Introduction: ERP – The Quiet Revolution?

Peter Seddon, Graeme Shanks and Leslie Willcocks

Organizations invest in enterprise system software from vendors such as SAP, Oracle, PeopleSoft, Siebel, and i2 Corporation to gain access to powerful computer-based information systems more cheaply than through custom-built software development. What they acquire is a highly flexible software product, containing solutions to the needs of many of the vendor's existing customers, that impounds deep and detailed knowledge of many good and less good ways of conducting a wide range processes in a broad range of industries. The difficulty for the licensing organization is to identify the right combination of processes for its own changing needs and to implement those processes in its own organization.

Enterprise systems and the internet were probably the two most important information technologies to emerge into widespread use in the 1990s. According to Technology Evaluation.com, the total revenue from the enterprise system software and services market was US$18.3 billion in 1999 and US$19.9 billion in 2000 (Gilbert, 2000; Jakovljevic, 2001). Enterprise system implementation costs are often reported to be five to ten times the cost of software licenses (Davenport, 2000; Scheer and Habermann, 2000). If so, organizations worldwide spent something like US$100 billion per annum on enterprise systems in both 1999 and 2000. In short, organizations around the world have made huge investments in enterprise systems in the past decade.

Enterprise systems (ES) are large-scale organizational systems,[1] built around packaged enterprise system software. Enterprise system software (ESS):

1 is a set of packaged application software modules, with an integrated architecture, that can be used by organizations as their primary engine for integrating data, processes, and information technology, in real time, across internal and external value chains;
2 impound deep knowledge of business practices that vendors have accumulated from implementations in a wide range of client organizations, that can

[1] Systems composed of people, processes, and information technology.

exert considerable influence on the design of processes within new client organizations;

3 is a generic 'semi-finished' product with tables and parameters that client organizations and their implementation partners must configure, customize, and integrate with other computer-based information systems to meet their business needs.

Enterprise system software includes enterprise resource planning (ERP), customer relationship management (CRM), supply chain management (SCM), product life cycle management (PLM), enterprise application integration (EAI), data warehousing and decision support, intelligent presentation layer, and eProcurement/eMarketplace/electronic exchange software. In June 2000, AMR Research forecast that by 2004, sales of CRM, SCM, and eMarketplace software would *each* match the current US$20 billion per annum sales of ERP software. By 2002, sales forecasts were not as rosy, but as vendors rushed to 'internet enable' their products and allow 'best of breed' solutions to cooperate, the enterprise software market was still undergoing a period of tumultuous change.

The focus of this book is on the most important class of enterprise system, namely ERP systems. Today's ERP systems have evolved from packaged software for supporting material requirements planning (MRP) and manufacturing resource planning (MRP II), hence the strange name: enterprise resource planning. To those unfamiliar with the capabilities of MRP software, the combination of words 'enterprise', 'resource', and 'planning' conveys very little information about the capabilities and purpose of ERP software, but the label has stuck; the acronym 'ERP' has effectively become a new three-syllable word in the English language. There is no universally accepted definition of 'ERP' software (Klaus, Rosemann, and Gable, 2000), but Deloitte Consulting's ERP's Second Wave report provides a useful starting point. According to Deloitte:

> An Enterprise Resource Planning system is a packaged business software system that allows a company to:
> - Automate and integrate the majority of its business processes
> - Share common data and practices across the entire enterprise
> - Produce and access information in a real-time environment
>
> (Deloitte Consulting, New York, NY, 1999)

This definition, which treats an 'ERP system' as a piece of software (not an organizational system), is similar in meaning to the first point of our definition of ESS above. What it does not say is that ERP is essentially 'back

office' software: organizations use ERP software to integrate enterprise-wide information and processes for their financial, human resources, manufacturing and logistics, and sales and marketing functions. In addition, the Deloitte definition does not address points 2 and 3 in our definition above.

