

Why Focus on Roles when Developing Future ERP Systems

Björn Johansson¹

¹ Copenhagen Business School, Center for Applied ICT, bj.caict@cbs.dk

Abstract. In this paper the question: ‘Why software vendors focus on roles when they aim at developing the future enterprise resource planning (ERP) systems?’ is discussed. At the moment there is increasing interest in developing ERP systems that support a person’s different working roles in the organisation, which could be described as role-based ERPs. It could be asked if this is a reaction to the well-known problem of misalignment between users’ requirements for ERP systems and what functionality ERP systems de facto support the organisation with. The question also relates to a discussion about the impact ERP systems have on users and how they are affected by the implementation, as well as the resistance to change that often is seen in an ERP implementation. One conclusion is that ERPs can have different influencing roles in an organisation, which ranges between increased control and increased agility. This can definitely be seen as an influencing factor for why role-based ERPs are in focus. The main conclusion delivered is that development of future ERPs could benefit from having a role-based perspective but it is necessary to combine this with a business process perspective, if not; the gap between delivered functionality and needed functionality will still exist.

1 Introduction

Enterprise resource planning (ERP) systems had their introduction in the 1950s and 1960s when computers were introduced in organisations (Møller 2005). ERPs are often defined as standardized packaged software designed with the aim of integrating the entire value chain in an organisation (Lengnick-Hall, Lengnick-Hall and Abdinour-Helm 2004; Rolland and Prakash 2000). ERP has its origin in the manufacturing industry where the first generation of ERP systems were introduced (Kumar and Van Hillegersberg 2000). According to Kumar and van Hillegersberg development of the first generation ERP systems were an inside-out process going from standard inventory control (IC) packages, to material requirements planning (MRP), material resources planning (MRP II) and then further on expanding to a software package that aims at supporting the entire organisation (second generation ERPs). This evolved software package is then described as the next generation ERP labelled as

ERP II which according to Møller is the next generation enterprise systems (ES). By stating that it is an ES and not an ERP the focus on the entire organisation and all its business processes is clearer.

ERP as a concept probably had its invention in the 1990s, with the introduction of for instance SAP R/3. SAP R/3 followed SAP R/1 and SAP R/2. SAP R/1 was a financial accounting system where the R stands for real-time processing. R/1 was replaced by R/2 at the end of 1970s. The R/2 introduced the mainframe as a delivery form for the software, and had a focus on large multinational European organisations. This software aimed at supporting their users with real-time business applications in which multi-language and multi-currency capabilities were built in. The difference between R/2 and R/3 is then that R/3 uses the distributed client-server architecture and this change resulted in that R/3 can be managed on multiple platforms and operating systems. R/3 consists of distinct functional modules, and these have a focus on best practices and therefore it more or less forces the adopting organisation to change its business processes (Koch 2001). The evolution of SAP to a great extent also describes the evolution of other ERPs, both when it comes to focus on best practices as well as the structure of functional modules. It also relates to the definitions of ERPs and the ways ERPs are described.

A major problem with existing ERPs is the “misfit” between delivered functionality and needed functionality, described as a gap between the processes the ERP supports and the processes the organisations work by. This misfit has many causes, for example increased costs of customizing and upgrading ERP systems, leading to unwillingness to adapt the system to the organization. In part, the misfit is also due to a poor process of defining exact requirements for the system. This has created an interest among ERP vendors to improve future ERP systems so that the architecture better supports coordination between users’ perceptions of the system and their work tasks and thereby minimizes the misfit between the system and the organisation.

Microsoft (Microsoft Dynamics 2006) and SAP (SAP 2007) have recently focused on “Role-based ERPs”. When SAP describes mySAP and SAP NetWeaver they focus a lot on describing roles and how their ERP aim at being a role-based ERP. The same can be said about Microsoft Dynamics (2007) that describe it as “Role Tailored Business Productivity”, gained by software designed for your people.

The question is then: why do software vendors focus on roles when developing the future ERP systems? This is the question discussed and the aim is to describe and suggest some future research area related to this question.

