vendor, we identified tensions in the relationships with other organizations. For instance, in case A, differences in opinions between the customer and a database vendor forced the parties to terminate their relationship. We identified that the individual challenges (i.e., inefficient customer-vendor collaboration, difficulties with third parties, blind trust, power imbalance, and differing objectives) were all related to relationship conflicts in the case EDNs. In Sections 4.1.1 to 4.1.5, we present how these challenges were associated with other challenges.

4.1.1 Inefficient Customer–vendor Collaboration

Inefficient customer-vendor collaboration hindered project management, software development, and communication. Communication problems further hindered software development (see Figure 3).
In the beginning of the ERP development project in case A, the customer and the vendor collaborated closely. In the later phases of development, they started to drift apart. Currently, the customer and the vendor created the business and technical roadmaps for the ERP development separately. The vendor felt that the separate roadmaps hindered the software development because they could not use their expertise fully (A-I). In addition, the lack of collaboration hindered communication. The customer was not always willing to participate in the joint project group meetings (A-J). The vendor in case A described the current situation as follows:

*Their [the customer’s] business unit is their IT customer, and our customer is their [the customer’s] IT organization. This is an old model that we’ve stuck with. ...We’d like to create better solutions, and that requires us having direct contact with the [customer’s business].* (Case A, vendor, service owner)

Similarly, in case C, the lack of collaboration between the customer and the vendor caused project management issues (A-H), as the customer’s business made system-related decisions without consulting their IT department or the vendor, and prioritizing development needs became more difficult for the vendor and the customer’s IT department. An interviewee commented the situation as follows: "[Our] IT team and [the vendor’s] team are partly silos. ...Cooperation is good on a personal level, …but I’m not sure whether this is the most efficient and optimal way of working." (Case C, customer, project manager).

In case B, communication problems emerged as a result of off-the-record solutions that the vendor made to satisfy the customer’s stakeholders (A-J), which increased the amount of informal communication and caused the vendor to bypass the agreed-on overall solutions. The communication problems eventually slowed down the software development processes (J-I).

### 4.1.2 Power Imbalance

Power imbalance caused software development and communication problems (Figure 4).
The power imbalance between the customer and the vendor caused problems in software development. In case B, the vendor and the customer differed considerably in size and revenue. The customer was in the driver’s seat: "We have tricks so that all the other [vendor’s] tasks will stop if we face that kind of [major] problem." (Case B, customer, CEO).

In case B, the customer dictated the order in which the vendor developed new features for the system. As a result, the vendor had to discard the original plans for system development, which it had to update and modify according to the customer’s commands. This imbalance of power gave rise to a vicious circle. The system was not ready for rollout, which led to further quality issues and restrained the development of new features (B-I). From the vendor’s perspective, the power imbalance between the customer and the vendor inhibited it from creating proper internal documentation, which the offshore department needed. Continuous balancing between fixing bugs and developing new features put pressure on the vendor's backlog policies. The vendor did not manage these issues systematically, which hindered communication in the vendor organization and between the customer and the vendor (B-J).

4.1.3 Blind Trust

Blind trust between the customer and the vendor hindered project management, software development, and communication. Because of the communication problems caused by blind trust, software development problems emerged (Figure 5).

![Figure 5. Influences of Blind Trust](image)

Long-term relationships created and strengthened personal connections, but this familiarity between the customer and its vendor also introduced challenges. In case B, the customer trusted the vendor’s project management ability and business logic understanding in the initial development phases too much, which resulted in project management problems in terms of inefficient use of resources (E-H). Having too much trust between the stakeholders did not support knowledge transfer because one party tended to assume that the other party already had this knowledge, which caused communication problems. Sometimes, the customer assumed that the vendors and partners had a good understanding of the customer’s business logic and, thereby, took the competence of these parties for granted. In case B, this assumption led to a situation in which the requirement specifications were vague, generic, and unsuitable for every party in the EDN (E-J). For example, the original requirement specifications were created in various ways: “It is difficult to dig up that information when there is no single specification document. ...The specification that is done with the vendor can be just email conversations.” (Case B, customer IT, IT support).

