

ERP for SMEs – is proprietary software an alternative?

Kai A. Olsen

Molde University College and University of Bergen, Norway

kai.olsen@himolde.no

Per Sætre

Molde University College

per.satre@himolde.no

Kai A. Olsen is a professor in informatics at Molde College and at the University of Bergen, Norway and an adjunct professor at the School of Information Sciences, University of Pittsburgh. Olsen has more than thirty years of academic experience with IT in industry, and has published many papers within this area and several books. He has experience over many years acting as a consultant for Norwegian and US organizations, developing systems based on the principles raised in this article.

Per Sætre is an associate professor. His experience is from Molde College, Norway and from CERN, Geneva. He has more than thirty years of academic and practical experience working with data base systems and logistics, and acting as a consultant for Norwegian industry.

Molde University College
Box 2110, N-6402 Molde, Norway
Phone: +47 712 14000

ERP for SMEs – is proprietary software an alternative?

(Conceptual paper)

Abstract

Vendors of ERP systems now look at Small and Medium sized Enterprises (SMEs) as an interesting market, selling the concept of a packaged system that can do everything. However, there are reasons to believe that the success rate will be even lower here than for the larger corporations. Niche companies rely on their idiosyncrasies and their ability to conform to customers demands in a flexible manner. A standard ERP system may impose a rigid structure on a company, and threatening the dynamic nature of many SMEs.

In this paper we propose an alternative based on proprietary software. In-house development leads to systems that are tuned to the company. Full control of the software makes it possible to remain flexible and dynamic, and to conform to the need of the customer at any time. However, in-house development is often not considered as an alternative. Managers remember earlier software projects that were both time consuming and expensive. But software development today is very different from twenty or thirty years ago. We show that proprietary software is a feasible solution, especially as new standards allow us to compose a system out off both off-the-shelf products and our own components. Small companies have an advantage here, as they easily can integrate “best-of-breed” products as components in a proprietary ERP system.

These ideas have been implemented in a niche company over the last years. The experiences from this implementation are presented here as a case.

Keywords: SME, ERP, proprietary software

Introduction

“IT Doesn’t Matter” writes Nicholas G. Carr in a much debated paper (Carr, 2003). He elaborates the arguments in his book “Does IT Matter? Information Technology and the Corrosion of Competitive Advantage” (Carr, 2004). He states that IT is becoming a commodity and its strategic importance has been diminished, i.e., when IT is available for everybody, it cannot offer a strategic advantage. In this situation, says Carr, the risks, operational or financial, become more important than the advantages. He especially warns against overspending, on hardware or software, and sites Oracle’s Larry Ellison “most companies spend too much [on IT] and get very little in return.”

Carr makes an exception. “A few companies may still be able to wrest advantages from highly specialized applications that don’t offer strong economic incentives for replication, but those firms will be the exceptions to the rule.” We argue that this window of opportunity is somewhat bigger than what Carr seems to believe. This is especially the case for many niche companies. A niche company utilizes a specialized market. It may

exist due to its technology, quality, good customer relations or, for example, the ability to customize products (Levy et al, 2001; Levy and Powell, 1997). The SME survival strategy is to be innovative, flexible and efficient, to outsource simple tasks and concentrate on the complex activities (Ghosh et al., 2001). Their strategic advantage may be that they are idiosyncratic.

Most SMEs operate in a highly dynamic world, where both internal and external requirements may change (Branzei and Vertinsky, 2004). Changes may come from the need to be more cost-effective, from customers in the form of requirements for new products and product variants, from government agencies in the form of regulations, or by advances in technology. Often the SME is the weaker part in a supply chain and thus the ability to adapt to changes imposed by customers or suppliers will be an important competitive factor. Most SMEs have utilized the flexibility that comes from having a lower number of orders, customers, employees, etc. when changing processes and practices. It is therefore important that this flexibility is retained when new IT systems are implemented.

The use of commodity software, such as ERP systems, may force a more rigid structure on a SME and thus weaken their competitive advantage. The solution that we advocate is to develop proprietary software that can conform to their special needs, and thus strengthen their competitiveness. In this way the SMEs will be able to utilize the computer as it was meant to be, as a *programmable* machine. Today, when most software comes in boxes, we tend to forget this opportunity.

