14 KNOWLEDGE DIFFUSION IN ERP DEVELOPMENT: The Case of Open Source ERP Downloads

Björn Johansson
Center for Applied ICT
Copenhagen Business School
Frederiksberg, Denmark

Abstract *This paper reports on an investigation of challenges in enterprise resource planning systems (ERPs) development. The investigation, conducted as interviews with executives at a major ERP software vendor, identified six challenges when developing future ERPs. The challenges are then related to a question of knowledge sharing in ERP development. The question is, can downloads of open source ERPs be seen as a knowledge sharing activity with the potential to decrease the gap between ERP developers and users of ERPs? From identified challenges and by discussing reasons for the high attention and the high numbers of download of open source ERPs, the article presents some conclusions that could act as input for future research. The paper aims at building a foundation for the basic question: In what way could knowledge sharing in ERP development be improved? The main conclusion is that challenges for future development of ERPs addressed by proprietary ERP software vendors could be one reason for the high attention among developers of open source ERPs.*

Keywords Enterprise resource planning systems, ERP, knowledge sharing, open source ERP, systems development

1 INTRODUCTION

In the past decade, we have seen an explosive of interest in enterprise resource planning systems (ERPs) as well as in free and open source software (FOSS) systems.

Please use the following format when citing this chapter:

Johansson, B., 2008, in IFIP International Federation for Information Processing, Volume 287, Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion, eds. León, G., Bernardos, A., Casar, J., Kautz, K., and DeGross, J. (Boston: Springer), pp. 247-259.

The paper presents some of the challenges in ERP development and relates these challenges to open source ERPs attention. This is interesting at least for two reasons. First, the open source development model has changed during the past few years and the question is, how do the "new" open source development models influence future development of ERPs? Second, software vendors of proprietary ERPs face some challenges, which they need to take care of if they are to be able to stay in business in the future. The question arises of whether or not open source can serve as a useful input for figuring out how to manage future challenges. However, the basic question discussed in the paper is, can downloads of open source ERPs be seen as a knowledge sharing activity in ERP development? The paper will discuss this issue by comparing it to the reasons an individual software developer works in an open source development project and relate that to the challenges an ERP software vendor suggests future ERP developers will face.

The rest of the paper is organized in the following way: Section 2 introduces the subject by defining enterprise resource planning systems (ERPs). Section 3 then describes challenges identified and the research method used to identify them. Section 4 discusses open source ERPs and describes its status using data about open source ERP downloads. The penultimate section discusses reasons developers could have for downloading open source ERPs by relating these to the identified challenges. The final section of the paper offers suggestions for future research related to the discussion about knowledge sharing in ERP development.

2 ENTERPRISE RESOURCE PLANNING SYSTEMS

Enterprise resource planning systems have received a lot of attention in the past few years and have generated a number of reviews (e.g., Botta-Genoulaz et al. 2005; Esteves and Pastor 2001; Esteves and Bohorquez 2007; Shehab et al. 2004). ERPs are often defined as standardized packaged software designed with the aim of integrating the entire value chain in an organization (Lengnick-Hall et al. 2004; Rolland and Prakash 2000). Wier et al. (2007) argue that ERPs aim to integrate business processes and information and communications technologies into a synchronized suite of procedures, applications, and metrics that transcend the boundaries of the firm. Kumar and Van Hillegersberg (2000) say ERPs originated in the manufacturing industry, where the first generation of ERPs was introduced. Development of first-generation ERPs was an inside-out process, going from standard inventory control packages, to material requirements planning, material resources planning, and then further on expanding it to a software package aiming at supporting the entire organization (second-generation ERPs). According to Møller (2005), this evolved software package, labeled ERP II, could be described as the next generation enterprise systems. This evolution has increased the complexity both of ERP usage and development. The complexity comes from the fact that ERPs are systems meant to perform both interorganizational as well as intra-organizational integration, as well as integrating all business processes into a one-suite package (Koch 2001). ERPs, as well as the way organizations use them, have changed significantly. These changes also influence the way ERPs are developed and sold. The ERP market is prone to change, impacting not only the stakeholders in the ERP value-chain (Ifinedo and Nahar 2007; Somers and Nelson 2004), but also how these stakeholders gain competitive advantage. An organization no longer receives competitive advantage by just implementing an ERP (Karimi et al. 2007; Kocakulah et al. 2006), one reason being that ERPs focus on best practices. However, from the vendors' perspective, finding the "right" requirements and implementing these could be seen as input to competitive advantage. Downloads of open source ERPs could be an activity in which developers participate in order to increase their own knowledge about ERP development and thereby gain competitive advantage.