The rest of the paper is organised in the following way: first, ERPs and the misfit problem are discussed, the discussion is then about how to solve the misfit problem and three different directions for doing this is discussed. After that the paper ends with a concluding discussion that suggest some questions for future research related to development of role-based ERPs and how to close the functionality gap that in the paper was described as the “misfit” problem.

2 Existing ERPs and the Misfit problem

There is a great extent of ERP research such as Shehab, Sharp, Supramaniam, and Spedding (2004), Esteves and Pastor (2001) and Botta-Genoulaz, Millet and Grabot (2005). Reviewing these reports give the impression that a major part of the research is on implementation of ERP systems. It also shows that the main problem presented is the misfit between ERP functionality and business requirements. Soh, Kien and Tay-Yap (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 organisation. 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 desire” (Askenäs and Westelius 2000 p 433). Another way of describing this is as said by Bill Swanton, vice president at AMR research, saying that only 35 per cent of the organisations are satisfied with the ERP they use at the moment, and he says the reason for the dissatisfaction is that the software does not map well with the business goals (Sleeper 2004).

The question is then if it is possible to more clearly describe what this misfit is about. One way could be to use the categories of misfits between ERP functionality and business requirements that are described by Soh et al. (2000), and relate the categories of misfit to architecture of the specific software, IT-architecture and business architecture. Soh et al., group the observed misfits in the three broad categories: data, process, and output.

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 probably could be compared to the data model that is the base for the specific software architecture.

Process misfits are described as functional misfits involving three different dimensions. 1) 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. If it is 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. 2) Control misfits which mean that the ERPs source code does not allow the introduction of validation routines without changing the source code. 3) Third, operational misfits, is present when the ERP does not support normal operational steps or the support are inappropriate.

Output misfits are described by Soh et al as the most prevalent form of misfit. This misfit is strictly about 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 needs to have an external solution partner to help the customer create the report they need. Output misfits could be compared to the business architecture and the reason this misfits

occurs is because the ERP does not support the business model and/or the business processes in the organisation to the degree that is wanted.

The misfits can be related to the fact that when an organisation implements an ERP it has to either change its business processes or customize the software. In most cases what happens is that organisations do a little bit of both, this means that they customize the software to some extent and they change the business processes to some extent. If this is good or bad there are disagreements about, but what can be stated is that it costs a lot of money for the adopting organisation both during the implementation but also during upgrades of new versions of the software in the future (Koch 2005). The duality in changes relates to the main misfit problem described by Soh et al., (2000) as the misfit between ERP functionality and business requirements, they also state that this is a common problem when adopting software package.

One reason for why this problematic situation shows up is that there exist different views of what actually is required. Alvarez (2002) describes this as a conflict from a “communication problem” between business analyst and customer. According to Daneva and Wieringa (2006) most vendors have their “standard” process for requirements engineering that they use in establishing new ERP projects. Luo and Strong (2004) state that a key issue in ERP implementation is how to find a match between ERP functionality and the organizations’ business processes. The question is then how to find the requirements as well as how to find the functionality that the ERP delivers. 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 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 neither do they try to influence requirements that they think is wrong.

Rolland and Prakash (2000) identify 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 and 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.

3 How to solve the misfit problem

The question is then how to solve the misfit problem, and it can be asked if any of the three different ways: 1) better integration between ERPs and the enterprise architecture (EA), 2) development of role-based ERPs, and 3) change users' perception of ERPs, that are presented below could solve the misfit problem.

