



LSCITS

Large Scale Complex IT Systems

White paper: Key issues in ERP system implementation

*Gordon Baxter
School of Computer Science
University of St Andrews*

15th February 2010

Contents

Introduction	3
The criteria for success (and failure)	4
Learning from the past	5
An alternative development lifecycle for ERP systems	6
The common factors that contribute to success	7
Conclusions	9
References	9

Introduction

When companies introduce new computer-based systems it is often part of a larger organisational change project. For many years, these systems were developed from scratch, either in-house, or by a company specialising in system development. This model of system development—sometimes referred to as greenfield development—has changed since the mid-1990s, with a move towards so-called brownfield development, where systems are constructed from existing components and applications (Hopkins & Jenkins, 2008) and where new systems have to integrate and inter-operate with a range of existing systems. One of the most widespread uses of brownfield development is where generic applications such as Enterprise Resource Planning (ERP) and Commercial off the Shelf (COTS) systems are configured to meet the organisation's requirements (Sommerville, 2008). Here we will focus our discussions around ERP systems, although many of the same arguments apply to system development based on other COTS packages.

The move towards the use of ERP systems partly arose out of the fact that somewhere between one half and three quarters of bespoke system development projects were being classed as failures. Companies viewed the switch to off-the shelf solutions (like ERP systems) as a way of containing costs and risks whilst also providing the added benefit of vendor support for the ERP product. They saw them as an opportunity both the share data and to standardise processes across the organisation. The promised benefits of a successful ERP implementation appear attractive to an organisation, because they include reductions in costs (inventory, raw materials and production), customer lead times and production times (Gunn, 1998, cited in Somers & Nelson, 2001).

ERP systems offer a one size fits all solution which provides a company-wide view of corporate information. The central notion underpinning ERP systems is that they encapsulate best business practice for a particular industry, integrating manufacturing, financial and human resource operations into a single framework. One of the main reasons for adopting ERP systems is the potential reduction in long-term costs, although in the short-term they tend to be higher because of the costs of implementation and staff training involved. Making a particular ERP system work effectively for an organisation often requires significant resources, particularly time and people. The amount of resources depends on how well the ERP system fits with existing (or planned) business processes (Hong & Kim, 2002): if business processes are closely aligned with the best practices model built into the ERP system the need for extra resources will be minimised. If there is not a close match, however, a process of mutual adaptation is needed in which the company may need to both adapt some of its business processes to align with those of the ERP system, and have the ERP system adapted to existing processes that cannot be changed. Adapting processes to the ERP may impose some rigidity on the processes which is not reflected in how work is carried out. In any eventuality an ERP package will usually not meet all of the business goals, so it is vital that plans are put in place to ensure that the other requirements are also satisfied.

In general, implementing an ERP system is not a simple task, usually because the companies where they are being deployed are large, and often comprised of several heterogeneous, distributed units. The diverse nature of the system stakeholders exacerbates the problems of ERP system development, because of their varied and often conflicting needs and requirements. It is therefore important that these so-called “soft” issues are addressed from the point where the project is initiated.

The rate of successful IT projects according to the latest (2009) edition of the Standish Group’s annual CHAOS report¹ 32%. These projects were on time, on budget and delivered the required level of functionality. However, 44% of the projects failed because they were either late, over budget or did not deliver the required functionality, and 24% of projects were cancelled prior to completion. These figures include ERP systems, although the figures for ERP systems are not presented separately.

So what can be done to improve the chances of success in ERP system development projects? We start by identifying some high level criteria for success. We then briefly consider the notion of learning from past experiences—both failures and successes—as a way of informing new development projects, and highlight some examples of high-profile ERP failures, and one of success. Making the system development process more systematic may help, so we then consider the idea of a project lifecycle for ERP system development, before finally considering what appear to be the critical factors for successful projects.

The criteria for success (and failure)

Whether a system is branded as a success or a failure is a judgement, usually made at some point in time by one or more people with the benefit of hindsight. Most people, for example, consider that the new system at Heathrow airport’s Terminal 5 was a failure when it opened in 2008. Now, however, that same system (more accurately, system of systems) is operating successfully with few reported problems on a day-to-day basis.