When reviewing the existing literature on ERPs, a major part of the research was found to be on implementation, and the main problem presented was the misfit between ERP functionality and business requirements. Soh et al. (2000) describe this as a common problem when adopting software package. The problem of "misfit" means that there is a gap between the functionality offered by the package and the functionality required by the adopting organization, a definite knowledge gap caused by the weak knowledge sharing in ERP development projects. Can open source ERPs solve this problem? The next section describes challenges identified from interviews with executives in a software vendor organization, building the base for our later discussion on the ways open source ERP development projects could be seen as a knowledge sharing activity.

3 CHALLENGES IN ERP DEVELOPMENT

This section reports briefly on an interview study done with eight executives of an ERP software vendor organization. The interviews were conducted at the vendor's offices. The interviewees have different roles, focusing on the different ERP products in the vendor's portfolio. The eight interviews lasted between 1.5 and 2 hours. The respondents were asked to describe what they see as potential problems when developing future ERPs. Follow-up questions were asked but no specific questions were prepared beforehand. The participating researchers' notes were collected and these notes acted were used to generate a summary of potential problems. The summary was analyzed for patterns, with the patterns identifying a number of problems raised by the respondents during the interviews. The analysis, which could be described as a content analysis (Krippendorff 2004), was done by asking the following questions: What problem (or problems) is the respondents talking about? At what problem was the solution described by the respondent aimed? These questions were asked because the respondents in their descriptions often referred to problems or solutions not yet completed. The analysis also asked: What could we learn from the problem described as well as from the solution? The objective of the interviews was to identify challenges in the development of future ERPs. Six areas describing challenges in ERP development were identified.

3.1 Challenge Area 1: Requirements Gathering

One of the major concerns when developing future ERPs is the time from feature identification to implementation. This challenge indicates that there is a need for an improved process for requirements management. This process should deal with the entire chain from identification of new requirements by the end-users to presenting them to the developers so they can be implemented shortly after identification. This has a clear connection to scalability and flexibility of the ERP system. Based on this, the basic problem is that it takes too long from identification of a new feature to its implementation. Another problem is whether the implemented requirements are the "correct" requirements. A summary of this challenge area is that there are deficiencies in the process for requirements management in ERP development.

3.2 Challenge Area 2: ERP Implementation

Another issue in the development of ERPs is the trade-off between a standard product and a customizable product. This is closely related to questions regarding implementation and the need for an improved process for implementation. There is also a need to clarify the relation between customization and the costs for maintenance of ERPs that are highly customized. In other words, there is a need for describing how customization should be done, if it is done, so that new upgrades can be implemented without the work needing to be redone. The challenge of improving the process of implementation is closely related to the challenge of improving requirements gathering. In both cases, time is crucial and a basic problem is that it takes too long. There are several reasons implementation takes too long, including that the system is too complex to implement, and that the people doing the implementation lack project management expertise. A possible solution is to develop software that is easier to implement, solving both the time problem and the lack of project management expertise. A summary of this challenge area is that there are deficiencies in the ERP implementation process.