Improper documentation methods increased the communication challenges and confusion about who possessed the relevant information, which caused delays in the development (J-I) because the needed information had to be “fished” from various locations. In addition, in case A, the middleware provider considered the customer’s misplaced trust in the vendor’s ability to choose the technologies as a mistake because architectural problems emerged during the initial rollouts (E-I). These problems ultimately required fixing with a major redesign of the system architecture.
4.1.4 Difficulties with Third Parties

Difficulties with third parties had an influence on communication and software development (Figure 6).

In addition to the customer’s relationships with the vendor, we identified conflicts in the relationships with other organizations. In case A, differences in opinions between the customer and a database vendor forced the parties to terminate their relationship because, according to the customer’s IT manager of a business area, the vendor “became a little greedy at some point” and “was not flexible anymore”. Some organizations, particularly subcontractors and implementation consultants, had a temporary role in the development. In case B, certain subcontractors and offshored developers were not motivated to share and receive development-related knowledge (C-J), which introduced communication problems when these parties joined the EDN: “On one hand, they are [outsiders]. On the other hand, they aren’t our own employees, so they aren’t interested in the business knowledge.” (Case B, vendor, product development leader).

In case A, the middleware provider that entered the project in the middle of the busiest implementation saw that cooperation with the vendor presented initial difficulties when redesigning the system architecture. According to the middleware consultant, “[the vendor] did not really pull us in”. It took some time for the middleware provider to convince the vendor of the value of the new, previously unknown technology, which delayed the software development (C-I).

4.1.5 Differing Objectives

Differing objectives introduced project management and software development problems. The different objectives also hindered third-party relationships (Figure 7).

The differing objectives of individual units stalled the development efforts. In case A, the business people were reluctant to participate in the project in later phases: “As the [ERP] project dragged on and ran into complications, I must say the business people disappeared along the way.” (Case A, sales representative).

Likewise, in case B, the business representatives were not interested in participating in the project after the planning phase, which hindered software development because the business representatives dismissed the requirement specification reviews, and IT had to follow the original definitions (D-J).

In cases A and B, the vendors’ objectives conflicted with the overall EDN goals. The vendors attempted to build or customize a system to serve the customer’s needs while simultaneously attempting to build a
general product for other customers. In case A, the vendor’s software manager stated that the attempt to build a product was “one of the fundamental issues that distracted the project”. The vendor later discarded the initial plans for making a general product as the amount of customer-specific logic increased. The vendor in case B saw that making a product required compromises, and the vendor’s lead designer even stated that “we can’t make everything according to their [the customer’s] wishes”. However, trying to achieve both goals caused resource problems (D-H). Particularly, the other customers’ needs tended to conflict with the original customer’s requirements, and the vendor had to constantly balance the requirements and make compromises. This situation disturbed the relationships between these organizations by causing tensions (D-C). For example, in case B, the customer was unhappy about paying for the development of the basic functionality for the vendor’s product. Additionally, in case C, the project manager noted that the vendor could use the best practices and ideas learned during the project with other customers.

4.2 Changes in EDN Structure

We identified two categories that changed the EDNs’ structure. These changes altered the current configuration of the organizations and their relationships. The EDNs evolved from new organizations’ joining the EDN and some organizations’ leaving. The customers found establishing close relationships with the new organizations as sometimes problematic. In case B, the third-party organizations that temporarily joined the EDN were not always used to the existing working methods. When additional parties joined, the customer and the vendor could no longer use agile methods to exchange development ideas. In all three EDNs, the vendor offshored the development work to remote countries, which shaped the EDNs and brought in new organizations. In addition, in all cases, unlike originally intended, the customer deployed the ERP systems under construction to new business locations, which not only brought new organizations into the network but also increased operational problems. In addition, organizational rearrangements in any organization can introduce additional challenges for development. For example, a key individual person may leave the project and cause a significant loss of knowledge.

4.2.1 Joining and Departing Organizations

When new organizations joined the EDNs, difficulties in third-party relationships and differing objectives arose. The new organizations also caused operational problems due in particular to offshoring (Figure 8).