The need to be in control

It is important that a company aligns its information systems with its business strategy (McGovern & Hicks, 2004; Levy et al, 2001; Soh et al, 2003; Yen & Sheu, 2004). This becomes more important as we move from automating simple processes, such as accounting or payroll to “knowledge” systems that incorporate best practices and core processes (Jacobs and Whybark, 2000; Soh et al, 2000). That is, as IT takes care of higher level functions it is important that the system implements the business philosophy of the company. In a way, for many companies, the core IT system *is* the company. This is very apparent for Internet banks and other “virtual” companies, but is also true for companies with physical products. Ideally the day to day activities should be handled by the IT system, while management implements their ideas and policies through the system. Management define long term strategies, take care of customer relations on a high level, and the IT system performs production planning, procurement, formalized customer relations, etc., automating some processes and informing others.

As Carr points out, standard software embodies a standard way of doing business. This is especially true for packaged ERP systems that come with their own reference model, often also with consultants who implement this model (“company in a box”). This may not suit the customer, especially if this is a niche SME. These encompassing systems will reduce the idiosyncrasies of the company from the transaction processing part all the way up to company management and customer relations. While this may be an advantage for some companies, it may be disastrous for companies that thrive on their ability to conform to customer demands, to be flexible and dynamic (Akkermans et al., 2003). In a

way, when choosing an IT system the SME has to decide if it want to run the business as everybody else, or to continue developing their own idiosyncrasies.

Over 70% of the Fortune 1000 companies have implemented ERP systems (Yen et al, 2002). Many report successful implementation (Markus and Tanis, 2000) even if success is a difficult criterion to define when it comes to ERP systems (Markus et al, 2000). The problems of getting ERP systems to work are well documented (see for example Davenport, 1998; Parr and Shanks, 2000; Sarker and Lee, 2003; Scott and Vessey, 2000) and there are many failures even for large firms that have the resources needed to perform a careful planning and implementation (Bingi et al, 1999; Griffith et al 1999; Hayes et al, 2001; Mandal and Gunasekaran, 2003; Vogt, 2002).

Small and medium sized firms may not have the resources to do a careful study of available systems, neither to plan nor to perform a low-risk implementation. Further, many small companies, especially niche companies, have more to risk by standardizing core functions, and by conforming to a system that reduce their ability to comply with changing internal and external factors. When the IT system embodies important business processes, i.e., when the IT system has strategic significance, the system has to reflect these changes. The company must therefore be able to modify the software and to add new functionality whenever needed. This is not easy to achieve with a general ERP system. Attending user conferences to promote general modifications to the kernel part of an ERP system is at least time-consuming and therefore unacceptable. Add-on modules are a possible solution, but these often only work until the next version of the standard system is installed.

With proprietary development, however, the IT systems can be used to strengthen the idiosyncrasies of a company, to tune the company to its markets, customers and products. At the same time the company retains full control over its core processes. Updates can be performed whenever needed, independent of software vendors. What we can hope to achieve is that the company, with the new IT system, is as, or more, flexible than before.

In-house development

A major effort of developing one's own core functions is the specification part, defining functionality, important processes, and terminology. This is a complex and time consuming part of system development, but has so many positive side effects that it should not be looked upon as an IT-cost. The specification phase gives a metaview of the company, very different from the close horizon one has in most SMEs during day to day activities, on getting new orders, running processes, handling exceptions, installing new equipment, etc. During system specification the focus is on generic parts, not so much on details. To be able to perform system design, the overall strategy and goals of the company must be clarified. For most firms this is an important and necessary process that will be triggered by a decision to develop a new system. Inefficiencies in current processes may become apparent when these are mapped into process diagrams and discussed by the organization. An information needs analysis may show deficits in existing data base structures, and expose the importance of agreeing on a basic terminology for the SME: What is an order, a due date, a delay?

An in-house development offers full freedom. All processes are candidates for reengineering, and all options are available for implementation. In opposition, an ERP implementation is often viewed as a mapping from the existing systems to the new ERP system. The idea is to find a common denominator between the company and the business models embedded in the ERP system. Success is determined when the system work, seldom of how well it services the company (Markus et al, 2000).

While an implementation of a standard ERP system usually will have to include all IT functions in a company, also replacing systems that are functioning well in the quest of getting one system for everything, an in-house development can concentrate on those parts where new or improved functionality is needed. That is, the focus can be on important core functions, following an “if it ain’t broke don’t fix it” philosophy for all other parts.

