

Identifying Requirements for Future ERP systems

Björn Johansson
Center for Applied ICT,
Copenhagen Business School,
Denmark
bj.caict@cbs.dk

Niels Bjørn-Andersen
Center for Applied ICT,
Copenhagen Business School,
Denmark
nba.caict@cbs.dk

Abstract. This paper discusses some of the theoretical challenges in identifying the business requirements for a future standard enterprise resource planning (ERP) package capable of supporting small and medium sized enterprises (SMEs), and is part of the 3gERP project¹. Identifying future business requirements is a task that has a high level of complexity and it is likely that the success of the requirements modeling has high impact on the success story for future ERPs. The paper discusses different ways of developing successful models of needed and future business requirements, and ends up with a model that aims at identifying and presenting business requirements.

Keywords. Business Requirements, ERPs, Requirements analysis, Software requirements.

Introduction

From early on the ambition of ERP-systems has been to: 1) integrate all transaction systems within the same system; 2) share common data and practices

¹ The “Third Generation ERP” (3gERP) project is a large collaborative project between Copenhagen Business School, Computer Science at Copenhagen University and Microsoft. The purpose is to establish the academic and market foundation for developing a ‘standard’ flexible and configurable global ERP-system for Small and Medium sized Enterprises (SMEs), which can be implemented and maintained at a fraction of the current costs.

across entire enterprises, and 3) produce relevant information for decision-making purposes in real-time (Nah et al., 2003; Shanks et al., 2003).

The **first ERP-systems**, introduced in the 1980's were tailor made to the individual organizations. They were extensions of accounting systems, and they were very often very costly to maintain and develop. For small and medium sized enterprises (SMEs) one could hardly talk about ERP-systems, since most just used separate systems including accounting systems.

From the early 90'ies we might talk about the **second generation ERP-systems**. The large organizations introduced very large integrated ERP systems like SAP and Oracle, and although these were standard packages, it is characteristic that they required huge tailoring efforts to meet the requirements of the organizations. There are many reports of implementation being ten times the costs of the ERP license.

For SMEs, ERP vendors typically introduced smaller ERP-packages build on accounting systems, which were gradually extended to cover a wide range of other business processes like supply chain, logistics, CRM systems etc. However, even though modifications to suit the needs of the individual organizations were typically much lower than for large companies, the implementation costs have typically been 1 – 2 times the costs of the license for the ERP-package.

Accordingly, for the SME-market, the focus is on developing what could be called the **third generation of Enterprise Resource Planning (ERP) systems**. Contrary to the systems above, the vision is to come up with a much more comprehensive global ERP-system, which can be localized to the different countries/industries/enterprises with a minimum of initial and on-going efforts like the MS Office Package, even though an ERP-system is likely to be vastly more complex due to very different requirements.

The main objective of the third generation ERP (3gERP) project is too produce the research breakthroughs that will **enable the development of a much more comprehensive ERP-system**, which can be localized to the different countries/industries/enterprises with a minimum of efforts like the MS Office Package. More specifically, the idea is to identify the key elements of a **third generation** of ERP systems for SMEs, which compared to existing ERP systems

- may be distributed globally at relatively low costs,
- will be much easier to implement,
- will be easy to localize to different markets/industries/enterprises,
- will be easy to maintain/update when there are changes in the environment on the international, national, industry or enterprise level (Soh et al., 2000)
- will allow for collaboration between enterprises through easy integration with ERP systems of other enterprises in the value chains/value networks (Akkermans et al., 2003), and
- will provide better business insight (data mining) for managing the enterprises.

Obviously, one of the most crucial questions for this to succeed is that we might somehow identify the business requirements of SMEs. But identifying as well as presenting requirements for information systems as such is a difficult task to conduct (Alvarez, 2002). There are several reasons why this is a very complex or maybe even an impossible task.

First of all, ERPs are a type of information systems, which are supposed to address all functions of an organization (Worley et al., 2005), from R&D to customer service, from accounting to business intelligence.

Secondly, the challenge is to develop a standard ERP, which should be applicable not just to one organization but to 'all' organizations. The fact that ERPs are not developed for one organization itself increases the complexity of the development.