3.1 Better integration between ERP and EA

One way to discuss this is to look into the concepts enterprise architecture and enterprise architecture integration (EAI) and discuss if EAI and better integration can close the functionality gap. Bucher, Fischer, Kurpjuweit, and Winter (2006) describe enterprise architecture as consisting of five different layers: Business architecture, process architecture, integration architecture, software architecture, and technology or infrastructure architecture. Bucher et al. describe software architecture as the fundamental organisation of software artefacts. However, since ERPs are much more than software artefacts it can be argued that it also influences the integration architecture as well as the rest of the different architectures. According to Bucher et al (2006) ERP represents the integration architecture, the integration of information systems components in the relevant enterprise context. This means that ERPs consist of both the software as well as the integration architecture since ERPs are systems that aim at integrating the entire organization's business processes. This also means that there seems to be a close relationship between ERP systems' basic architecture, business processes and the business architecture. The interesting issue is then to consider how ERPs impact and influence the totality of the architecture. Bucher et al. state by referring to the hierarchical, multi-level systems theory approach, that each architecture layer influence each other. The basic thinking they suggest is that each layer reduces the freedom of action of subsequent layer. This means that ERPs to a great extent integrate the different architecture layers in an organisation. Melin (2003) states that implementation of ERPs are dependent on the architecture of the ERP. He argues that since ERPs more or less builds on the assumption that all data is stored in one "bucket" it suggests a high level of centralisation and standardisation. This means that an organisation when implementing an ERP solution most likely goes for an increase of centralisation and standardisation.

This indicates that ERP is one way to integrate different layers in enterprise architecture. Another possibility to integrate the different layers is according to Lee, Siau and Hong (2003) enterprise application integration (EAI), and they claim that EAI automates the integration process with less effort than required with ERP. Lee et al. describe ERP as an approach that addresses operational integration aiming at supporting daily operations in organisations. Data warehousing is also described as an approach aiming at integrating, but it focuses on informational integration supporting decision-making (Lee et al. 2003). They describe EAI as an approach of integrating existing applications with some kind of middleware, and they label it an externalization approach. The opposite according to Lee et al. is the internalization approach which is made when an organisation implement an ERP solution. They also state that the biggest challenge in integration is not the technical integration instead

they say it is the behavioural integration. It can be argued that by developing an ERP towards being role-based, the behavioural integration would probably be better, and in that way could a role-based ERP also be seen as an externalization approach for integration.

According to Carlsson and Hedman (2004) the development of ERPs has made organisations truly integrate different sources of data, and the ERP vendors have recognized that the client/server architecture did solve the problem with inflexibility that the mainframe architecture had. But, when solving this inflexibility problem another problem with non-integrated client-server solutions surfaced. Carlsson and Hedman state that client-server solutions have resulted in many organisations developing into an incoherent collection of disparate sub-systems. The incoherent collection of disparate subsystems can be seen as one reason for the development of enterprise portals and maybe it is a reason for developing role-based ERPs.

3.2 Development of role-based ERPs

The aim of developing a role-based ERP is according to Sleeper (2004) that the software vendor tries to think in terms of functions and transactions for the overall structure, but through the eyes of a particular person who has a particular responsibility. Sleeper suggests the need for less training as the main benefit with a role-based ERP. He states that one of the major drawbacks of existing ERPs is that they demand a huge amount of training if they should be that useful as they could be. He also states that the role-based ERP could be compared to self-services applications which demands zero training, and claim that if ERPs can be completely intuitive, the organisation will be better off (Sleeper 2004).

One way to solve the misfit problem could be to use **roles** as the base for finding the requirements, by identifying different roles work tasks and how they execute the tasks, but that probably depends on how roles are defined. Microsoft Dynamics defines 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 or has different roles. Looking at business requirements from a role-based perspective means that the ERP should have the technical solution to fulfil new business requirements that organizations have with the feature of having one-point access for different work roles in an organization.

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 be categorized in task-oriented roles or managerial roles, and do a reverse engineering approach of what different access rights there are. 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.

Role-based ERPs can be compared to enterprise portals (EPs), since EPs focuses on designing a specific user interface for a specific role. EPs are defined by Carlsson and Hedman (2004) in the following way: An EP is designed as a single access point

to an organisation's external and internal information and communication technologies (ICTs). They describe EP as one technical solution to fulfil new business requirements that organisations have on ICT with the feature of having one-point access for different work roles in an organisation. According to Carlsson and Hedman an EP differs from other corporate information systems in three ways. First, EP provides the user with a single access point to all computer-based data and information the user needs to fulfil his/hers work tasks. Second, EP provides the user with services and these services does not necessarily have to be strictly connected to the users work tasks. Carlsson and Hedman describes this as self-employee services and gives the example of that employees pay their own bills from the portal they use at their working place. Third, EP gives the possibility to personalization. The last mentioned difference is the one that can be said give the basic thinking of role-based ERPs. Carlsson and Hedman state that SAP introduced the concept of roles to support the implementation of portals. They describe roles as something that determines what information, application, and service a user can access or need to have access to when carrying out its tasks and activities. According to Carlsson and Hedman SAP define roles as: *"a collection of activities that an employee carries out in one or more business scenarios of an organisation. Users access the transactions, reports and web-based applications in a role via a series of menus. Roles are specific to individual employees and match their specific tasks and service/information needs"* (Carlsson and Hedman 2004 p 268).