![Figure 8. Influences of Joining and Departing Organizations](image)

The ERP developments often occurred in several waves when they involved different organizations, which inhibited them from establishing close relationships with each other and even affected their attitudes toward each other. The vendor did not fully include the third parties in developing the ERPs, and these third parties felt that they had been left out (F-C). A new organization sometimes found it difficult to join the EDNs because they needed knowledge about the organizations and the customers’ environments. In offshoring, the vendor had to further transfer the domain knowledge that was embedded in practice to remote locations, which was not a straightforward process (F-J):
There's a lot of know-how in the heads of our guys in this country. It has been challenging to transfer that knowledge to outsiders who have only technical IT understanding and no understanding of the business. (Case A, vendor, service owner)

When vendors joined the EDNs, differing objectives emerged (see F-D above). In addition, problems occurred due to misunderstandings between the customer and the vendor. In case A, the vendor did not have well-established software development practices at the beginning of the project. Because of insufficient testing, the first system deployment was unsuccessful (F-I). According to the middleware consultant, the vendor conducted the testing “in a vacuum” that did not match the real environment. In addition, the vendor had no tools to manage change requests during the development: “I asked how many change requests we have and what size. The vendor didn’t have any. They were scattered all over the place. It took two months to collect them.” (Case A, customer IT, project manager).

Misunderstandings with organizations that had joined the EDNs recently caused communication problems (F-J). For example, in case B, the vendor regarded the current phase of the project as piloting, while the customer’s management thought the project was in the planning phase because it missed half of the originally intended modules. Another example of a misinterpretation concerned the number of missing features. From the perspective of the vendor’s lead designer, most of the crucial features were in place: “At the moment, we have an understanding that there shouldn't be a long list of new features”. However, at the same time, the customer CEO thought they were just the initial features: “We have four years of features and wishes waiting to be developed”. These misinterpretations hindered software development (J-I).

Offshoring increased the complexity of the EDNs by bringing more organizations to them, and it introduced additional operational problems. The customer and the vendor did not always establish the necessary channels or methods for communication. For example, the offshore developers felt they did not have enough information about the business environment to avoid “reinventing the wheel” (F-J). Thus, the customer could not fully realize the system’s potential because the subcontractors could not provide novel ideas to the vendor. Knowledge exchange problems accumulated as new organizations joined in, which hindered software development further (J-I).

According to the vendor in case A, coordinating the offshored development efforts increased the managerial activities because the EDN now had an additional layer (F-H). Offshoring increased the need for testing because the customer found more bugs in the code (F-I). Offshoring also drastically slowed down the software development process as a customer representative in case C pointed out: “It takes about two to three weeks [after if I create a service ticket]…. [Someone from India] starts calling, and then it is messed up.” (Case C, customer, development manager 1).

In case B, due to the vendor’s familiarity with the customer’s business, the vendor did not need complete specifications before offshoring. Later, the offshored department that joined the EDN found these limited specifications to be inadequate. The specification work increased (F-I):

> We got a requirement definition document from the customer, or actually a list of features that they wished to have. I examined the specifications we were actually going to do, and I described to India how it should be implemented. (Case B, vendor, lead designer)

### 4.2.2 Organizational Rearrangements

Organizational rearrangements hindered communication and software development. They also introduced differing objectives, which again introduced project management problems (Figure 9).

The customers faced problems due to organizational rearrangements. In all three EDNs, the ERP system under development was global and was supposed to be deployed in new business locations in other countries as the business kept growing. In cases A and C, the customer did not fully consider the expansion when planning the system: “When [we started the project in 2003], we didn't have any idea that [another geographic area] would be also involved.” (Case C, customer, team manager).