The judgement of whether a project has ‘failed’ is not a simple yes/no decision. It is common for systems that initially did not live up to expectations to evolve over time to deliver useful services. However, management usually regard a project tends as a success if it meets three high-level criteria:

- It should be delivered on time
- It should be delivered within budget
- It should deliver the expected functionality

In addition to these should be added the considerations of the users, to make sure that the system fits in with their everyday working:

¹Summarised at http://www.standishgroup.com/newsroom/chaos_2009.php

- It should be acceptable to the users (and hence used).

If a project fails to satisfy one or more of these criteria when it is deployed, it is likely to be labelled a failure. If we closely examine the causes of system failures, we see that most of them are not attributable to failures of the technology. Instead, they are failures of the socio-technical system, often arising because the social and organisational aspects either have not been appropriately considered, or have been separated from the technological aspects. It is important that the social and technical aspects of the overall system are developed in parallel, because they are often interdependent. If they are developed separately, any mismatches may not be detected until late in the project when they are invariably expensive and time-consuming to correct, and can even lead to the project being abandoned.

Learning from the past

One way of trying to reduce the failure rates of ERP systems implementation projects is to build on lessons learned from other projects, both failures and successes. In the early days of ERP, failure rates were said to average about 70% (Gillooly, 1998). These failure rates have remained high, and there are several well-documented high-profile examples of major ERP failures including:

- Foxmeyer Drug Company (1996) who went into bankruptcy after having to abandon a \$40M ERP system after deployment.
- Hershey Foods Corporation (1999) who had problems deploying an ERP system which contributed to \$151M annual loss.
- Hewlett-Packard Company (2004) who also had problems with deployment of an ERP system which contributed to \$160M annual loss.
- Avis Europe PLC (2004) who cancelled an ERP system after having spent the equivalent of \$54.5M.
- Cadbury Schweppes PLC (2006) who had ongoing problems arising from implementing an ERP system in 2005 which led to a major product recall in 2006, contributing to a £12M loss, and also contributed to reported financial difficulties in 2007.

As well as these high profile failures, there are a vast number of anecdotal accounts of how ERP systems, which have cost many millions, have led to organisational disruption, with the end result being a system that delivers a much lower level of functionality than originally promised.

Failures get more widely reported than successes in the media, which explains why they have more frequently been the subject of analysis by researchers and industrial practitioners

alike to try to identify any common causes of failure. It is important to note that there are many successes too, such as Marathon Oil's Renaissance project (Stapleton & Rezak, 2004), which was initiated at the end of 1999 and went live early in 2002. It is worth pointing out at this juncture that the Marathon Oil explicitly looked at documented ERP failures so that it could avoid potential pitfalls during the Renaissance project.

An alternative development lifecycle for ERP systems

ERP systems are inherently implemented differently from bespoke systems. The word "implemented", although widely used in this context, is somewhat misleading. It is really the vendors who implement the ERP software, which the purchaser then has to *configure* for their organisation. Given that there are major differences between ERP systems development projects and bespoke development projects, it seems sensible that to suggest that they should be planned and executed differently. This idea has been captured by alternative lifecycle models for the development of ERP systems. There does not appear to be one best model, however, and many of them use a mixture of new and old terminology to name the phases within the lifecycle. This makes things somewhat confusing because the same name is used to describe different activities, such as "implementation" which really means configuration and adaptation in an ERP systems development project.

The model proposed by Esteves and Pastor (1999) is novel in that it includes dimensions as well as phases. The dimensions are really areas of concerns or viewpoints that need to be considered during each of the phases. These dimensions are:

- Product: the particular ERP system that has been chosen, its hardware and software.
- Process: how the business processes relate to the ERP system, and how those processes or the ERP system need to be adapted.
- People: making sure that the project team are appropriately educated and skilled to deal with the development as it proceeds, and that users are given the skills to allow them to work on the new system when it goes live.
- Change management: ensuring that the appropriate knowledge is in place to implement the organisational change in a timely manner and on budget.

The phases of the lifecycle that these dimensions apply across are:

- Adoption decision: this includes defining the system requirements, identifying the goals and objectives, analysing the impact of the new system and deciding which solution is best (whether that be ERP or not)
- Acquisition: this is essentially about making sure that the right ERP product is selected, i.e. the one that is the best fit for the organisation
- Implementation: this is really about customisation and adaptation based on the needs and requirements of the organisation. The adaptation of the ERP system and organisational processes is an iterative process that requires on-going social action

that is constrained by the company's structural organisation and the inherent properties of the ERP system. Adaptations can consist of configuration where processes and parameters are selected; extensions, such as adding extra software via built-in hooks; and modifications where the ERP code itself is changed (Hong & Kim, 2002).