3.3 Challenge Area 3: Customization Versus Configuration

Related to the implementation challenge is the question of customization versus configuration. Worth mentioning here is that there are different definitions of these terms, implying different approaches to dealing with the problems. There is a need for a clearer view on customization and configuration and their relation to upgrades and new versions. To some extent, this area was addressed above. However, that discussion dealt with customization and configuration mainly from the perspective of time. Here, the basic problem is lack of knowledge on how these system changes should be done. There is also a question about the basic architecture and how that allows for customization and configuration, which directly relats to adjusting the ERP to the business processes in the adopting organization. Also, how does customization versus configuration influence future costs of ERPs usage? However, a fundamental question—whether the organization should adjust the ERP according to its business processes or the organization should adjust its business processes to the ERP—remains. A summary of this challenge area is that it is unclear how customization and configuration relate to upgrades.

3.4 Challenge Area 4: ERP Architecture

A challenge for future ERPs is emphasizing a software architecture that supports scalability and flexibility. Scalability and flexibility need to be supported in future ERPs because there is a need to provide support for new business models and new demands from end-users without disturbing the ongoing business. Better support for scalability and flexibility would improve the organization's ability to integrate different systems, both internally and external partners. This indicates that interoperability is an important feature for future ERPs. That the basic problem with existing ERPs is that they act too much as closed boxes; there is a need for improvement of the architecture so that it better supports interoperability. Low interoperability indicates that it is hard to connect other systems with the ERP. The answer to this problem could be to develop the future ERPs more as autonomous entities. If doing so, it is important that the relations between the different autonomous entities are clearly defined. A summary of this challenge area is that the architecture of existing ERPs is inadequate, resulting in a weak support of scalability and flexibility.

3.5 Challenge Area 5: ERP and Support of Business Processes

Existing ERPs try to focus on business processes, but they do not explicitly describe the business processes supported. There is insufficient representation of business processes in existing ERPs. Future ERPs should more clearly be built on a business process notation. The basic problem stemming from the current deficiency is that ERPs are not easily adjustable to changes in business processes. The ability to quickly adjust business processes to new demands will increase in importance in the future. The difficulty of adjusting ERPs to changes in business processes currently is that ERPs are too much a "black box," making it hard to know what to change and how to change it. A summary of this challenge area is that there are insufficient representations of organizational business processes in existing ERPs.

3.6 Challenge Area 6: Variations in ERP Requirements

The huge variation among customers, industries, and countries is something that future ERPs need to consider. This variation will influence development both in terms of what requirement the ERP needs to fulfil and what business processes it has to support. The basic problem is the need for ERP flexibility to support the variety of business processes. This is, to a great extent, a problem of *all* standardized software. However, a basic question is, why it is more difficult to manage the variation within ERPs than within, for instance, a word processing package? One possible answer to that question is the range of processes in which the ERP needs to be involved in an organization. In a software application such as MS Word, the information directly used in another software application is specified. In an ERP, the information needs to be able to be transferred between different software applications without human involvement, one reason why ERPs are hard to manage. A summary of this challenge area is that the huge variations in customer requirements that exist when developing future ERPs will increase the complexity of development.

4 OPEN SOURCE SOFTWARE

The origin of open source software development can, according to Hars and Qu (2002), be traced back to the 1950s and 1960s, with macros and utilities freely exchanged in user forums. When Richard Stallman founded the Free Software Foundation (FSF), providing the conceptual foundation for open source software, it really took-off. Open source has matured and become industry-strength in many areas, such as operating systems, application servers, and security tools (Bruce et al. 2006). When it comes to applications, however, open source is not a mature area. Business applications such as customer relationship management, enterprise resource planning, content management, and business intelligence are exceptions from this. Bruce et al. (2006) describe this as the third wave of open source adoption, the first wave being the adoption of open source as operating systems and the second wave being the adoption of open source as infrastructure systems such as middleware, browsers, and databases.

Riehle (2007) states that there are two different types of open source software—community open source and commercial open source—with community open source being software developed by a community and commercial open source software being developed and owned by a for-profit entity. The major difference between them is who decides the future direction of the software. In the community case, individual developers, often in the form of a broad community of volunteers, decides on which contributions will be accepted in the source code base. The community also decides where the software is heading. In the commercial case, a company decides what will be accepted into the software code base, maintains the copyright, and decides what to implement next. There are market-entry barriers to commercial open source. In the community open source situation, no market-entry barriers exist and, therefore, given the right license, anyone can set up a company and start selling software. Riehle states that if someone starts a company delivering open source software, they do not sell the software as such; rather, they sell its provision, maintenance, and support.