The consequence of the second point in our definition of enterprise system is eloquently expressed in Davenport's (1998) much-quoted line:

> An enterprise system imposes its own logic on a company's strategy, culture and organization. (Davenport, 1998: 122)

Essentially Davenport is saying that unlike custom-built software, which is by definition tailored to the precise needs and nuances of the organization, packaged enterprise system software is unlikely to be a perfect fit with every client's needs and this can cause problems. Although highly configurable to different situations, ERP systems' data structures, program code, and in-built assumptions about processes can impose patterns of behaviour on organizations that some find very difficult to accept. If the client organization is not careful, it may find the enterprise system in the driver's seat, defining how the business will be run.

The problem for the client organization is to understand the various business process options supported by the software and to make the right choices about which process variants to implement. The knowledge impounded in the various process variants has been accumulated by vendors from implementations of their software in a wide range of their clients' organizations. Furthermore, the capabilities of the software grow day by day as vendors respond to existing clients' requests for new features. Understanding what the software is capable of doing, then either customizing the software or changing processes in the organization to fit the software, is probably the central challenge in enterprise system implementation.

This brings us to the third point in the definition above. Enterprise system software is a generic 'semi-finished' product that user organizations must tailor to their needs. Tailoring the software using parameter settings provided by the vendor is usually called *configuration* (Bancroft, Seip, and Spregel, 1998: Chapter 5; Davenport, 2000: 150–153). For example, financial software must be configured so that it 'knows' which companies exist, which companies are subsidiaries of which other companies, the base currency of account for each subsidiary, the sales tax regimes for each subsidiary, and so on. Adding non-standard features to the software is usually called *customization*. Customization involves writing program code. It can range from relatively simple changes, such as developing very special new reports, through to major changes, such as

changing the actual program code in the vendor's software (Brehm, Heinzle, and Markus, 2001). An implementation that involves configuring but not customizing the software is called a 'Vanilla' implementation.

Vendors usually recommend against customization, both because of the software-development risk and because new releases of the software may need re-customization. One way that organizations can avoid customization is by changing their processes to match those supported by the software. The risk in this option is that changing organizational processes is hard. Furthermore, the processes embedded in the system software may not be appropriate for the needs of the organization.

Apart from all the normal problems of information system project management, the novel difficulty for teams implementing ESS is to decide which mix of configuration, customization, and process change is best for the organization. This difficulty is compounded by uncertainty about which processes will be best for the organization (both now and in the future) and, frequently, by lack of understanding of the capabilities of the ES software. After user managers have learnt more about what the software can do, for example, after a year or so, they are often in a much better position to specify what they want to achieve with their software.

ERP vendors experienced rapid growth throughout the 1990s and by 2002 most Fortune 500 companies had them installed. Revenues flattened out in 1999, both because year 2000-related implementations were being completed, and because the market was saturating: many large organizations had already converted to ERP systems. By 2002 five major vendors dominated the ERP market. Market shares in 2000 were SAP 32%, Oracle 14.5%, PeopleSoft 9%, JD Edwards 5%, Baan 3%, and 'Other' 41% (Jakovljevic, 2001). By 2002, there was also a sizeable and growing market in upgrading ERP software to accommodate new software releases, particularly as ERP vendors internet-enable their product offerings.

Because they affect so many parts of an organization, ERP systems can provide a huge range of benefits and problems, often with different benefits in different organizations. Many organizations around the world, having collectively spent hundreds of billions of dollars implementing enterprise systems, are now asking, 'How can we gain greater benefits from our investment in enterprise systems?'. In fact, according to Markus and Tanis (2000), 'the key questions about enterprise systems*from the perspective of an adopting organization's executive leadership* are questions about success. For example: Will our investment pay off? Did our investment pay off?' These questions are echoed by Davenport (2000), Deloitte Consulting (1998), Markus and Tanis

(2000), and Ross and Vitale (2000) who note that business benefits from ES use are multidimensional, ranging from operational improvements through decision-making enhancement to support for strategic goals.