Thirdly, the basic development is done by software vendors that do not have direct contact with the final user of the software. This to some extent increases the complexity in the development, but one might also argue that it 'shelters the software vendor from the true complexity of the 'real world', and leave these problems to the implementer, who is either faced with modifying the software or simply modifying the organization. This does not differ from other software packages such as for instance office packages.

Fourthly, there is often a "gap" in the understanding of the importance and the use of requirements between the different stakeholders. Schindler (2007) states that requirements in information systems development often are vague and subject to interpretation. The question is then how the gap that often exist between what the business analysts describe as requirements and what the developer want to have when it comes to a more detailed description of requirements can be closed or at least decreased. Is it possible to develop some kind of process for requirements identification, collection and presentation that makes that the development of the future ERP develop better into the direction of what the end users want to have when it comes to functionality received from the system? The question for this paper is: how to capture business requirements when developing "the" future standard ERP package?

The rest of the paper is structured as follows: Firstly we shall briefly discuss why there is a misfit between many ERP-systems and organizations. Secondly we shall attempt an investigation of some of the many attempts at identifying requirements in the literature. Even though in principle it is possible to distinguish between identifying, describing and representing business requirements, this is often not possible in practice and we shall not do it here. Instead, we are discussing these attempts one by one, more or less in chronological order. Thirdly, we present a conceptual model for how to do requirements identification, collection and presentation. Finally there is a short conclusion pointing out some of the future challenges.

Misfits of ERP-systems

One way to describe business requirements on future ERPs could be to relate them to types of misfits between ERP functionality and business requirements as described by Soh et al. (2000). They group the observed misfits in the three broad categories: data, process, and output. The misfits in **data** arise from incompatibilities between business requirements and the ERP package regarding the data format as well as the relationships among entities in the underlying data model, which could be compared to the architecture of the specific software.

Process misfits are described as functional misfits which could be said are about three things. First access misfits, that means the user does not have access to the functionality they need or that the organization lacks enough licenses for the used software. If it is a lack of licenses the customer organization can relatively easy negotiate with the vendor for additional licenses. But if it is about lack of access to a specific function the customer organization also needs to negotiate with the vendor of maybe buying this functionality or alternatively developing this functionality by themselves or by the help of a solution provider. Second, control misfits which mean that the ERPs source code does not allow the introduction of validation routines without changing the source code. Third, operational misfits, is present when the ERP does not support normal operational steps or the support are inappropriate. The process misfits could be compared to the IT-architecture.

Output misfits are described by Soh et al as the most prevalent form of misfit. This misfit is strictly that the user does not get the information they want to have regarding both the presentation as such or the content of the information. It is stated that this to a great extent can be managed by the ERP system's report writer. However, the report writer is often hard to handle and it quite often necessary to have an external solution partner to help the customer create the report they need. Output misfits could then be compared to the architecture of the business and it seems that the reason this misfits occurs is because the ERP does not support the business model and/or the business processes in the organization to the degree that is wanted.

The main misfit problem, however, seems to be between ERP functionality and business requirements, and 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 functionality offered by the package and functionality required from the adopting organization. Askenäs and Westelius (2000) describe this in the following way: "Many people feel that the current ERP system has taken (or been given) a role that hinders or does not support the business processes to the extent desired" (Askenäs & Westelius, 2000 p 433).

A crucial question that Schindler (2007) raises is that the right people should define the requirements. What Schindler claims is that it often is the wrong

people that identify the requirements and then “throw it over the wall”. This metaphor emphasizes on the fact that it often is business analysts that identify the requirements and these requirements are then very weakly presented to the development staff in an ambiguous way. The development staffs then work from the requirements without questioning these and they do not try to influence requirements that they think are wrong. One reason for the problematic situation is that there exists different views of what actually is required, Alvarez (2002) describes this as a conflict that shows up because of a “communication problem” between analyst and client. It can be stated that despite the fact that most vendors, according to Daneva and Wieringa (2006), have their “standard” process for requirements engineering that they use in establishing new ERP projects, which are used indiscriminately. Before going into how to do it, it should be asked what the requirements are about.