It could be said that ERPs are process-based or at least have the attempt of being process-based. According to Koch (2001) the basic architecture building on a department/stab model as for instance SAP R/3 makes ERPs not supporting the idea of business processes and thereby not the integration between different departments in an organisation. Koch states that it does not help that the software vendor attached some words about business processes and so on to their ERP products, since the basic architecture does not support business processes.

It can also be said that by focussing on being process-based the characteristics of human agents is missing. The reason for why an organisation wants to have alignment of its business processes, its enterprise systems architecture and its ERP system are several. One reason could definitively be that the organisation wants to have a competitive advantage over its competitor.

According to Worley, Chatha, Weston, Aguirre and Grabot (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 ERPs are information systems supposed to support all concerned functions of an organization. That it is highly intertwined means that ERPs have to support a lot of different roles in an organization and the work tasks these roles have. Worley et al. describes 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 is to change the way people work and change the process that they conduct.

3.3 Change users' perception of ERPs

The above can be compared to the discussion about the relation between information technology (IT) and organizational change suggested by Markus and Robey (1988). Melin (2003) has this as a starting perspective, when he describes the relationship between technology and human agents and by referring to Orlikowski (1992) he states that there exist two different modes, the design mode and the use mode, for the interaction. The design mode means that the user of the technology builds certain interpretive schemes when using the technology. The use mode then means that the user assigns inter-subjective meaning from the usage of the technology. The technology then mediates user activities; at the same time as it constrain performance by facilitating the usage in a particular manner. But, as described by Markus and Robey (1988) technology does not determine social practices, it only conditions social practices according to Melin (2003). Markus and Robey (1988) describe three different directions of what they call causal agency between IT and organisational change: The technical imperative, the organisational imperative, and the emergent perspective. These differ according to the role of IT as a driving force or as a causal agent for the change. The technical imperative views technology as the driving force, the organisational imperative views human agents as the driving force and finally the emergent perspective view the change as a reaction to the interaction between technology and human agents.

Askenäs and Westelius describe five different roles an ERP could have. The description builds on the fact that they see ERPs as having an "actor" role in organisations. The roles are labelled as: Bureaucrat, manipulator, consultant, administrator, and dismissed. The last role is actually not a role and it is described by Askenäs and Westelius as the status when the ERP not is used or is used only partially by some users. This means that the impact of an ERP that has the role "dismissed" is close to zero. However, the other roles influences and impact the organisation and its employees in different ways. Figure 1 describes the direction of impact different roles have according to Askenäs and Westelius. The figure shows the role an ERP can have from two different directions, first, direction of control, and, second, the fit the ERP has with the existing structure of the organisation.