- Use and maintenance: once the system is deployed, making sure it delivers the required results, and gets maintained to ensure that it continues to support the organisation's needs and requirements
- Evolution: this is where extra capabilities are added into the system, which may provide new benefits to the organisation
- Retirement: this is where a decision is made about replacement with a system (ERP or otherwise) that is better suited to the current needs of the organisation

There are clearly several similarities to lifecycle models for bespoke system development. The main difference is the acquisition phase. What is interesting, and perhaps somewhat surprising is the lack of any specific reference to risk management, particularly since the risks associated with ERP system development different, although well known by the experts.

The common factors that contribute to success

The idea of critical success factors is one that is well established in the field of information systems for many aspects of development, such as requirements analysis, planning, and project management. Their use in relation to ERP systems is comparatively new (Somers & Nelson, 2001), and Finney and Corbett's (2007) review highlighted the need for more work in this area to ensure that stakeholder viewpoints are taken into consideration, and to investigate the role of change management in more detail.

Several academics and practitioners have tried to capture the main reasons for the failure or success of ERP implementations (e.g., Ewusi-Mensah, 1997; Stapleton & Rezak, 2004; Weightman, 2004; Anexinet, 2006; Kimberling, 2006; Ibrahim, et al., 2008; Lindley, et al., 2008). Most of these analyses and lists focus on the factors that contributed to failures than those factors that contributed to success. If all of these factors are effectively normalised so that they are phrased as positive factors that contribute to success (e.g. *the lack of agreement on a set of goals and objectives* is rephrased as the need for *clear, agreed goals and objectives*), there are six factors that appear on most of the lists:

1. Top level management support for the project. It is important that there is clear, executive level support for the project, and that this support continues throughout the project.

2. User training and education. ERP systems invariably change the way that people work throughout the company, so it is important that the users are educated about the changes, and trained to use the new system.
3. Project management. Managing an ERP system implementation can be tricky, because there are often unanticipated events that increase the overall cost of the project. Making sure that the organisational structure and management are aligned can be particularly important where several subsidiary organisations are being brought together in the ERP system.
4. Project team competence. The project team needs to be made up of the people who are best suited to the job of implementing an ERP system, and ideally have experience of doing so.
5. Clear, agreed goals and objectives. It is important to build a business case for the ERP implementation. The business processes that need to be supported, and the requirements that the system needs to meet have to be specified. The way that the business processes are going to be aligned with the best practice model embedded in the ERP system needs to be identified. The measures for determining success should be laid out at the start of the project and be agreed by the system stakeholders.
6. Change management. It is crucial that changes are appropriately managed on two levels. First, there are the changes to the organisation. Second there are the changes that are made to the programme of implementation of the ERP system (which will be part of the larger organisational change project).

Ostensibly, these factors look like a list of important factors for the success of any development project, and this could be where some of the particular problems of ERP projects lie. So, for example, if people try to apply traditional (i.e., bespoke system development) project management techniques to an ERP system development project they may not work.

There are some other fundamental issues that also need to be considered. If the company decides that the solution is an ERP system, it is important to understand why. ERP systems, for example, can constrain how a company can change in the future, particularly when that company has adapted its business processes to fit with those of the ERP. Once the choice of solution has been made, the decision about which ERP software to buy can be considered. What is clear is that if a company does not get the adoption and acquisition phases right, then it will be much harder to make a success of the project as a whole even if all the identified factors for success are in place.

One other factor that is critical to success is communication. The scope of ERP systems is necessarily broad, so they will affect most, if not all, parts of an organisation. It is therefore important to ensure that all the available communication channels are used so that the project team, stakeholders and users can be kept up to date with all the changes in the organisation and the technology as they happen. Marathon Oil's successful Renaissance project (Stapleton & Rezak, 2004) used one-way communication (newsletters, a web site, road shows and so on), augmented by interactive sessions like workshops, conference calls and collaboration websites. They went even further, though, by including hands-on interaction so that people could validate concepts as they were being developed, and play with versions of the application as they were being developed. The two key aims of the communication can be summarised as:

- Increasing awareness and understanding (largely achieved through one-way communication)
- Gaining the commitment of the people involved (largely through two-way communication).