Business Requirements for Future ERPs

On the general level, it can be argued that the requirements to some extent are obvious. It is clear that the ERP requirements should aim at supporting the organization so that the solution fosters a future development of the organization. But when the chips are down that is very ambiguous. The question is whether it is possible to be more clear about what the requirements should be about, and if so, how to do that.

In the work of identifying business requirements for future ERPs the point of departure is of importance. Point of departure should in this setting be seen as the analytic level from where the future business requirements are identified. Traditionally, points of departures discussed in the literature over the more than 30 years of information systems research has varied substantially.

Originally, a **functional** analysis provided the starting point for any analysis of information requirements. This was especially evident, since most systems were localized to single functions.

Langefors (1976) suggested that **objectives**, in the form of goals for the organization, such as increased profit, better competitive position, and market share should be the starting point of analysis. If only the ultimate objectives could be defined, it would be possible to break these down into more and more detailed objectives, which eventually would result in a definition of information requirements.

Other researchers (e.g. Bjørn-Andersen, 1974) argued that decisions would be a more revealing way to understanding information requirements. If only one could identify the **top management decisions** (eg choice of products, markets, customers), a modeling of these decisions would reveal information requirements. Later **critical success/failure factors** were identified as possible key starting points for information requirements (Bullen & Rockart, 1981). And again, if the

key success and failure factors were identified, we could use these for identification of requirements.

Experience showed, however, that even though each of these point of departures provided insight into relevant aspects of requirements for an ERP-system, none of them came even close to covering the full story. In order to be able to come up with a comprehensive analysis, there is a need for a much more comprehensive approach for information systems requirement for future ERP-systems. The question is then how to find the business requirements for future ERPs.

Since the early 90'ies, the dominant philosophy for identifying business requirements has been **business processes**, and it can be stated that almost all ERPs today are process-based or at least have the attempt of being process-based. This development is among others based on the seminal work of Davenport (1994), but also the work by A.-W. Scheer in Germany has very strongly influenced the architecture of SAP, and has to a large extent inspired Business Process Modeling. It can also be noted that despite the fact that ERPs have focused on business processes for several years, ERP still enforces the concept of best practices when an organization implement an ERP package and that it is strongly argued that the organization should change its business processes to fit the ERP (Soffer et al., 2005) rather than the opposite. The problem with describing business requirements for ERP as business processes according to Rolland and Prakash (2000) is that it enforces different levels of granularity, and alignment to organizational needs is difficult to achieve. The reason for this is that 1) it is extremely hard to manage the amount of details, and that 2) the requirements are often described in terms of goals and objectives for the organization and not in terms of ERP functionality.

According to Volkoff et al, a major business requirement on ERPs is to make sure that the business process is done in the right way. But it could be questioned whether it should not be reversed. For instance that the ERP should allow employees to do the work as they want and that the system should be accommodating and assist in such a way that the result from the business process becomes correct. This could be compared to a problem-based solution or an opportunity-based solution. It could also be compared to an event driven approach where the ERP should support, such as if it should support diverse functions as trading, accounting, production, human resources (HR), and/or customer relationship management (CRM).

Another way of analyzing the requirements is to look deeper into the **business model** as well as the nature of the business. This also means that requirements could be found or specified at different levels in an organization, such as internal (intra-organizational) or external (inter-organizational). Requirements also differ between what kind of organization the ERP is supposed to support, for instance, service organizations, manufacturing organizations or governmental

organizations. All these different dimensions stipulate that requirements can be found at different levels, and it also indicates that the requirements to a great extent are about integration between different dimensions.