Riehle describes costs as one reason why organizations adopt open source. However, he states that the cost perspective is mainly a reason for solution providers. The customer pays for the software he or she uses from a market perspective, the fee is based on the market demand. If the solution provider can produce the software cheaper by using open source, they can increase their profit or decrease the cost for delivered services to each costumer, resulting in solution providers gaining the most from open source software with increased profits through direct cost savings and the ability to reach more customers through improved pricing flexibility. Economides and Katsamakas (2006) stipulate that open source, despite the fact that it can be used for free, has costs related to usage, including learning, installing, and maintaining, which can be higher than for proprietary software.

Costs could be one motivation for why software vendors use open source. What, then, are the motivations for individuals to contribute in open source? Hars and Qu (2002) discuss potential motivations for why individuals participate in open source projects. They state that individuals' motivations can be categorized into two broad areas: internal factors and external rewards. These two categories relate to three factors: social motivations, collective motivations, and reward motivations. Proponents of open source emphasize that open source programmers are not motivated by monetary incentives. Instead, participants are said to be motivated by personal hobbies and preferences or by the rewarding sense from increasing the welfare of others (Hars and Qu 2002). Regarding information support, according to the research of Lakhani and von Hippel (2003), people respond to questions because they expect reciprocity, they "help the cause" (i.e., like the idea of open source software), they expect to gain reputation or enhance their career prospects, they consider answering questions to be intrinsically rewarding, and for some, it is part of their job (if they work in a company that sells a commercial version of the software along with documentation and support).

Although it seems that there are many benefits accruing from usage of open source software, there are indications that not too many companies adopt open source. Goode (2005) conducted research on a sample of 500 of Australia's top companies to determine why their managers rejected open source software. According to his findings, the main reasons are that managers perceive no relevance in its offerings, are concerned about unreliable or transient support sources, and the lack of available resources, or perceive no need for open source technology in their businesses. The smallest companies noted that they did not have time to implement it; they were already using commercial closed source software. This suggests that at least some respondents see the adoption of and migration to open source software as a significant undertaking, with a long and steep learning curve.

Last but not least, there is a question of measurement of an open source software package's market value. When software has no price, no purchase contracts, and no buyers and sellers, it may be difficult to estimate its impact. Thain et al. (2006) addressed this problem and came up with three suggestions: plan for measurement far in advance, use multiple techniques to collect data, and give the users stake in the measurement. This advice is meant mainly for project teams developing open source software and is not easily utilized by researchers. Another approach to estimation of the interest of open source ERP systems is presented next.

There is no doubt that there is a great interest in open source. However, the question remains if this translates to open source ERPs and, if so, why this interest exists. In order to get some kind of answer to these questions, an investigation of the distribution channel SourceForge was made. SourceForge is a distribution channel of open source projects and it is described as "the world's largest Open Source software development web site. SourceForge.net provides free hosting to Open Source software development projects with a centralized resource for managing projects, issues, communications, and code" (www.SourceForge.net). In order to get a grasp of the status of open source ERPs, a search on SourceForge was conducted at the beginning of September 2007 and repeated in November 2007 and January 2008, as shown in Table 1. The search showed that there were 336 open source ERP projects registered on September 13, 2007. On November 27, there were 356 open source projects registered. This means that in 10 weeks there had been an increase of 20 new open source ERP projects. An interesting finding is that the new projects are generally connected to existing projects, meaning they focus on developing ERPs for specific industries. Another way of showing the interest in open source ERPs is to look into downloads of the software as such. However, it is important to remember that downloads do not say anything about actual adoption and usage. It can be assumed, however, that a high number of downloads indicates an interest in open source ERPs. To examine this, downloads of six different open source ERPs were investigated. These open source ERPs were chosen based on the recommendation of a consultant working in the open source ERP sector. Table 1 shows the statistics about downloads from these six open source ERPs projects.