Conclusions

There is no silver bullet that can be used to kill off the potential for failure of ERP system development projects. The proportion of failures remains stubbornly high, even though several of the factors that are associated with failures appear to be known. A quick look at several of the "Top 10" style lists of factors associated with failures (and successes) reveals that no two lists are identical, although there are several factors that recur on many lists. The lack of agreement suggests that the analyses of the reasons for failure may be overgeneralising, by treating all failures as being more or less the same, whereas there are really different types of failure that arise through different combinations of factors.

There are two things that appear most evident from the literature. The first is that applying traditional methods to an ERP development project does not work. The second is that the earliest stages of the project are most critical. Developing a clear and accurate picture of the current state of the business and its processes is vital, and needs to be carried out before deciding whether an ERP system can provide the appropriate solution. It is also important factor to identify the potential risks before and after the ERP package is selected, so that they can be appropriately managed. Furthermore, it is crucial that an appropriate change management mechanism is put in place, and that the impact of any changes (to the organisation or the ERP system) on the project risks are adequately considered. Whilst these suggestions may not guarantee success, if they are not given appropriate consideration, the chances of failure are likely to be greater than they could be.

References

Anexinet, R. B. (2006). Top 10 ERP implementation pitfalls Retrieved 11th February, 2010, from http://www.anexinet.com/pdfs/ERP_top10pitfalls3-2006.pdf

- Esteves, J. M., & Pastor, J. (1999). An ERP life-cycle-based research agenda *Proceedings of the 1st international workshop in enterprise management and resource planning: methods, tools and architectures - EMRPS'99* (pp. 347-357). Venice, Italy.
- Ewusi-Mensah, K. (1997). Critical issues in abandoned information systems development projects. *Communications of the ACM*, 40(9), 74-80.
- Finney, S., & Corbett, M. (2007). ERP implementation: a compilation and analysis of critical success factors. *Business Process Management Journal*, 13(3), 329-347.
- Gillooly, C. (1998). Enterprise management disillusionment. *Information Week*, 16 February.
- Hong, K.-K., & Kim, Y.-G. (2002). The critical success factors for ERP implementation: an organizational fit perspective. *Information & Management*, 40, 25-40.
- Hopkins, R., & Jenkins, K. (2008). *Eating the IT elephant: Moving from greenfield development to brownfield*. Upper Saddle River, NJ: IBM Press.
- Ibrahim, A. M. S., Sharp, J. M., & Syntetos, A. A. (2008). A framework for the implementation of ERP to improve business performance: A case study. In Z. Irani, S. Sahraoui, A. Ghoneim, J. Sharp, S. Ozkan, M. Ali & S. Alshawi (Eds.), *Proceedings of the European and Mediterranean Conference on Information Systems (EMCIS 2008)*.
- Kimberling, E. (2006). 7 critical success factors to make your ERP or IT project successful Retrieved 11th February, 2010, from <http://it.toolbox.com/blogs/erp-roi/7-critical-success-factors-to-make-your-erp-or-it-project-successful-12058>
- Lindley, J. T., Topping, S., & Lindley, L. (2008). The hidden financial costs of ERP software. *Managerial Finance*, 34(2), 78-90.
- Somers, T., & Nelson, K. (2001). The impact of critical success factors across the stages of enterprise resource planning implementation *Proceedings of the 34th Hawaii international conference on system sciences*.
- Sommerville, I. (2008). Construction by configuration: Challenges for software engineering research and practice *Proceedings of 19th Australian Conference on Software Engineering* (pp. 3-11): IEEE.
- Stapleton, G., & Rezak, C. J. (2004). Change management underpins a successful ERP implementation at Marathon Oil. *Journal of Organization Excellence*, 23(4), 15-21.
- Weightman, C. (2004). The top 10 ERP mistakes. *Business Management*(February), 36-40.

LSCITS is the UK's national research and training initiative in the science and engineering of Large-Scale Complex IT Systems.

Leading British academics and industrial practitioners established this national strategic coordinated research and training initiative with a headline funding of over £15m. Research is being undertaken at a consortium of universities including Bristol, Leeds, Oxford, St Andrews and York.

The motivation for the LSCITS Initiative is the on-going growth in the size and complexity of information technology (IT) systems. Our ability to develop, maintain and manage such systems is falling behind the growth in their complexity. There is a high risk that we will find ourselves reliant on IT systems that we don't fully understand and that we cannot effectively manage.

We are addressing this risk at different levels of abstraction through the research of: Complexity in Organisations; Socio-Technical Systems Engineering; High Integrity Software Engineering; Predictable Software Systems; and Novel Computational Approaches.