Luo and Strong (2004) state that a key issue in ERP implementation is how to find a match between ERP functionality and the business processes of the organization. In other words, how to identify the requirements and the functionality that the ERP delivers. Rolland and Prakash (2000) identifies four views of ERP functionality: content view, form view, purpose view, and the customizing process view. The framework they suggest from these different views can maybe be used to evaluate ERPs from a functionality perspective, but it could maybe also be used as a way of identifying business requirements on ERPs. The different views are described in the following way by Rolland and Prakash: *“content refers to the knowledge that is included in the representation system, form refers to the structure and notation used, purpose refers to the objective fulfilled by the representation system and the kind of use which it facilitates, customization process refers to the process by which the ERP functionality is customized to meet specific organizational needs”* (Rolland & Prakash, 2000 p 188). Both Rolland and Prakash’s as well as Luo and Strong’s call for the importance of matching the organizations need with the ERP-system functionality make it mandatory to further look into how to identify and present business requirements for “the” future ERP.

One way to solve this could be to use **roles** as the base for finding the requirements. To do so the concept role has to be defined. Microsoft Dynamics define roles in the following way: “A role is a specific grouping of tasks that a persona is responsible for or participates in” (Microsoft Dynamics, 2006 p 8). Personas in this context should then be understood as a representation of a typical view of the people that can occur within an organization defined by the collection of roles they have. This indicates that a persona can have different roles. Accordingly, looking at business requirements from a role-based perspective means that the ERP should have the technical solution to fulfill new business requirements that there has to be a one-point of access for different work roles in an organization.

According to Worley et al. (2005) the concept of role is particular valuable when describing the interaction between information systems and its users. It is of special interest in ERPs since ERP is an information system that is supposed to support all concerned functions of an organization. This highly intertwined demand means that the ERP has to support a lot of different roles in an organization and the work tasks these roles have. Worley et al. furthermore describe roles as a group of functions aiming at achieving a specific purpose, and they say there are four generic classes of roles found in all organizations: Interpersonal, informational, decisional, and operational roles. These roles then have specific needs to the software that is supposed to support them. Worley et al.

conclude that this means that optimization of ERPs therefore is hard to make since it demands both a change in the system as well as a change in the role. The change in role can be said to be to change the way people work and change the process that they conduct their work. This can be compared to the discussion about the relation between information technology (IT) and organizational change suggested by Markus and Robey (1988).

An extension of the role analysis is to look into what roles/composite roles that exist in different organizations as well as the kind of combinations of roles that exists. The roles could also be categorized in task-oriented roles or managerial roles, and one might then think of the tool as a reverse engineering approach of what different access rights there exists. Finally, another direction of looking into roles and how these roles influences business requirements could be to categorize what decisions and what kinds of decisions are made in different organizations.

Many of these approaches build on the assumption that is possible to bridge the gap between the developer and the user by some kind of joint activity. However, when developing a standard ERP-package it is obviously not feasible to have the full story, since to some extent the developer do not know for whom the system is developed. But, many of the approaches hold a potential for contributing to part of the story. Consequently, we shall attempt to draw upon most of these in order to identify a methodology relevant for identifying requirements of standard ERP-systems in the next section.

A model for identifying and presenting ERP business requirements

There seems to be some kind of trade-off between different stakeholders about the level of business requirements. The one that represents the organizational level probably want to have the requirements on a more abstract level, which suggests that requirements could be described such as: “we want the ERP to support our business processes”, or “we want such and such functionality”. These abstract requirements are not on the level that developers with a more technical viewpoint want to have. The developers’ level of requirements is on the level “what data do you want to be able to see on this screen” or “how many decimals should it be possible to store for this information with”.

This trade-off means that there has to be some kind of “translator” between these parties. The question is then what kind of translator there exists or how to do this translation. This directs to the question if there exists some kind of modeling tool that could be of help in doing this, from the discussion so far we suggest narratives and scenarios as tools for finding and presenting business requirements.

According to Hedman and Borell (2004) a narrative might be a communication medium suitable for presentation of ERP requirements since narratives can manage a high degree of uncertainty and ambiguity. The statement they make is that *“In fact, narratives might be the only way to capture invaluable input regarding project size and complexity of an ERP implementation”* (Hedman & Borell, 2004 p 287)

Clausen (1994) describes the use of **narratives** in system development and claims that the approach is “more or less” the solution on the problem with the gap of knowledge between end-users and systems designers. This is also a clear improvement over the “trial and error” strategy that is the most common approach for development and implementation software applications.