One of the best attempts, to date, to answer these questions about benefits from enterprise systems and how to achieve them is the benchmarking partners study published by Deloitte Consulting (1999). That report, based on 'in-depth interviews with 164 individuals at 62 Fortune 500 companies' concludes:

> Until now, conventional wisdom saw going live as the end. In sharp contrast to this view, our study uncovers at least two distinct waves of ERP-enabled enterprise transformation. The First Wave refers to the changes to an organization that include and accompany going live with ERP. The Second Wave, on the other hand, refers to the actions that are taken after going live that help organizations achieve the full capabilities and benefits of ERP-enabled processes. (Deloitte Consulting, 1999)

It is our belief, as we assemble this book, that many organizations have now begun to focus on the second wave, in terms of maximizing benefits, making continuous improvements, and taking advantage of new, including web-based, technologies and new ways of configuring systems in a journey to establish the integrated, extended business enterprise. Hence the title of this book: *Second-Wave Enterprise Resource Planning Systems: Implementing for Effectiveness*. In this book, we have assembled some of the world's best research on ERP systems, with a view to providing a foundation for second wave improvements to enterprise systems.

One recent study of ERP systems concludes that the way to maximize benefits from enterprise systems over the long term is to set up process improvement teams, led by highly motivated process owners, whose job is to understand both the evolving capabilities of the software and needs of the organization, and to strive to maintain a reasonable on-going level of fit between the software and changing organizational needs (Shang, 2001). Achieving such on-going fit is hard. Enterprise system software is so complex that, no matter how good the initial implementation, it is unlikely that it will be a good fit for the organization. Over time, a growing understanding throughout the organization of the potential benefits of enterprise system use, and changes in leadership, strategy, competitors' behaviour, and organizational structure, will lead to changes in perceived needs. In addition, the enterprise system software will change as vendors issue new releases, extend the scope of their software, implement support for new technologies (for example, the internet), and as the vendor firms themselves merge or fail. These pressures result in an on-going need for

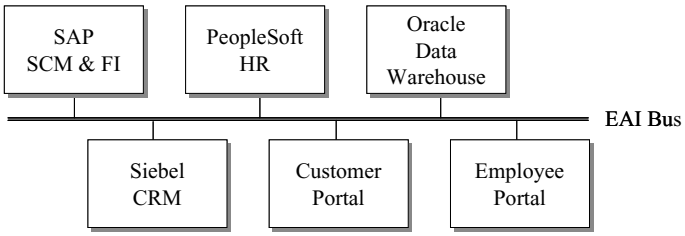

Figure 1 Integration of application software through enterprise application integration (EAI)

enterprise system improvement initiatives. In the organizations Shang studied, the best people to lead such initiatives were the process owners. These managers had both the necessary deep understanding of organizational needs and the authority to implement necessary changes in the organization. However, they also had to be prepared to devote considerable energy to understanding the capabilities of their enterprise system software.

The future for ERP software and its brothers and sisters in the enterprise system software family is, at time of writing, unclear, though the final chapter of this book suggests two plausible scenarios. Three issues commanding attention in 2002 were:

- application software integration (for example, through EAI and intelligent portals);
- inter-organizational process integration through exchanges;
- use of shared services and application service provision.

With respect to application software integration, many organizations are presently asking themselves, 'How do we integrate our application software, including our various "legacy" or custom-built systems?'. The most-favoured solution at present is to use so-called enterprise application integration (EAI) software from vendors such as WebMethods, BEA, IBM (MQ Series, WebSphere), and Tibco to create a middleware or EAI communication channel or bus that links the various applications. This situation is depicted in Figure 1.

Enterprise Application Integration (EAI)

The second issue for the future is inter-organizational process integration through exchanges, hubs, and eMarketplaces (see also Chapter 18). Whereas ERP systems support back office functions, such as financial, human resources, manufacturing, and logistics *within* an organization, its younger siblings, such