An in-house development also has social benefits. It will be easier for employees to identify themselves with a system that is developed by the company. The probability that the system will match the preferences of the users is also higher than if the development starts by parameterizing a standard ERP system. While customers of standard systems are recommended to adjust the business processes to that of the ERP system, the in-house system will be modeled based on the company, not on a common understanding of all companies. Training will also be simplified when the system is developed with just in-house users in mind. Then terminology, menus, forms, reports, etc. and functionality will directly reflect the requirements of the users.

There is also another factor that should be considered. The introduction of a new IT system will always be demanding for users, in-house development or not. However, with in-house development most details will be as before, at least where existing formats and processes are considered adequate. That is, users can then concentrate on understanding the important new parts of the system. While the implementation of standard ERP systems requires singular systems to be abandoned, we may follow a best-of-breed approach – keeping all useful sub systems.

Costs

While proprietary development was the only alternative for many companies in the seventies and eighties, it is frowned upon today. Many managers have recollections of projects that ran over budget, that exceeded all time limits and that often failed. However, software development today is vastly more efficient than twenty years ago. New tools, new development methods, the availability of open source components and especially new standards have made proprietary software development an interesting possibility.

In-house development should be limited to the core functions. It is not necessary to develop a full body ERP system. Instead a custom designed system should be put together out of several different systems, where only the core part is developed in-house. (Ferney and Bell, 2005; Light et al, 2001). The standards that have been developed over the last thirty years make this possible. The most important may be:

- Relational databases and SQL, a database language standard that is part of most commercial database systems. That is, a query expressed in SQL can be sent to and executed by most database systems.

- ODBC, JDBC and other database drivers for database connection that allows client programs to view different database system as if they were identical.
- VBA, a programming language that is used for customization in many applications, among these MS Word, Excel, Microsoft Access and AutoCAD.
- The Web “standards” for formatting (HTML) and communication (HTTP), along with the underlying Internet standards for naming (IP) and communication protocol (TCP).
- Office “standards” such as the PDF-, DOC- and XLS-formats for documents and spreadsheet files.
- XML, the standard for defining markup languages that can be used as a basis for formalized data exchange between computers.

Some of these standards are defined by international organizations, other have become de facto standards due to their widespread use. These standards allow us to mix off-the-shelf programs, pre-made components and our own programs in a very free manner. Thus we no longer have to develop the complete system ourselves. It can be put together out of parts. That is, with in-house development only the core functions that identify the business are implemented. For all other functions off-the shelf products can be used. Here we can pick the best from each category, the best accounting system, the best planning system or the best drawing package. We do not have to take everything from one vendor and will also find that the standalone products available in the marketplace often have more modern user-interfaces and greater functionality than similar systems embodied in a traditional ERP system (Light et al, 2001).

It will be the job of the proprietary software to integrate these functions in a seamless manner, relying on the standards mentioned above. With these, it is also quite easy to replace one system by another. A new accounting system, or a new computer aided design (CAD) package, can easily be integrated with other systems as long as the new system supports the underlying standards.

While each system may have its own database, we still enforce the rule of storing data only once, in only one database. That is, the integration will not be as complete as in an ERP system, but the most important principle from the database administrator’s point of view is maintained. With the database standards it is possible for the core part to retrieve data from different databases, and to store data in different databases. In practice, the users will see only one “virtual” database. While an ERP system offers a common user interface for all program modules, we will here get a conglomerate of different interfaces. However, many programs have a similar “look & feel”, so the similarities will be greater than the differences. Also, as ERP users only relate to a subset of functions on a daily basis, most users will only access the core functions and a few other programs.

In developing the core functions we can use high level application generators or standard programming environments. Either way development today is much simpler than just a few years ago. Even within the core part of the system one will be able to use premade components for many sub parts, for example taking advantage of open source software (Bitzer, 2005; Bonaccorsi & Rossi, 2003; West, 2003). We can therefore view parts of the core program as “glue” to join premade programs or components.

An application generator increases the efficiency of an experienced programmer tremendously compared to conventional programming. In many cases the development costs will not exceed what companies pay in licenses and parameterization for standard ERP systems. We should emphasize that these methods of developing an in-house “ERP system” will be best suited for small and medium sized companies. For large enterprises the need for strict security, very efficient processes, and implementation on different software platforms will make it more difficult to “glue” together applications out of separate programs.