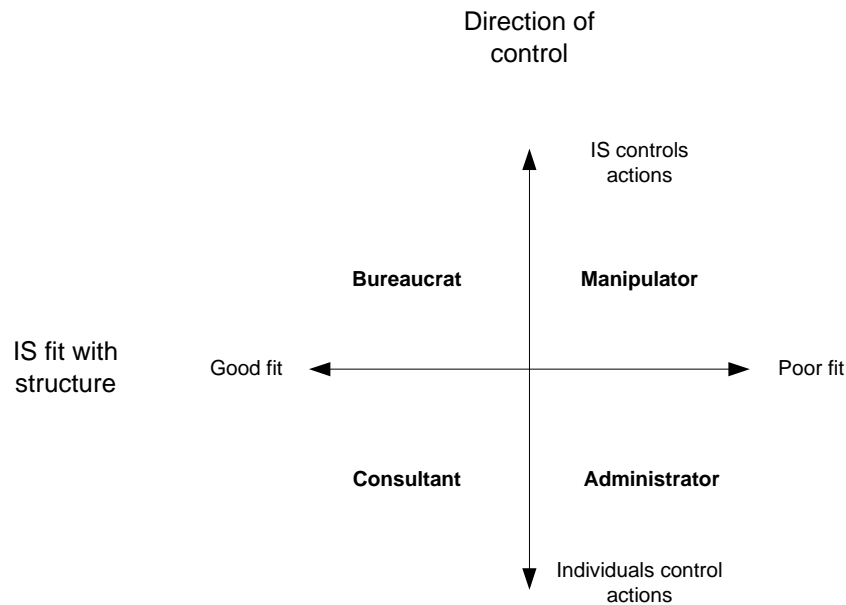


Fig. 1 Four Roles ERP can have (Askenäs and Westelius 2000 p 427).

Staff (2006) describes mySAP as the next generation of ERPs, and says that this software is fast, flexible and an efficient foundation offering organisations new functions, greater productivity, integrated analytical insights into business processes which increase the flexibility to design the software so that it meets customer requirements. This is described as role-specific development aiming at increasing employee productivity by supporting them with a tool tailored to their specific work tasks. This contradicts the statement by Melin (2003), who states implementation of an ERP is a strategy aiming at implementing a highly standardized information system. The statement from Melin is supported by Millman (2004) who says that one of the biggest mistakes an organisation can make is to customize the ERP so that it fits their processes instead of changing their processes to fit the software. However, it could be asked how and if development of role-based ERPs could support increased flexibility at the same time as it suggests standardisation. Millman states that organisations often want to have a customisation of a “bad” process, and since the consultant earns money on customisation it happens that the ERP are adjusted to the “bad” process. Experience shows that it is not the customised ERP that make organisations increase their profit, instead it is the reengineering of existing business processes that make it happens (Millman 2004).

4 Concluding Discussion

Melin (2003) says the opinions from executives are that ERPs should be centralised and standardised. It could be asked if and how role-based ERPs could be the centralised and standardized ERP solution that executives want to have (Askenäs and Westelius 2000), and at the same time be the flexible solution that the user would like to have. According to Kumar and van Hillegerberg (2000) the fact that ERP is a packaged solution creates a disconnection between the specified business requirements and the ERP solution. The solving of this is quite often that an “expert” on the ERP adjusts the ERP so that it better fit business requirements. But, the problem, as described by Kumar and van Hillegerberg, is that there is no guarantee that the expert has enough knowledge or sufficient knowledge to understand the requirements from a business perspective. The result could be that the implemented solution is more influenced by the inherent capabilities in the ERP than from the actual underlying business requirements.

According to Kumar and van Hillegerberg the problem with the mismatch between business requirements and ERP functionality is focussed on by the ERP vendors and they aim at solving this by customizing the ERP. However, this creates new problems in the migration between different versions, and either it could be that the new version not is backward compatible or it could be that the customer organisation have made modifications of the ERP, and these modifications do not automatically adjust to the new version of the ERP.

Askenäs and Westelius (2000) state that it is not possible to adjust or configure an ERP so that it fits all users need. It could be said that there are at least two different reasons for why this is the case. The first is that users to a high extent are different and even if they perform the same work tasks they do it differently from each other. The other reason is that an ERP builds on an architecture that make it hard to do the necessary changes in the configuration. The first reason is not possible to do so much about – it is hard to change people’s behaviour – but, the second reason could maybe by taken care of. It may be argued that ERPs has the intention of changing working behaviour which can be compared with Markus and Robey’s (1988) description of change through technical imperative, this attempt to change can probably be seen as one reason for ERPs implementing problems. The difficult question is to find out how the architecture should be constructed or changed to make it possible to have that high flexible solution that a totally role-based ERP demands.