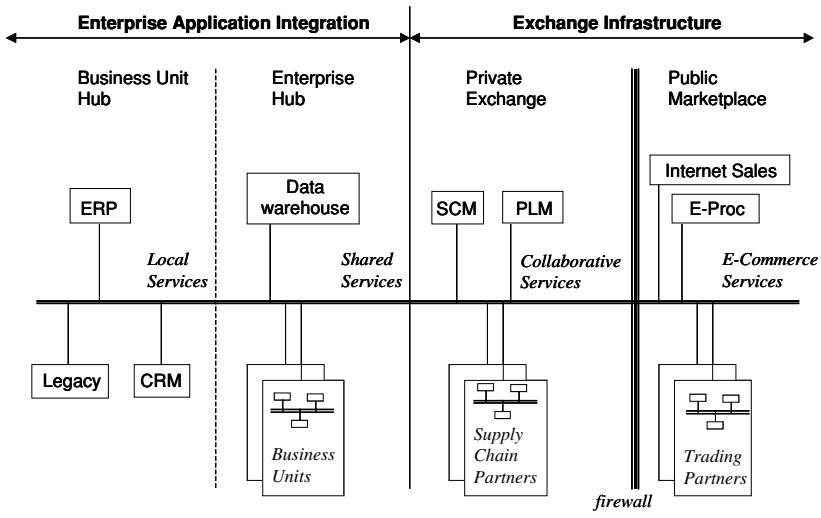

Figure 2 Possible future enterprise system hub and exchange architecture
(*Source:* Based on H. Kagermann, Sapphire Conference Presentation, Nov. 2001.)

as CRM and SCM, are more concerned with collaborative links to partner organizations up and down the supply chain. As noted earlier, developments in the area of customer-facing CRM systems, and supplier-facing SCM systems and exchanges are where major future growth in the use of packaged application software is expected. ERP systems provide a core facilitating infrastructure for many of these new applications. One possible future hub and exchange architecture that integrates the various components of a firm's enterprise system architecture is depicted in Figure 2. Moving from left to right, from deep inside the organization to the view from the outside, the figure depicts:

- local application services, not necessarily from the same vendor, being provided from each business unit hub;
- enterprise-wide shared services, possibly including financial accounting, human resources, data warehousing, and analytic services being provided from the enterprise hub;
- links to business partners being provided through collaborative SCM and PLM services provided by a private exchange hub inside the corporate firewall;
- e-commerce services, such as internet sales and e-procurement, being provided to trading partners via a public marketplace exchange.

In this vision of the future, the enterprise application integration (EAI) bus from Figure 1 links services within the organization at both the business unit hub and enterprise hub (the two left-hand columns in Figure 2), and collaborative links to business partners might be provided using message-based technologies delivered over the internet, such as XML, and managed by exchange and marketplace software probably based around so-called 'web services'.

The third issue for the future is the use of shared services and application service provision. Wide area networks are now so fast and cheap that it really does not matter where in the world the application and database servers for enterprise software are located. For instance, in teaching with SAP R/3 at The University of Melbourne, we issue presentation layer software (SAP GUI) to students to install on their home computers and use an application service provider 2000km away to provide all ERP services. Students access the service from their homes through their own internet service providers. In return for a fixed sum per semester, Queensland University of Technology manages the entire technical infrastructure (R/3 version 4.6c running on Sun hardware with an Oracle database). The system works faultlessly. ERP services are delivered to the screen through the internet just as mains power is delivered through the power plug in the wall. It does not matter where in the world the application and database servers are located.

Various organizations have moved to cut the cost of back office services by centralizing accounts payable services in a single city where labour is cheap, rather than distributing the accounts payable function around the world. This can work on an internal ASP/hub model of shared services, as introduced globally, for example, by the oil conglomerate Shell during 2001, or as Hagel and Brown (2001) posit may involve a sourcing model incorporating in-house supply with a single third-party supplier, or multiple third-party suppliers, of applications, services, and infrastructure. The relevance of external sourcing models is that, as this book will make clear, ERP systems can be difficult and expensive to implement. From 2000, the ASP model has been offered on the promise of relatively pain free ERP outsourcing. Pain free in this context means cutting costs and offering a fixed fee per month, removing the complexity of owning and operating these systems in house, and bridging the ERP skills gap. However, users have found that they need to find out what level of support they are getting for their money. In fact by 2002 a broad range of services was on offer, ranging from basic hosting – in which the ERP application runs on the ASP's computers and is delivered over the Internet or leased lines – to managed services – in which the ASP maintains and upgrades the ERP

system – to complete installation, hosting, and maintenance (Kern, Lacity, and Willcocks, 2002).