Case

Our case company is a producer of propeller blades. Since the propulsion system of modern ships is designed according to hull characteristics, each series of blades will have their own geometry. Based on these data, which may come in the form of a complete CAD model or by coordinates for a set of sections, the company produces a wood model of the propeller blade. This model is used to create a mold in a sand-fixtured solution, and the blade is then cast in nickel aluminum (NiAl) bronze. A final process includes measuring the pitch and thickness of the blade at predetermined points, and grinding the blade until the tolerances are within ISO requirements.

The company has a small niche, specialized product, and survives in the world market by having excellent contacts to major customers and by mastering the difficult processes of NiAl casting and propeller manufacturing. The idiosyncrasies of this company are defined by:

- Being a foundry. In this respect they compete with all other foundries, some of which are located in countries with very low labor costs.
- Specializing in NiAl casting. On a scale of difficulty NiAl casting is at the high end. Therefore there are relatively few foundries that can handle this complexity, but even some of these are located in low cost areas.
- Being a propeller manufacturer. There are still fewer foundries that produce propeller blades, that can design models, and perform the end control of finished blades.

Its customers are major companies that design and manufacture complete propulsion systems.

Core functions	Industry functions	Business functions
Entry and storing of order- and model data	CAD program	Office applications
Production planning	CNC software tools	Accounting and financial systems
ISO functionality	Casting software	Sales support systems
Quality control, documentation	Energy management tools	Document handling software

Figure 1. IT-applications in the case company

The IT-applications for our case company are presented in Figure 1. The three columns show that this company has core functions that define its identity, that it belongs to a type of industry (foundry, manufacturing) and that it is also a common business. That is, they share the business applications with most other firms and the industry functions with other manufacturing firms and foundries. They may share their core functions with their competitors, or these may be unique for this company. Either way, these functions are the target for in-house development.

Order entry is particular to our company as the system has to accommodate for all the different ways that their customers describe the propeller blade model, the geometry of the blade. This includes handling different formats (from ASCII-files, through XML descriptions to complete CAD models), different coordinate systems (there are at least 15 systems in use) and also different methods to represent the basic geometry. The company also stores “best practice” data for each type of propeller, for example the ideal temperature for the casting process.

The planning application will find an optimum or near optimum production plan with regard to tardiness based on the list of orders, their due date and other characteristics, and the available resources in the foundry. Capacity is determined by the available work space, the workload for each type of blade and the blade size, expressed in the form of a combination table. This table offers different combinations of blades that can be produced on a single day. While this problem has certain similarities with a cutting stock problem, it is unique in the sense that standard optimization algorithms will not work. Thus, the planning function in this company really reflects their idiosyncrasy, and cannot be fulfilled by any standard system. Or, said in a different way, most of the functions that come along with a standard ERP system, for procurement, order handling, inventory etc, are of no use here.

Since casting is a non-exact process the propeller blades will be casted somewhat larger than the nominal requirements, and then grinded into the correct shape defined by a set of ISO standards. Based on data from a computer controlled measuring machine (CMM), a “best fit” process is performed to find the minimum of grinding necessary to make the blade comply with the ISO standards. This is a “brute force” algorithm that finds the optimum solution. These processes are of course proprietary to propeller manufacturing and will never be a part of any standard package. At the same time they are crucial to our case company, guaranteeing quality and efficiency even for the most complex blades.

The firm has a set of quality control and documentation routines. Some of these are internal, but many are external, required by customers or classification agencies. Most of these are idiosyncratic to the business (foundry, propeller manufacturing).

Based on this review it is clear that the company had to develop its own core functions, while both industry and standard business applications could be met with off-the shelf products. The core administrative system (order entry, etc) is developed in Microsoft Access (an application generator) while the planning system is developed in Visual Basic (to give faster performance). The resulting system can capture any order and geometric description that is received electronically, independent of format, and the same time as it offers manual entry for orders that are received on paper. When needed, usually once a week, the planning system calculates an optimum plan for all orders. Based on this plan

the administrative system will set up detailed jobs for the foundry, register important process data and create the necessary documentation.