Gammelgård, Lindström and Simonsson (2006) describe the following quality attributes of IT systems: Availability, reliability, data quality, functional fit, information security, interoperability, modifiability, performance, safety, usability and user productivity. All these could be described as capabilities received from IT systems or specifications of demands on IT systems. The question could then be how an ERP system and specifically a role-based ERP relate to these attributes. It can be stated that an ERP aims at fulfilling all these, however, the role-based ERP can be said aim at making a difference in: Availability, reliability, functional fit, modifiability, and usability and user productivity. But, to do so it probably has to change its basic architecture and maybe aim at what Kumar and van Hillegerberg describe as component-based ERP architecture. It can be argued that role-based ERPs aims at increasing the

level of adoption among users in organisations and that the result of having the role that Askenäs and Westelius describe as dismissed ERP should be avoided. The focus on role-based ERP and the aims at being customized is interesting to further follow and it could be asked if role-based ERPs is the solution to the negative reaction that users often have when ERP is introduced and it can be asked if and how role-based ERP change the adoption rate of ERPs. It could also be asked if the attempt of developing role-based ERPs is a solution to the problems that exist between ERPs and the organisations desire for customization of ERPs.

5 Conclusions and future studies

The question asked in this paper was: Why do software vendors focus on roles when developing the future enterprise resource planning (ERP) systems? The answer to that question can be described as: software vendors focus on roles since they see this as a way for closing the functionality gap existing within ERPs. But, it can be concluded that the main problem with doing this is to identify and describe relevant roles and the work tasks in which these roles need and use an ERP to fulfil the task. This indicate that future research are needed when it comes to how to find the roles as well as how to make sure that the roles are reflected in the ERP in the correct way. This emphasises on the specific problem that exists when developing a role-based ERP that suites small and medium sized enterprises (SMEs) since there are fewer people. But that kind of organisations probably more or less have the same amount of roles as in bigger organisations which means that one person has several roles or do the work tasks that belongs to several roles This means that the ERP needs to support different roles in the same user interface or have the possibility to be very easy to shift between different user interfaces developed for different roles.

Despite the fact that software vendors of ERPs started to talk about role-based ERPs in the late 1990's it has not really taken off. A question to ask is then why? It could be that it is not the way to go for the development. However, the indication from the discussion in this paper does definitely support role-based ERPs as the way to go. Another reason could be that the software vendors has not really solved the technical side of role-based ERPs or that they have not really found the business requirements needed to be able to develop a role-based ERP. It can also be concluded that, as described in figure 2, several factors, architecture, new business requirements and users perception of ERPs, influence the development at the same time as these factors influences each other and this definitely complicates the development of a role-based ERP.

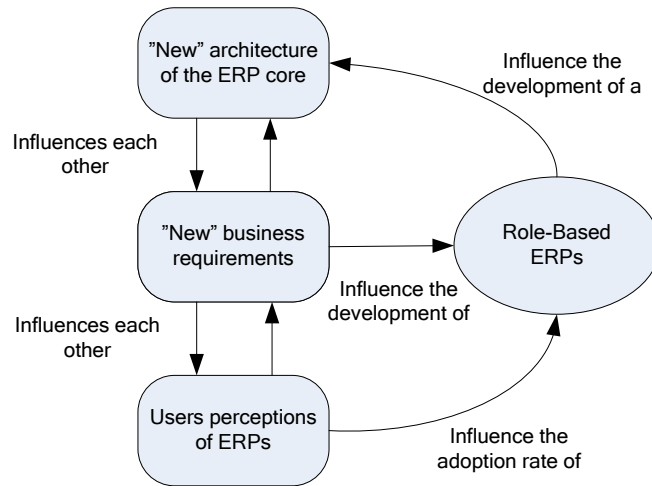


Fig. 2 Influencing factors on development of role-based ERPs

Another direction of future research on role-based ERPs could be to analyse if and how role-based ERPs aim at being more of a “consultant” from the user perspective while at the same time keeping a high level of control from the entire organisation perspective. This change of ERPs definitively impacts the entire enterprise architecture and it could be asked how the future architecture of ERPs should be constructed so that it supports this change.

The interesting issue that deserves future research is in what direction this role-based ERPs aims. Is ERPs heading for having a clearer role as a Bureaucrat, Manipulator, Administrator, and/or Consultant? In other words; do future ERPs aim at increasing control or increasing agility or increasing control and agility at the same time? Role-based ERP seems to go in the direction of being more of a Consultant. Simultaneously, the development of role-based ERPs impact the future enterprise architecture of organisations. But, the question remains if this is the way to go for ERP development if future ERP investments should decrease the gap between desired requirements and delivered functionality and thereby become a more successful investment for organisations.