Multinational cooperation also increased communication challenges because it required two-way knowledge transfer with customers. The customer had to retrieve other countries’ needs, and [who?] had to transfer the system’s business logic to these locations. However, the IT departments in all cases were almost entirely situated in a single country, but they still required information from all parts of their respective organizations. The vendor’s lack of skills in knowledge sharing and foreign languages limited this process (G-J):
Our people...are not professional teachers. They may be very familiar with the system and understand it well, but transferring that information to others is another story. Someone has worked on this for over ten years and should then, in a matter of months, transfer the knowledge and understanding to someone else on the other side of the world in a foreign language. (Case A, vendor, service owner)

Figure 9. Influences of Organizational Rearrangements

Geographic distance made organizing rollouts and supportive operations difficult. In addition, the joining organizations sometimes came with differing objectives that hindered the development. For example, in case C, as the project expanded to a new business location, a local manager at the customer's international site resisted the system implementation (G-D). The project lacked managerial support and was eventually halted (D-H). Similarly, case A underwent a merger during the project. The merger changed the power relationships, and a functional area was assigned to a different leader. Competing systems and differing objectives emerged (G-D). As a result, the customer took certain functionalities away from the ERP system and replaced them with other systems:

They [logistics] started making separate islands.... They wanted to “freeze” the system to a certain point and include all kinds of additional systems. Now it has been ongoing for ten years.

We have ended up with serious problems. (Case A, Customer, Business IT Negotiator)

This change not only increased the complexity of the ERP system but also led to increased software development costs (D-I).

Some individuals were crucial to the ERP projects because of their experience and tacit knowledge. Their absence created a void that disrupted the project dynamics further or, in the words of service owner 1 of the customer in case C, even “stopped the development like hitting a brick wall” (G-I). In case B, the project manager’s departure just before the pilot implementation hindered the development because he took with him important know-how and communication channels (G-J). In addition, the project manager had acted as a bridge linking the customer’s business with the IT department and with the vendor. To overcome the missing link, the stakeholders had to recreate these channels and come up with new, more rigorous documentation practices to avoid losing any more knowledge. Organizational rearrangements also took place in terms of changes in job positions, which caused the connections with important stakeholders to break down as occurred in case C (G-J): “The co-determinations heavily affected [the network of specialists, coordinators, and super-users]; they practically killed [the network].” (Case C, customer, system specialist).

4.3 Explaining the Challenges in EDNs

Based on the characteristics of the EDN challenges and the directions of the associations between them (as Figure 2 presents), we categorized the EDN challenges into three types: consequences, root causes, and triggers. Figure 10 presents our explanation for EDN challenges by describing the interrelations of their different types. We identified the operational problems (project management, software development, and communication problems) as consequences because these problems mainly resulted from changes in EDN structure and relationship conflicts. The relationship conflicts (difficulties with third parties, differing
objectives, blind trust, inefficient customer-vendor collaboration, and power imbalance) are the fundamental issues (i.e., root causes) behind the operational problems. Root causes can either reinforce or cause operational problems (arrow 1 in Figure 10). Changes in the EDN structure (joining and departing organizations and organizational rearrangements) are triggers that affect the root causes and consequences. The triggers may generate new issues directly at the operational level of the development (arrow 2) or indirectly make the root causes more severe (arrow 3).

Challenges in ERP development networks

Consequences refer to operational problems that are noticeable in day-to-day development work. They result from other types of EDN challenges (triggers and root causes). We found that these consequences did not lead to new EDN challenges; that is, they would become triggers or root causes. However, they can be related to each other (e.g., poor communication hampered software development).

Relationship conflicts are root causes. The conflicts are fundamental issues behind the operational problems in ERP development. In practice, the root causes may not be as visible challenges as the consequences. Instead, root causes and consequences may reinforce either each other or can make the operational problems more severe. Root causes may evolve over time without any direct effect by another EDN challenge. For instance, inefficient collaboration between the customer and the vendor in case A drifted subconsciously to a situation in which the vendor especially did not consider collaboration as efficient as before. Furthermore, one-sided decision making that resulted from organizations’ drifting apart caused difficulties in software development when the vendor could not contribute to planning the new system features. Blind trust between customer and vendor that grew over time exemplifies another root cause. When stakeholders blindly trusted each other, they took each other’s competencies for granted and bypassed formal cooperation methods, which led to misaligned development needs and overlapping development activities. However, differing objectives and difficulties with third parties emerged sometimes after an organization joined an EDN. In other words, changes in the EDN’s structure triggered these issues, which further caused operational problems and indicates that root causes do not necessarily emerge over time but that changes in EDN structure can trigger them.