The system will also collect the data that is needed for sending an invoice. Whenever an order has been shipped the core system will automatically create a new record in the accounting systems database, using the SQL and ODBC standards. All further handling of the invoice is now performed by the standard accounting system. That is, from the view of the accounting system our case company is a standard business that sells stocked items.

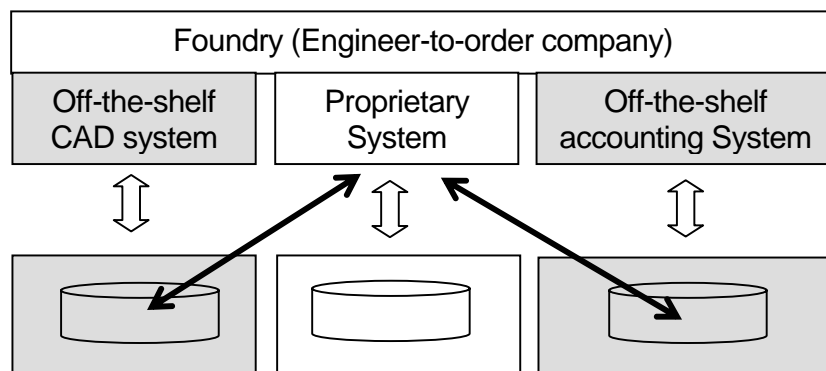


Figure 2. One company – three integrated systems

Similarly there is a connection between the core part and the CAD programs. The layout is illustrated in Figure 2. Of course, we end up with a system of several databases: within the core system, the accounting system and also within the CAD application, but a data item is stored only once. For example, at order entry the core system needs customer data. To avoid duplication these data are retrieved from the accounting system database, which is no problem using the standards mentioned above.

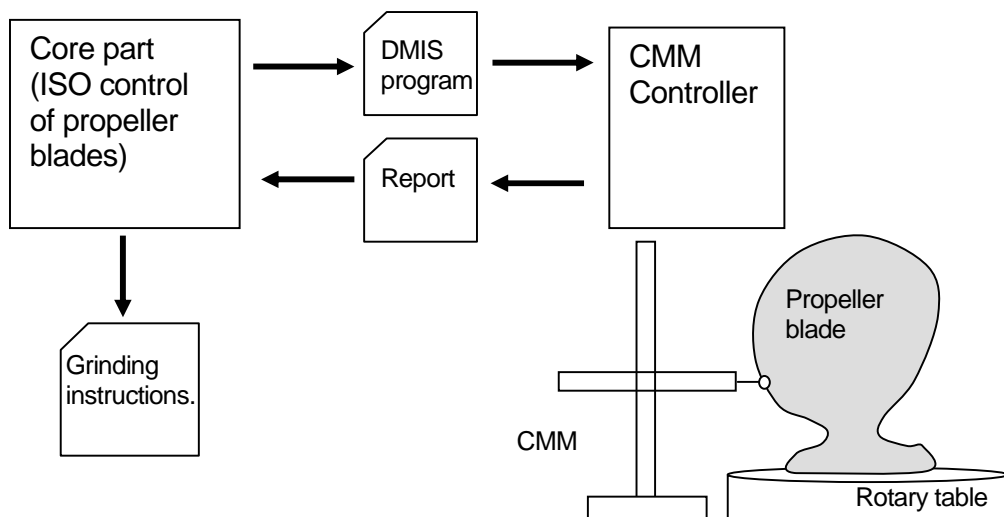


Figure 3. CMM setup for measuring propeller blades

The system for automatic measuring and ISO control of finished blades is presented in Figure 3 as an example of a core process. The system uses a CMM with a rotary table (to

measure both sides of a propeller). The CMM controller can execute programs written in a standard programming language for measuring machines. Programs for this machine are generated automatically based on geometric data from the core database. The process of generating a measuring program, performing measurements, importing the result reports, and finding the minimum grinding that is needed is close to being completely automatic, integrating core parts seamlessly with standard machinery.

The total development effort is kept to a minimum by integrating standard programs and standard tools for all common business and industry functions, nevertheless it was a major effort to design and implement all core functions. External consultants, both with propeller design and IT knowledge, were involved, along with university researchers and students (Ph.d and M.Sc). Still, development costs are comparable, probably lower than implementing costs for an ERP system. Further, since most systems have been developed in MS Access, which is included in the Office package, no additional licensing costs are incurred in this case.