According to Nielsen (2004), the term narrative has several meanings and includes descriptions of: the human way of organizing experience, the process of telling, as well as the plot structure of a text. She describes the differences between a narrative and a story in the way that a story includes prior events or events the reader of the story has to assume or guess. The narrative on the other hand is described as an organization of events as a movement from a start point to an end point.

Clausen presents two different kinds of narratives that could be useful at different phases in a system development project. The first is “the history” which is a narrative about the existing situation in the investigated organization and describes the situation of a specific user when conducting a specific work task. To have a “useful” narrative, the users has to express their concepts, and the way to catch that could be to use qualitative interviews. The other kind of narrative is “the scenario”. According to Clausen, scenarios are narratives that designers formulate from the empirical narratives – the history – and these could be described as fictional narratives since these should describe the future. The idea is that the designer should work like a composer and create a narrative which communicates the designer’s vision as well as how the designers have interpreted the user’s situation.

Hedman and Borell (2004) state that the potential of narratives is that they can influence action, by conveying meanings, interpretations and knowledge of the system. The conclusion Hedman and Borell (2004) give from their usage of narratives in evaluation of ERPs, is that narratives provides a richer picture on the evaluation and that this knowledge can change the users’ mental maps and thereby improve the use of the ERP evaluated.

Nielsen and Madsen (2006) use stories and narratives synonymously, and they describe **storytelling** as consisting of the four elements: the perspective of the narrator, the agents controlled actions that aim at a goal, the established and followed sequence, and the sensitivity towards the social practices in a given context. According to Nielsen and Madsen narratives have been used in IT literature in four different ways: 1) looking at political explanations for IT

implementation, 2) providing a basis for systems design and development, 3) serving as a technique for requirements analysis, 4) and illustrating IT project post mortems. In this context we would say that the last three are all interesting to investigate further when it comes to identifying, collecting and classifying business requirements for future ERPs.

It is also interesting that Nielsen and Madsen (2006) describe storytelling as a way of sharing knowledge by externalization of tacit knowledge. The gap identified between users expectations of the future system and what the developer perceive as requirements on the same system could be described as lack of knowledge depending on that the different stakeholders has tacit knowledge. Schindler (2007) describes the problem of how developers understand the business requirements as a lack of knowledge about the business processes that the organizations business builds on. According to Nielsen and Madsen (2006) there are generally speaking two different approaches for sharing one's own experience and that of others. Our understanding of the first, which emphasize the learning experience, is that one learns by trying and that one learns over time by participating in projects. The other approach, according to Nielsen and Madsen, is to learn from documentation of earlier projects. They suggest that documentation in form of storytelling, and in particular oral storytelling, is very useful for building a shared understanding. They also state that this is a suitable approach for making sense of past actions as well as envisioning the future. From the discussion so far we suggest the model shown in Figure 1 as a model for how to capture business requirements of "the" future ERP.