We considered changes in EDN structure (i.e., events that altered an EDN in terms of rearrangements in organizations and individual persons’ job positions) as triggers. New organizations joining the EDNs and rearrangements in the existing organizations directly caused operational problems. As an example, we found that the temporary involvement of third-party organizations or personnel losses immediately placed certain pressures on existing documentation standards and communication methods. Likewise, structural changes in an organization (e.g., when a vendor outsourced their operations) introduced problems in project management and resourcing. However, triggers also caused operational problems indirectly by introducing, for example, differing objectives. In case A, organizational rearrangement enabled more independent decision making by different units and introduced differing objectives. This rearrangement increased complexity and software development costs. Similarly, in case C, the absence of a committed local project manager in another country halted the project completely. The main difference between the
triggers and the root causes is that triggers are onetime events while root causes have more permanent characteristics. In addition, triggers illustrate that organizations cannot necessarily control changes in EDN structure. Unlike root causes, triggers do not result from other types of EDN challenges but initiate other types of EDN challenges instead. To sum up, triggers cause operational problems (consequences) directly or indirectly by introducing relationship conflicts (root causes).

5 Discussion

This study contributes to the ERP and IS research. The study approaches ERP development and its challenges from the EDN perspective, an area that has not gained much attention according to our literature review. Although the three cases were unique and from different contexts, all three EDNs featured the 10 challenges. The study also illustrates how EDN challenges appear and relate to each other in practice and explains why they occur. Our main contribution involves identifying and linking changes in EDN structure, relationship conflicts, and operational problems (namely, triggers, root causes, and consequences), which makes it easier to understand ERP development and the reasons for its problems in complex organizational settings. Next, we discuss the EDN challenges in relation to the current literature.

The literature has established the criticality of key personnel in the ERP project teams (Finney & Corbett, 2007; Ngai et al., 2008; Shaul & Tauber, 2013). However, studies have paid less attention to other changes in the EDN structure in terms of joining and departing organizations or organizational rearrangements. These changes have an impact on the relationships in the EDN and can even end them. Research has identified the lack of close relationships between stakeholders as a challenge (Aloini et al., 2007; Ngai et al., 2008; Shaul & Tauber, 2013). Research results show that close relationships are not always possible due to their temporal nature. Existing development practices may not be suitable for new organizations as occurred with offshore developers who needed more rigorous specifications in case B. Our analysis also revealed that changes in EDN structure triggered other types of challenges; that is, it had an influence on the relationship conflicts and operational problems. Moreover, changes in EDN structure could set off a chain of consecutive effects that seriously hindered the development activities.

Research has also considered vendors’ instability and underperformance as challenges (Aloini et al., 2007; Shaul & Tauber, 2013). We suggest that such challenges may actually result from vendors’ dual objectives. In our study cases, these challenges caused significant problems because the vendors had insufficient resources to develop a general product and a customized system. When a stakeholder in an EDN has an agenda that contradicts others, the contradiction can create major issues (Momoh et al., 2010; Nah et al., 2001); however, areas such as interdepartmental communication or project management may eventually reflect the repercussions (Akermans & van Helden, 2002). Such differing objectives and other EDN relationship-related issues are the root causes of operational problems.

Because ERP development requires many organizations to cooperate, one needs to consider the customer’s relationships with vendors and consultants (Dittrich et al., 2009; Sammon & Adam, 2002). Our findings indicated that incompatible working practices between organizations can seriously hamper cooperation. Distrust between partners combined with external parties’ lack of industry competence (Al-Mashari, Al-Mudimigh, & Zairi, 2003) were not major obstacles to successful cooperation according to our analysis. However, customers’ trusting vendors too much led to problems. Moreover, third parties can establish important relationships among themselves and, thus, complicate an EDN’s structure and power relations. Such tensions in interorganizational relationships are also root causes.