Table 1. Downloads of Six Open Source ERPs

Downloads/ Open Source ERPs	Start of the Project	September 13, 2007	November 27, 2007	Downloads During 10 Weeks	January 23, 2008	Downloads During 8 Weeks
Compiere	2001-06-08	1,267,160	1,296,098	+28,938	1,320,192	+24,094
OpenBravo	2006-03-09	300,716	401,262	+100,546	470,941	+69,679
OpenTaps	2005-08-10	284,429	311,964	+27,535	327,651	+15,687
Facturalux	2006-09-01	231,031	235,897	+4,866	238,753	+2,856
WebERP	2003-01-07	138,064	149,010	+10,946	156,537	+7,527
TinyERP	2005-03-25	6,953	21,116	+14,163	28,729	+7,613

Table 1 implies that open source ERPs have existed for several years. It also shows that the interest seems to have increased, since there have been an extensive number of downloads lately. As mentioned earlier, downloading an ERP does not equal ERP adoption.

An interesting question arises: What are the reasons for the increasing amount of downloads of these specific open source ERPs? Another interesting observation is that OpenBravo has been ranked as number one when it comes to overall downloads at SourceForge for quite some time. It could be interesting to see how OpenBravo relates to the challenges identified by software vendors of proprietary ERPs.

5 DISCUSSION

A major challenge (challenge area 1) identified in ERP development is how finding the correct requirement as well as the time from identification to implementation. This can be compared with discrepancies between the software and organizational practices described by Chiasson and Green (2007), as well as a combination of functionality that is "too far" or "too close." Software can be too far from the specific needs of the organization, thus requiring extensive configuration and development. Software functionality can also be too close, with irrelevant or inappropriate functionality that cannot be modified. Sia and Soh (2007) perceive a software–organization misalignment in two dimensions: institutional theory (imposed or voluntary) and ontological theory (deep structure or surface structure). The majority of imposed–deep misalignments are implemented. Imposed–surface and voluntary–deep misalignments are more often resolved via organizational adaptation and voluntary–surface misalignments are almost always resolved via organizational adaptation. Open source ERP developers may not experience the problem with discrepancies to the same extent as proprietary ERP developers. One reason could be that open source development takes place closer to the using organization, which also could be one reason for downloading of open source ERPs from other developers, as the implemented requirements in open source ERPs may be closer to the specific needs of the using organization.

Boulanger (2005) asks how a disparate loose-knit group of developers can produce for free software that has comparable quality with proprietary software. He describes the feedback loop as one difference between development of proprietary software and open source software that make it possible. The most common approach in a proprietary software development process is the waterfall model, with the development more or less following a clear structure and using a set of five well-defined phases. Boulanger presents the following five phases as a generic structure for proprietary software projects: the requirements phase, the system and software design phase, the implementation and unit testing phase, the integration and system testing phase, and the support and maintenance phase. He says that this structure is an iterative process, but that open source development phases are more intertwined in each other. Suggesting that open source development is more intertwined means that downloading open source ERPs could be to gain inspiration for what requirements to implement in another ERP. The more intertwined development process may decrease the time from identifying a specific requirement to its implementation, since the feedback is more direct.

The implementation challenge (challenge area 2), which basically is how to deal with the implementation process as such, is of interest when analyzing open source ERP downloads. It could be that downloads are made from the perspective of gaining knowledge about how to deal with the implementation. The basic idea in open source is that the end-user of the specific ERP should be able to download and implement the system by him/herself. This means that proprietary developers may download ERPs because they are searching for knowledge about solutions on architectural questions as well as how different architectures influence the implementation process.