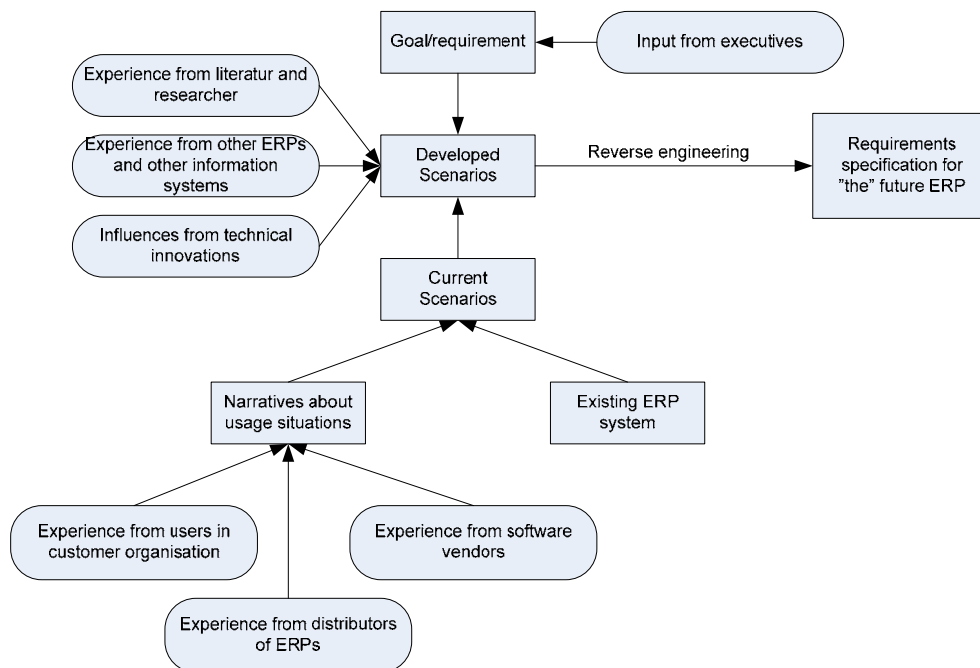


Figure 1 A model for identifying and presenting business requirements for ERPs (based on Jarke et al., 1998 p 157)

The model builds to a great extent on the idea that narratives can describe “real” requirements. The input to these narratives could come from at least three different stakeholders involved in the development of ERPs. By collecting information in the form of qualitative interviews from “end-users”, independent software vendors (acting as developers in the form of customizing the ERPs) it should be possible to develop empirical narratives. These narratives will to some extent depend on the existing ERP system, but it will probably result in some suggestions for functionality that already is present in the existing ERP, but which for one reason or another not are used. This means that the narratives in addition to knowledge from the existing ERP will show potential for change of the existing system. From this information it will be possible to develop scenarios that describe the current situation. The scenarios of the current situation will then be developed into scenarios about the future using input from literature, experience from other ERP installations as well as ideas regarding influence from technical innovations. By relating the developed scenarios to the organizations goals and requirements for fulfilling these goals, the end result should be a better connection between the requirements and the objectives for the organization. The final step is then a reverse engineering from the scenarios in order to achieve a requirements specification for “the” future ERP. The ambition and goal of the model is to be able to present requirements that are on a level that both developers and executives in the user organizations can agree on as useful.

Conclusion

There are huge challenges in developing a global standardized ERP package. These challenges can be described as belonging to three areas: 1) future ERPs need to be a global product with local reach that can be implemented with a minimum of effort, 2) the business landscape has changed and will change over the years to due to globalization and most organizations will become more and more virtual 3) new technical innovations offers huge potential for increasing the benefits of ERP systems, if these are used correctly.

The fact that the future ERP needs to be a global product with local reach means that the ERP needs to deal with issues of different tax rules, regulations, local practices, competitive landscapes, accounting standards and different languages in different countries. This is a huge challenge.

The changing business landscape can be described as a requirement for dealing with issues as different as, e-commerce, demand for vertical as well as industry solutions, supply chain integration/optimization, and ad hoc reporting (Sammon & Adam, 2005).

The technical innovations that influence future ERPs are abundant. Examples of technical innovations that influence future ERP requirements are: enterprise application integration (EAI), extended markup language (XML), service oriented architecture (SOA), software as a service (SaaS) as well as other forms of external hosting. All of these holds promises of vastly different architectures for future ERP-system, but the question remains whether these are feasible?

In this paper we have attempted a first cut at creating an overview of a large number of recommendations for analyzing information requirements. Microsoft has a clear strategy of combining business processes with roles. Business processes constitute the organizational (and inter-organizational) building blocks, within which an individual will be responsible for a number of roles, each consisting of tasks. Our next step will be to analyze whether this combination may be augmented with some of the other analyses mentioned above.