More important is the fact that the company now has a set of programs that are tuned to their business. The value of being able to retrieve orders electronically independent of the format used is very high, as this reduces errors at the same time as efficiency is increased compared to manual entry. The planning system alone gave the company an increase in efficiency of 25 % compared to a previous manual system (Nonås and Olsen, 2005). Now the customers can get an exact due date for all orders from the company's Web-pages system. The planning routine also allows for "what if" functionality, for example to see the effect of accepting an emergency order, or to have an additional shift on Saturday. Customers can also run the planning system through the company's eBusiness system, getting a due date for new orders dependent on order characteristics.

The core system is a part of the company's competitive edge. Even if the "economic incentives" are in place, it may take some time before the competition has similar systems up and running. Since this requires IT competence, competence on propeller design, as well as manufacturing experience there are reasons to believe that this will take some time. Thus, it is probable that our company will keep its competitive edge at least some years into the future, years that can be used to improve efficiency even further with the help of sophisticated IT systems.

Niche companies have to adjust to market demands; this is true also for our case company. In one example, an important customer, a major manufacturer of thrusters, wanted to split every order in two. Instead, as one normally does, of ordering both the product, here a propeller blade, and the classification of this product (performed by a licensed classification agency specified by the ship owner) they wanted to separate these tasks. Since these classification agencies are reluctant to accept each others quality control, it would be necessary for the manufacturer to store a set of reserve blades for each blade type and each agency. In practice this would have been impossible. The alternative that was chosen is to produce spare blades independent of classification, delaying this task until the blades are needed. This reduces effective lead time from about three months (producing a blade) to three days (classification of a blade). However, splitting orders in this way imply that the system must correlate a classification order to a previous production order for a generic blade. While no big effort in a proprietary

system, modifications of this kind would be very difficult to implement in a standard ERP system.

It may be argued that a proprietary solution may make it more difficult to follow along with IT in other areas. However, this is not the experience from our case company. When they, for example, wanted to enhance their eBusiness solution from a “Web presence” to an “integration” phase (see Rao et al, 2003) this could easily be achieved using current Web design tools to retrieve data from the proprietary data base. Here, as in other areas, we see that control of the kernel software facilitates the implementation of new solutions.

Conclusion

All companies need to be in control of core IT functions, that is, the functions that embody the basic business ideas. For many small and medium sized companies we argue that this requirement can best be met by an in-house development of these systems. This allows the company to implement just the system they need, and to change the system along with the company. In-house development offers a competitive edge. While the implementation of a standard ERP system is an implicit acknowledgement of the fact that one does not want to compete on IT, development of proprietary software implies that one want to increase competitiveness by the use of IT.

IT development is expensive, but with the use of modern tools it is less expensive today than just a few years ago. More important is the fact that a company only has to develop its own core functions, other functionality, for most industry and most business applications, can be implemented by the use of off-the-shelf products. A set of underlying standards, for database connection, query languages, etc. makes it possible to connect these systems in a seamless manner. That is, even if the final system is put together out of many components, the users can still view the system as a whole.

References

- Akkermans, H.A., Bogerd, P., Yücesan, E., van Wassenhove, L.N. (2003) The impact on ERP on supply chain management: Exploratory findings from an European Delphi study, *European Journal of Operational Research*, 146, 284-301.
- Bingi, P., Sharma, M.K., Godla, J.K. (1999) Critical issues affecting an ERP implementation, *Information Systems Management Journal*, Vol. 16, No. 3, 7-15
- Bitzer, J. (2005) Commercial versus open source software: the role of product heterogeneity in competition, *Economic Systems*, in press.
- Bonaccorsi, A., Rossi, C. (2003) Why open software can succeed, *Research Policy*, 32, 1243-1258.
- Branzei, O. & Vertinsky, I. (2005) Strategic pathways to product innovation capabilities in SMEs, *Journal of Business Venturing*, in press.
- Carr, N.G. (2003) “IT Doesn’t Matter”, *Harvard Business Review*, May, pp 41-49.
- Carr, N.G. (2004) *Does IT matter? Information technology and the corrosion of competitive advantage*, Harvard Business School Press, Boston.