Deciding on deploying a specific system includes questions such as usability, compatibility, features, support costs, and software quality and reliability. Boulanger argue that FOSS-developed systems are a viable alternative to proprietary systems when taking software quality and reliability into consideration. The question, then, is do developers download open source ERPs in order to gain insight into dealing with the challenge of customization versus configuration (challenge area 3)? MacCormack et al. (2006) propose that open source and proprietary code differ in modularity, that open source is more modular than proprietary software, suggesting that open source development demands a higher level of modularity. They also state that if this were not the case, the huge number of developers in an open source project would result in software with a high level of problems. The number of developers involved in an open source ERP project, however, is questionable. When looking at the number of developers on different open source ERP projects, there are not so many involved; proprietary ERP projects probably have more developers involved. However, despite the attraction that modularity holds for ERP developers, it could be one reason for downloading open source ERPs. The modularity makes it possible to download and use the knowledge of a specific solution for solving a specific need.

Bonaccorsi and Rossi (2003) explain the peculiarity of the open source movement using recent developments in the theories of collective action, of coordination in the absence of a central authority and of the diffusion of technologies in the presence of network externality. This has made it possible to throw light not only on how the phenomenon arises and finds a vast economic application, but also on the mechanisms that underlie its massive diffusion. However, many questions remain; for example, more in-depth analysis of coordination mechanisms is needed, with particular reference to the precise functioning of open source projects, in order to understand the roles of both project leaders and the minor figures who do the less prominent work. This problem was studied by Crowston et al. (2007), who provided empirical evidence about how FOSS development teams self-organize their work, specifically how tasks are assigned to project team members. "Self-assignment" was the most common mechanism across the three FOSS projects investigated. This could be connected to downloading of open source ERP solutions by developers of other open source ERP solutions. The increase of new open source ERP projects may be a result of self-assignment, and downloads could be seen as knowledge sharing between developers of different open source ERPs.

Von Krogh and Spaeth (2007) state that there has been a huge interest in open source over the past decade. They describe open source as having two distinct features. The first, and what is most probably connected to open source, is software equipped with a license that provides rights to existing and future users to use, inspect, modify, and distribute modified and unmodified software to others. The second feature is the change in the development practice of open source. According to von Krogh and Spaeth, years of development have resulted in a new practice of innovation associated with open source software. The new practice displays a very distinct development process, which means that open source projects often are initiated by a project leader or project entrepreneur. Volunteers then join the project depending on their knowledge. The result of it is that open source projects highlight a change in the nature of how software is developed. Von Krogh and Spaeth argue that the pure top-down, planned software development project has changed into an evolutionary development involving many volunteers.

The question of architecture (challenge area 4) is something that garners a great deal of attention in ERP projects. ERPs need have to have a high level of scalability and flexibility. One way to achieve this is to change the basic architecture in the direction of software-oriented architecture or event-based architecture. Open source ERP projects are the place for developers to test these kinds of solutions, implying that it is also a source of inspiration for other developers in the search of a final solution.

Providing the system with the ability to deal with a huge variation of stakeholders' requirements is a challenge (challenge area 6) with relevance not only for ERPs, but for all software packages. However, it is probably more problematic when developing ERPs since these types of systems also need to take the variations in business processes (challenge area 5) into consideration. Proponents of open source projects often say that the benefit is the closeness to the end-user. If that is the case, it suggests that the interest among developers in open source ERPs is as a source of knowledge on how to develop for a variety of requirements and business processes.

6 CONCLUDING REMARKS AND FUTURE RESEARCH

The interest in ERP development seems to be increasing at the moment. This development is of special interest when it involves closing the gap between delivered functionality and desired functionality. There is also a huge interest in downloading open source ERPs. However, there are indications that downloads not are made by using organizations when it comes to open source ERPs. This interest in downloads could be the result of dissatisfaction with proprietary ERPs, or it could be the result of maturity in the open source phenomenon. However, the most mature open source ERPs do not show the highest number of downloads. One interpretation of this could be that downloads are made from curiosity. Another is that it is a way of increasing knowledge of ERP development, indicating that the downloads are made by developers who may redirect an existing project into a new project or use the knowledge gained by investigating the functionality of the ERP downloaded.