In the study cases, many issues that hindered ERP development in practice concerned software development, project management, and communication—issues that the literature also emphasizes (Dezdar & Sulaiman, 2009; Momoh et al., 2010; Shaul & Tauber, 2013). We also found that vendors were responsible for project management problems by, for example, being unable to create detailed plans for system development. In particular, offshoring, which the vendor organizations mainly organized, caused challenges in all cases, such as slowing down the realization of new requirements and complicating the EDN by bringing additional organizations. Offshoring stressed the need for distributing domain knowledge through the EDNs, which research considers problematic (Aloini et al., 2007; Desouza et al., 2006; Nidhra et al., 2013).

The literature covers individual EDN challenges well. Previous ERP studies have not, however, shed much light on the relationships between them (King & Burgess, 2006; Ram & Corkindale, 2014). Instead, they have focused primarily on the consequences of the problems, which, according to our analysis, are
caused by root causes and triggers. Only a few studies have investigated the connections between critical success factors or challenges in ERP development. Akkermans and van Helden (2002) found that the critical success factors were interrelated in a way that a change in one could influence all the others either directly or indirectly. Similarly, King and Burgess (2006) constructed a dynamic model of ERP success that shows the interconnections between organizational context, supporters, and project organizations and explains the vicious and virtuous cycles of ERP development. Our findings support the idea that the challenges in ERP development are more a combination of issues than the cause of any individual problem. Similarly, one should not view EDN challenges in isolation because they are linked. Our findings extend the current understanding of the interrelations of ERP challenges (Akkermans & van Helden, 2002; Bansal & Agarwal, 2015; King & Burgess, 2006). We more profoundly explain the interactions between the organizations in the EDN instead of examining the development from a single organization’s viewpoint. In addition, our findings reveal the interrelations between changes in EDN structure, relationship conflicts, and operational problems.

5.1 Implications for Practice

Although organizations that undertake ERP projects cannot easily avoid ERP development challenges related to changes in EDN structure, organizations should be aware of their possible consequences. Practitioners should prepare themselves for these changes by developing protective measures, such as efficient knowledge-management practices and training programs. In other words, companies should continuously evaluate whether the existing practices suit new forms of the EDN and whether they need new tools and practices. The differing objectives of the EDN organizations or individuals, however, are often a matter of attitudes and personal characteristics, which makes it difficult for organizations to prepare themselves for these issues. However, the organizational culture and leadership could be vital when mitigating these risks.

We argue that one organizations need to distinguish whether the challenges are triggers, root causes, or consequences. An ideal, challenge-free situation is unrealistic because EDNs constantly change. Therefore, it is more important to acknowledge the relatedness of the challenges in a network than to fix its individual problems. By recognizing different types of challenges, organizations can focus on sources (root causes) instead of symptoms (consequences). For example, putting more effort and resources into project management or communication tools and practices will not always provide optimal results. Instead, it would be more efficient to address the root causes behind the problems (e.g., by ensuring that the organizational objectives align with each other).

Both the customer and other organizations in a network may find being aware of the different types of EDN challenges and their interrelations as useful. Furthermore, our findings apply not only to the ERP context but also to any development activity conducted in networks of multiple stakeholders with diverse objectives and expertise, such as product development and innovation networks.

5.2 Limitations

This study has several limitations. We covered only three EDNs from different fields of business and similar geographic areas. However, the networks were independent and disjointed from each other, which ensures at least some level of confidence because the EDN challenges were very similar. A possible limitation concerning the quality of the data is the long lifecycle of the system development in case A. Some of the events occurred many years ago, which may have changed the interviewees’ insights and opinions. Furthermore, due to practical limitations, we interviewed only the customer personnel in case C. The viewpoints of other EDN organizations in case C remained unrevealed.