- Davenport, T.H. (1998) "Putting the Enterprise into the Enterprise System", *Harvard Business Review*, Vol. 76, No. 4, pp 121-131.
- Ferneley, E. & Bell, F. (2005) Using bricolage to integrate business and information technology innovation in SMEs, *Technovation*, in press.
- Ghosh, B.C., Liang, T.W., Meng, T.T., Chan, B. (2001) The key success factors, distinctive capabilities, and strategic thrust of top SMEs in Singapore, *Journal of Business Research* 51, 209-221.
- Griffith, T.L., Zammuto, R.F., Aiman-Smith, L. (1999) Why new technologies fail? *Industrial Management*, Vol. 41, No. 3, 29-34.
- Hayes, D.C., Hunton, J., Reck, J. (2001) Market reaction to ERP implementation announcements, *Journal of Information Systems* Vol. 19, No. 1, 71-98.
- Jacobs, F.R., Whybark, D.C. (2000) *Why ERP? A primer on SAP implementation*, Irwin/McGraw-Hill, New York.
- Levy, M. & Powell, P. (1997) Assessing the value of Information Systems planning at Car Spring Co. *International Journal of Technology Management*, Vol. 13, No. 4, 426-442.
- Levy, M., Powell, P., Yetton, P. (2001) SMEs: aligning IS and the strategic context, *Journal of Information Technology* 16, 133-144.
- Light, B., Holland, C.P., Wills, K. (2001) ERP and best of breed: a comparative analysis, *Business Process Management Journal*, Vol. 7, No. 3, 216-224.
- Mandal, P. & Gunasekaran, A. (2003) Issues in implementing ERP: A case study. *European Journal of Operational Research* 146:274-283
- Markus, M. Lynne, and Cornelis Tanis (2000) The Enterprise Systems Experience—From Adoption to Success. In R.W. Zmud (Ed.) *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Cincinnati, OH: Pinnaflex Educational Resources, Inc.: 173-207.
- Markus, M. Lynne, Sheryl Axline, David Petrie, and Cornelis Tanis (2000). Learning from Adopters' Experiences with ERP—Successes and Problems, *Journal of Information Technology*, 15, 4 (December) 245-265
- McGovern, T. & Hicks, C. (2004) How political processes shaped the IT adopted by a small make-to-order company: a case study in the insulated wire and cable industry, *Information & Management* 42, 243-257.
- Nonås, S.L., Olsen, K.A. (2005), "Optimal and heuristic solutions for a scheduling problem arising in a foundry", *Computers & Operations Research* Vol. 32, No. 9, 2351-2382.
- Parr A. & Shanks G. (2000) A model for ERP project implementation. *Journal of Information Technology* 15:289-303
- Rao, S. S., Metts, G., Monge, C.A.M. (2003) Electronic commerce development in small and medium sized enterprises. A stage model and its implications, *Business Process Management Journal*, Vol. 9, No. 1 (11-32).

- Sarker, S & Lee, A.S. 2003. Using a case study to test the role of the tree key social enablers in ERP implementation, *Information & Management*, Vol. 40, No. 8, 813-829.
- Scott, J.E. & Vessey, I (2000) Implementing Enterprise Resource Planning s: the role of learning by failure. *Information system frontiers*, 2(2):213-232
- Soh, C., Kien, S.S., Tay-Yap, J. (2000) Cultural fits and misfits: is ERP an universal solution?, *Communications of the ACM*, Vol. 43, No. 4, 47-51.
- Soh, C., Sia, S.K., Boh, W.F., Tang, M. (2003) Misalignments in ERP implementation: a dialectic perspective, *International Journal of Human-Computer Interaction* Vol. 16, No. 1, 81-100.
- Vogt, C. (2002) Intractable ERP a comprehensive analysis of failed enterprise-resource-planning projects, *ACM SIGSOFT* 27 (2), 62-68.
- West, J. (2003) How open is open enough? Melding proprietary and open source platform strategies, *Research Policy*, 32, 1259-1286.
- Yen, D.C., Chou, D.C. & Chang, G.D. (2002) A synergic analysis for web-based enterprise resource planning systems. *Computers standards & Interfaces*, 24:337-346
- Yen, H.R. & Sheu, C. (2004) Aligning ERP implementation with competitive priorities of manufacturing firms: An exploratory study, *International Journal Production Economics*, 92, 207-220.