Hosted and managed ERP services were originally intended to appeal particularly to mid-sized companies previously excluded by the cost and complexity of ERP. For leading ERP vendors, the ASP model has been seen, amongst other things, as a route to the smaller companies market. However, the ASP model depends on a one-to-many service. Thus US-based Host Logic founded itself as a one-size-fits-all managed service provider. Its 2000 product, Smart-Enterprise, was offered at an all-inclusive, fixed, monthly price, and included all the hardware, services, implementation, and software needed to install and run SAP R/3 in a mid-size company. It was delivered partly by Host Logic, and partly by partners dealing with, for example, servers, network services, load balancing, and encryption. However, such 'out-of-the-box' applications provide, at best, only 65–75% fit to enterprise requirements. Indeed the ASP model may well heighten customers' typical complaints about ERP: lack of flexibity, lack of customizability, and difficulty and cost of implementation. One likely further development then will be a continuum of ERP offerings based on the ASP, net-enabled delivery model, ranging from off-the-shelf, standardized systems to premium services supporting fully customized ERP systems (Moran, 2000; Kern, Lacity, and Willcocks, 2002). The issues surrounding this model's application to ERP implementation for effectiveness are discussed in detail in Chapter 4 (see also Chapter 18).

Having defined the ERP world, and given an overview of the major recent developments therein, we now move on to detail the precise focus, organization, and contents of this book.

Organization of the Book

We have organized the book into five parts, each dealing with a critical issue in the effective implementation of ERP systems. The first part provides an overview of implementation for effectiveness of ERP systems, and includes discussions about process models for ERP implementation, the reasons organizations have moved to ERP systems, problems encountered and success achieved during ERP implementation, and how to assess and manage benefits from ERP implementation. The second part focuses on critical success factors and risk management during ERP implementation and maintenance, and highlights the areas in which management should focus their attention during ERP implementation. The third part addresses organizational learning and

knowledge management issues in ERP implementation. Lessons learned from ERP implementations in both large and small to medium firms are presented, together with discussions about how knowledge about ERP systems may be represented, integrated, shared, and transferred. The fourth part provides an analysis of cultural issues in ERP implementation, and explains implementation problems using organizational and national culture characteristics that differ from values embedded within the ERP system. The final part discusses two possible scenarios that highlight very different futures for ERP systems.

Part I Implementation and Effectiveness: Overview

We selected four papers to provide an overview of implementation for effectiveness of ERP systems. In the first paper, 'Learning from Adopters' Experience with ERP: Problems Encountered and Success Achieved', Lynne Markus, Sheryl Axline, David Petrie, and Cornelius Tanis note that ERP packages touch many aspects of a company's internal and external operations. Consequently, successful deployment and use of ERP systems are critical to organizational performance and survival. They present the results of a study of problems and outcomes in ERP projects, conducted under the sponsorship of an ERP system vendor. Two basic research questions are addressed: First, how successful are companies at different points in time in their ERP experiences, and how are different measures of success related (that is, can early success be followed by failure and vice versa)? Second, what problems do ERP adopters encounter as they implement and deploy ERP, and how are these problems related to outcomes? Findings show that the success of ERP systems depends on when it is measured, and success at one point in time may be only loosely related to success at another point. Companies experience problems at all phases of the ERP system life cycle, and many of the problems experienced in later phases originated earlier but remained unnoticed or uncorrected. These findings suggest that researchers and companies do well to adopt broad definitions and multiple measures of success and pay particular attention to the early identification and correction of problems.

In his paper, 'Innovating with Packaged Business Software: Toward an Assessment', E. Burton Swanson notes that in the 1990s many firms turned to software packages when they replaced the older systems in their application portfolios. While packaged business software had already long been in use, it was in the 1990s with the advent of ERP that it began to dominate enterprise decisions throughout the world. This 'package transition', which continued