ERPs are complex business applications and their complexity has increased as the software has matured. This complexity, to a large extent, influences development, since the software aims at supporting the entire organization's business processes, requiring that the ERP have a high level of functionality. For developers, knowledge sharing, or making sure that one has the correct knowledge, is important, providing another reason for downloading open source ERPs, at least as long as the downloaded ERP has new knowledge to share. This could explain why OpenBravo is downloaded more often than Compiere, as shown in Table 1. The most innovative open source ERPs are most downloaded, suggesting that ERP developers are searching for new, innovative solutions and using the right to inspect the software to do so.

When looking at the challenges identified by the executives at the software vendor, one must agree with the proponents of open source, who state that these challenges are in focus when developing open source ERPs, indicating that downloads of open source ERPs are of interest to developers of proprietary ERPs. This implies, then, that it would be of interest to compare in depth the development process of open source ERPs with the development of proprietary ERPs.

Riehle (2007) describes the cost perspective as a major reason why software developers adopt open source. However, this can be questioned, and it can be suggested that adoption in the form of downloading of open source is a knowledge sharing activity and a way for developers to get closer to the end-user, thereby closing the gap between delivered functionality and desired functionality. However, the main conclusion from the discussion in the paper is that there seem to be several factors involved in why downloads of open source ERPs are made and it would be interesting to further investigate knowledge sharing as a reason for the attention given open source ERPs. The challenges presented for development of proprietary ERPs could act as an interesting framework for suitable questions to ask in a future investigation about knowledge sharing and knowledge diffusion among ERP developers.

References

Bonaccorsi, A., and Rossi, C. 2003. "Why Open Source Software Can Succeed," *Research Policy* (32:7), pp. 1243-1258.

Botta-Genoulaz, V., Millet, P. A., and Grabot, B. 2005. "A Survey on the Recent Research Literature on ERP Systems," *Computers in Industry* (56:6), pp. 510-522.

Boulanger, A. 2005. "Open-Source Versus Proprietary Software: Is One Nore Reliable and Secure than the Other?," *IBM Systems Journal* (44:2), pp. 239-248.

Bruce, G., Robson, P., and Spaven, R. 2006. "OSS Opportunities in Open Source Software—CRM and OSS Standards," *BT Technology Journal* (24:1), pp. 127-140.

Chiasson, M. W., and Green, L. W. 2007. "Questioning the IT Artefact: User Practices That Can, Could, and Cannot Be Supported in Packaged-Software Designs," *European Journal of Information Systems* (16:5), pp. 542-554.

Crowston, K., Li, Q., Wei, K., Eseryel, U. Y., and Howison, J. 2007. "Self-Organization of Teams for Free/Libre Open Source Software Development," *Information and Software Technology* (49:6), pp. 564-575.

Economides, N., and Katsamakas, E. 2006. "Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry," *Management Science* (52:7), pp. 1057-1071.

Esteves, J., and Bohorquez, V. 2007. "An Updated ERP Systems Annotated Bibliography: 2001–2005," *Communications of the AIS* (19), pp. 386-446.

Esteves, J., and Pastor, J. 2001. "Enterprise Resource Planning Systems Research: An Annotated Bibliography," *Communications of the AIS* (7:8), pp. 1-51.

Goode, S. 2005. "Something for Nothing: Management Rejection of Open Source Software in Australia's Top Firms," *Information & Management* (42:5), pp. 669-681.

Hars, A., and Qu, S. 2002. "Working for Free? Motivations for Participating in Open-Source Projects," *International Journal of Electronic Commerce* (6:3), pp. 25.

Ifinedo, P., and Nahar, N. 2007. "ERP Systems Success: An Empirical Analysis of How Two Organizational Stakeholder Groups Prioritize and Evaluate Relevant Measures," *Enterprise Information Systems* (1:1), pp. 25-48.

Karimi, J., Somers, T. M., and Bhattacherjee, A. 2007. "The Impact of ERP Implementation on Business Process Outcomes: A Factor-Based Study," *Journal of Management Information Systems* (24:1), pp. 101-134.