References

- Akkermans, H. A., Bogerd, P., Yücesan, E. and Van Wassenhove, L. N. (2003) The impact of ERP on supply chain management: Exploratory findings from a european delphi study. *European Journal of Operational Research* 146 (2), 284-302.
- Alvarez, R. (2002) Confessions of an information worker: A critical analysis of information requirements discourse. *Information and Organization* 12 (2), 85-107.
- Alvarez, R. and Urla, J. (2002) Tell me a good story: Using narrative analysis to examine information requirements interviews during an ERP implementation. *SIGMIS Database* 33 (1), 38-52.
- Askenäs, L. and Westelius, A. (2000) Five roles of an information system: A social constructionist approach to analyzing the use of ERP systems. In *twenty first international conference on Information systems*, pp 426-434 Association for Information Systems, Brisbane, Queensland, Australia.
- Bjørn-Andersen, N. (1974) *Informationssystemer for beslutningstagning*. Busck, Khvn.
- Bullen, C. & Rockart, J. (1981) A Primer on Critical Success Factors. CISR no 69, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA
- Clausen, H. (1994) Designing computer systems from a human perspective: The use of narratives. *Scandinavian Journal of Information Systems* 6 (2), 43-58.
- Daneva, M. and Wieringa, R. J. (2006) A requirements engineering framework for cross-organizational ERP systems. *Requirements Engineering* 11 (3), 194-204.
- Davenport, T. (1994) *Business Processes*, Wiley
- Dugan, M. A. (2003) "Envisioning" Beyond intractability. (Burgess, G. and Burgess, H., Eds), Conflict Research Consortium, University of Colorado, Boulder.

- Farbey, B., Land, F. F. and Targett, D. (1995) A taxonomy of information systems applications: The benefits of evaluation ladder. *European Journal of Information Systems* 4 (1), 41-50.
- Hedman, J. and Borell, A. (2004) Narratives in ERP systems evaluation. *Journal of Enterprise Information Management* 17 (4), 283-290.
- Jarke, M., Bui, X. T. and Carroll, J. M. (1998) Scenario management: An interdisciplinary approach. *Requirements Engineering* 3 (3), 155-173.
- Luo, W. and Strong, D. M. (2004) A framework for evaluating erp implementation choices. *IEEE Transactions on Engineering Management* 51 (3), 322-333.
- Langefors, B. (1976): *Theoretical Analysis of Information Systems*, Studentlitteratur, Lund
- Markus, M. L. and Robey, D. (1988) Information technology and organizational change: Causal structure in theory and research. *Management Science* 34 (5), 583-598.
- Microsoft Dynamics (2006) Roles-based business productivity "Software designed for your people". pp 1-57.
- Nah, F.-H. F., Zuckweiler, K. M. and Lau, L.-S. J. (2003) ERP implementation: Chief information officers' perceptions of critical success factors. *International Journal of Human-Computer Interaction* 16 (1), 5-22.
- Nielsen, L. (2004) Engaging personas and narrative scenarios. *Informatics*, Copenhagen Business School, Copenhagen.
- Nielsen, L. and Madsen, S. (2006) Using storytelling to reflect on IT projects. *Journal of Information Technology Theory and Application* 7 (4), 35-47.
- Rolland, C. and Prakash, N. (2000) Bridging the gap between organisational needs and erp functionality. *Requirements Engineering* 5 (3), 180-193.
- Sammon, D. and Adam, F. (2005) Towards a model of organisational prerequisites for enterprise-wide systems integration: Examining ERP and data warehousing. *Journal of Enterprise Information Management* 18 (4), 458-470.
- Schindler, E. (2007) Getting clueful: Five things CIOs should know about software requirements. CIO.com.
- Shanks, G., Seddon, P. B. and Willcocks, L. (2003) *Second wave enterprise resource planning systems : Implementing for effectiveness*. Cambridge University Press, New York.
- Soffer, P., Golany, B. and Dori, D. (2005) Aligning an ERP system with enterprise requirements: An object-process based approach. *Computers in Industry* 56 (6), 639-662.
- Soh, C., Kien, S. S. and Tay-Yap, J. (2000) Cultural fits and misfits: Is ERP a universal solution? *Communications of the ACM* 43 (4), 47-51.
- Volkoff, O., Strong, D. M. and Elmes, M. B. (2005) Understanding enterprise systems-enabled integration. *European Journal of Information Systems* 14 (2), 110-120.
- Worley, J. H., Chatha, K. A., Weston, R. H., Aguirre, O. and Grabot, B. (2005) Implementation and optimisation of ERP systems: A better integration of processes, roles, knowledge and user competencies. *Computers in Industry* 56 (6), 620-638.