Despite the long-term relationships in these networks, there was a single dominant vendor in each network. Some of the challenges might have been different if, for example, the cooperation was short-term or there were multiple equal vendors. Moreover, all the cases developed a custom system. When implementing SAP or a Microsoft system, for example, the ERP flagship organization is also involved in the EDN, which makes it more complex. The role of the vendor organization may not be the same, but we believe that our findings are also usable for an EDN that implements a packaged ERP or adopts cloud ERPs. For example, Chiasson and Green (2007, p. 553) suggest: “the differences between packaged software and customized development are…of degree, not kind”.

Finally, one should not dismiss the context. The networks in this study came from similar cultural environments that one can consider democratic in terms of coordination. The organizations (internally and
with each other) emphasized trust more than different legal agreements. Thus, one should apply our findings with caution in contexts where legal contracts define and drive development. However, legislative agreements may remove flexibility in the development activities and increase rigidity, which may introduce new challenges and alter existing ones.

6 Conclusions

In this paper, we explain why EDN challenges occur by first identifying them and then finding the associations between them. We uncovered 10 EDN challenges by analyzing 45 interviews in three EDNs. The findings deepen our understanding of the development of information systems in networks of organizations by explaining how the challenges occur in practice. Further analysis illustrated the types and associations between the EDN challenges through which we pinpointed the original sources of the problems and the situations that initiated the problems. We classify three types of EDN challenges: root causes, triggers, and consequences. Root causes are fundamental issues related to relationship conflicts leading up to, or at least reinforcing, consequences. Consequences are operational problems, such as project management and communication issues, which are also often mentioned in the literature. Triggers are events that change the EDN structure. They can either generate or reinforce the challenges in the other categories. One should recognize the interrelated nature of EDN challenges and their potential to generate consecutive chains instead of viewing the challenges as separate issues. One may also consider the types of challenges may as theoretical categories (Lee & Baskerville, 2003). Distinguishing between triggers, root causes, and consequences allows practitioners to focus their resources on the sources of the problems instead of the symptoms.

In the future, we aim to validate our findings in different contexts and with different research methods. We aim to understand how customers and vendors construct and adapt the collaborative methods they use to organize development and how such methods emerge and develop. Future research could also examine the use of agile methods in EDNs. Overall, EDNs should be studied more thoroughly. For instance, we need research that investigates the evolution of EDNs in more detail to understand the reasons that cause an EDN's structure to change.

Acknowledgments

This research was funded by Academy of Finland, grants #259831 and #259454.
References


About the Authors

Tommi Kähkönen is a PhD candidate at Lappeenranta University of Technology. He is soon about to defend his doctoral thesis titled "Understanding and Managing Enterprise Systems Integration". He is interested in organizational issues in integration and enterprise systems development as well as evolution and management of complex digital environments. At the moment, he is working as a software developer in Enevo Oy and implementing mobile software tools for intelligent waste management in smart cities.

Aki Alanne is a doctoral student at Tampere University of Technology. His research has focused on knowledge management and knowledge sharing in different settings as well as organizational issues in information systems development. Currently, he is working as a business consultant focusing on digital strategies and digital transformation acceleration.

Samuli Pekkola, PhD, is Professor of information systems at Tampere University of Technology and Adjunct professor at University of Oulu. He has worked as visiting associate professor in University of Agder, and held several positions in University of Jyväskylä. His research focuses on users in different manifestations of information systems, IS management and acquisition, and enterprise architectures. His research articles have appeared in journals such as Information Systems Journal, Scandinavian Journal of Information Systems, Enterprise Information Systems, Enterprise Information Management, Decision Support Systems, and the DATA BASE, among others. He is Associate Editor in Business Information Systems and Engineering, and a member of advisory board and past Editor-in-Chief of Scandinavian Journal of Information Systems.

Kari Smolander is Professor of Software Engineering in Department of Computer Science, Aalto University and in School of Business and Management, Lappeenranta University of Technology, Finland. He has a PhD (2003) in Computer Science from Lappeenranta University of Technology and a Licentiate (1993) and Master (1988) degree from University of Jyväskylä, Finland. He has more than 140 refereed research papers in international journals and conferences. His current research interests include change in software and systems development practices and software development organizations.