Kocakulah, M. C., Embry, J. S., and Albin, M. 2006. "Enterprise Resource Planning (ERP): Managing the Paradigm Shift for Success," *International Journal of Information and Operations Management Education* (1:2), pp. 125-139.

Koch, C. 2001. *ERP-systemer—erfaringer, ressourcer, forandringer,* Copenhagen, Denmark: Ingeniøren-bøger.

Krippendorff, K. 2004. *Content Analysis: An Introduction to its Methodology*, Thousand Oaks, CA: Sage Publications.

Kumar, K., and Van Hillegersberg, J. 2000. "ERP Experiences and Evolution," *Communications of the ACM* (43:4), pp. 22-26.

Lakhani, K. R., and von Hippel, E. 2003. "How Open Source Software Works: 'Free' User-to-User Assistance," *Research Policy* (32:6), pp. 923-943.

Lengnick-Hall, C. A., Lengnick-Hall, M. L., and Abdinnour-Helm, S. 2004. "The Role of Social and Intellectual Capital in Achieving Competitive Advantage through Enterprise Resource Planning (ERP) Systems," *Journal of Engineering and Technology Management* (21:4), pp. 307-330.

MacCormack, A., Rusnak, J., and Baldwin, C. Y. 2006. "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," *Management Science* (52:7), pp. 1015-1030.

Møller, C. 2005. "ERP II: A Conceptual Framework for Next-Generation Enterprise Systems?," *Journal of Enterprise Information Management* (18:4), pp. 483-497.

Riehle, D. 2007. "The Economic Motivation of Open Source: Stakeholders Perspectives," *IEEE Computer* (40:4), 25-32.

Rolland, C., and Prakash, N. 2000. "Bridging the Gap Between Organizational Needs and ERP Functionality," *Requirements Engineering* (5:3), pp. 180-193.

Shehab, E. M., Sharp, M. W., Supramaniam, L., and Spedding, T. A. 2004. "Enterprise Resource Planning: An Integrative Review," *Business Process Management Journal* (10:4), pp. 359-386.

Sia, S. K., and Soh, C. 2007. "An Assessment of Package–Organization Misalignment: Institutional and Ontological Structures," *European Journal of Information Systems* (16:5), pp. 568.

Soh, C., Kien, S. S., and Tay-Yap, J. 2000. "Cultural Fits and Misfits: Is ERP a Universal Solution?," *Communications of the ACM* (43:4), pp. 47-51.

Somers, T. M., and Nelson, K. G. 2004. "A Taxonomy of Players and Activities Across the ERP Project Life Cycle," *Information & Management* (41:3), pp. 257-278.

Thain, D., Tannenbaum, T., and Livny, M. 2006. "How to Measure a Large Open-Source Distributed System," *Concurrency and Computation: Practice and Experience* (18:15), pp. 1989-2019.

von Krogh, G., and Spaeth, S. 2007. "The Open Source Software Phenomenon: Characteristics that Promote Research," *Journal of Strategic Information Systems* (165:3), pp. 236-253.

Wier, B., Hunton, J., and HassabElnaby, H. R. 2007. "Enterprise Resource Planning Systems and Non-financial Performance Incentives: The Joint Impact on Corporate Performance," *International Journal of Accounting Information Systems* (8:3), pp. 165-190.

About the Author

Björn Johansson holds a Ph.D. in Information Systems Development from the Faculty of Arts and Sciences at Linköping University. He is currently working on the 3gERP project at the Center for Applied ICT at Copenhagen Business School (http://www.3gERP.org). He is a member of the IFIP Working Group 8.6 (Diffusion, Adoption and Implementation of Information and Communication Technologies), IFIP Working Group 8.9 (Enterprise Information Systems) (alternate member), and the research networks: Knowledge in Organizations (KiO)and VITS Work Practice Development, IT Usage, Coordination and Cooperation. He can be reached by e-mail at bj.caict@cbs.dk.