One answer to why role-based ERPs not yet have seen the light could be said is dependent on difficulties with eliciting business requirements for a role-based ERP. However, the most apparent reason is probably a combination of technical problems and lack of understanding about business requirements. The role-based ERP probably needs a totally new architecture and this architecture is probably to a high extent dependent on what the business requirements are. This combination of problems is difficult to solve and emphasises that further research on both ERP architecture as well as ERP business requirements is needed before alignment between requirements and functionality becomes reality.

References

- Alvarez, R. (2002) Confessions of an information worker: A critical analysis of information requirements discourse. *Information and Organization* 12 (2), 85-107.
- 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.
- 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), 510-522.
- Bucher, T., Fischer, R., Kurpjuweit, S. and Winter, R. (2006) Enterprise architecture analysis and application - an exploratory study. In TEAR 2006 workshop.
- Carlsson, S. A. and Hedman, J. (2004) From enterprise resource planning systems to enterprise portals. In *The enterprise resource planning decade: Lessons learned and issues for the future* (Adam, F. and Sammon, D., Eds), pp 263-287, Idea Group Publ., Hershey, PA.
- Daneva, M. and Wieringa, R. J. (2006) A requirements engineering framework for cross-organizational ERP systems. *Requirements Engineering* 11 (3), 194-204.
- Esteves, J. and Pastor, J. (2001) Enterprise resource planning systems research: An annotated bibliography. *Communications of AIS* 7 (8), 1-51.
- Gammelgård, M., Lindström, Å. and Simonsson, M. (2006) A reference model for it management responsibilities. In TEAR 2006 workshop.
- Koch, C. (2001) ERP-systemer: Erfaringer, ressourcer, forandringer. Ingeniøren-bøger, København.
- Koch, C. (2005) The ABCs of ERP. CIO.
http://www.cio.com/article/14605/The_ABCs_of_ERP.
- Kumar, K. and Van Hillegersberg, J. (2000) ERP experiences and evolution. *Communications of the ACM* 43 (4), 22-26.
- Lee, J., Siau, K. and Hong, S. (2003) Enterprise integration with ERP and EAI. *Communications of the ACM* 46 (2), 54-60.
- 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), 307-330.
- Luo, W. and Strong, D. M. (2004) A framework for evaluating ERP implementation choices. *IEEE Transactions on Engineering Management* 51 (3), 322-333.
- Markus, M. L. and Robey, D. (1988) Information technology and organizational change: Causal structure in theory and research. *Management Science* 34 (5), 583-598.
- Melin, U. (2003) The ERP system as a part of an organization's administrative paradox. In 11th European Conference on Information Systems, Naples, Italy.
- Microsoft Dynamics (2006) Roles-based business productivity "Software designed for your people". pp 1-57.
- Microsoft Dynamics (2007) Microsoft dynamics role tailored business productivity "software designed for your people". pp 1-57.
- Millman, G. J. (2004) What did you get from ERP, and what can you get? *Financial Executives International* 5 (May), 15-24.
- Møller, C. (2005) ERP II: A conceptual framework for next-generation enterprise systems? *Journal of Enterprise Information Management* 18 (4), 483-497.
- Orlikowski, W. J. (1992) The duality of technology: Rethinking the concept of technology in organizations. *Organization Science* 3 (3), 398-427.
- Rolland, C. and Prakash, N. (2000) Bridging the gap between organisational needs and ERP functionality. *Requirements Engineering* 5 (3), 180-193.

- SAP (2007) User productivity enablement with SAP netweaver.
- Schindler, E. (2007) Getting clueful: Five things cios should know about software requirements. CIO.com.
- 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), 359-386.
- Sleeper, S. Z. (2004) AMR analysts discuss role-based ERP interfaces - the user-friendly enterprise. SAP Design Guild.
- 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.
- Staff, C. (2006) Sap's next generation ERP software enhances and improves business processes. *Caribbean Business*.
- 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.