A paper presented to the
Department of Information Systems
University of Cape Town
in partial fulfilment of the requirements
for Enterprise Systems and BPM
(INF4012W)

Due: 7 March 2011

Experiences of putting the enterprise into an enterprise system

Factors critical to the successful implementation of an enterprise system

By Marc Pelteret (PLTMAR004)

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the APA convention for citation and referencing. Each contribution to, and quotation in, this paper from the work(s) of other people has been attributed, and has been cited and referenced.
3. This paper is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work
5. I acknowledge that copying someone else's assignment, essay or paper, or part of it, is wrong, and declare that this is my own work.

Signature: *Marc Pelteret* Date: *07/03/2011*

Full name of student: *Marc Pelteret*

Contents

1. Introduction

Enterprise systems are widely seen as the most important development in information technology for corporates in the 1990s (Vogt, 2002). They are designed to "enhance competitiveness by upgrading an organization's ability to generate timely and accurate information throughout the enterprise and its supply chain" (Umble & Umble, 2002). Enterprise resource planning (ERP) systems, which evolved from manufacturing systems developed in the 1970s (Vogt, 2002), are particularly popular among large corporations because they aim to replace multiple disparate systems with a single one. They promise instantaneous information on any part of the company to anyone who needs it, wherever they are (Minahan, 1998), thus allowing for quicker decision making and closer monitoring of the organisation's performance.

However, despite the fact that they have been around for the better part of two decades, enterprise system implementations still frequently go wrong. Kanaracus (2010), for instance, lists several of the biggest failures of 2010, many of which involve large, well-established companies and software vendors, millions of dollars, and years of wasted time. Some systems are not considered outright failures, but are not considered to be successes either, simply because they do not meet the organisation's expectations (Somers & Nelson, 2001).

The root of implementation problems is twofold. Firstly, implementing enterprise systems is technically challenging. They are complex pieces of software that take a large amount of time, money and expertise to implement (Davenport, 1998). The Panorama Consulting Group (2010) reports that on average it takes 18.4 months and $6.2 million, or 6.9% of total revenue, to implement an ERP system. Secondly, implementing an enterprise system has business implications – and this is where the problems most often occur: "companies fail to reconcile the technological imperatives of the enterprise system with the business needs of the enterprise itself" (Davenport, 1998).

This paper aims to explore the factors that are critical to the successful implementation of an enterprise system. Though various models are suggested for the phases of the implementation of an enterprise system (such as in Berchet & Habchi (2005)), for the purposes of this paper the process is viewed as consisting of three generalised stages: pre-implementation, during the implementation phase itself and post-implementation. The critical success factors of each are discussed in the next three sections and the paper concludes with a review of the factors examined.

2. Pre-implementation

2.1. Project team formation and organisational commitment

The project team should consist of a variety of experts. There should be people who understand the business and its strategies, as well as those with IT knowledge (Grossman & Walsh, 2004; Poon & Yu, 2010). The firm should be willing to dedicate some of its best employees to the project, even though they may no longer be available to perform their regular duties (Bingi, Sharma, & Godla, 1999; Vogt, 2002). Experience in implementing an enterprise system is helpful but not a necessity, as external consultants can provide this expertise.

It is critical that the project be seen as an enterprise-wide venture and not just an IT project (Minahan, 1998), so commitment from top management is particularly important – in fact, various studies indicate that it is the most important factor in achieving implementation success (Bingi et al., 1999; Ehie & Madsen, 2005; Somers & Nelson, 2001) . Ideally the team should be led by a senior executive who “possesses the strategic business vision” (Poon & Yu, 2010) and can champion the project (Ngai, Law, & Wat, 2008; Umble & Umble, 2002).

2.2. Business case definition

The first and most fundamental question for the project team to ask is: does the company need an enterprise system? The business’s requirements have to be determined and it needs to be decided whether implementing an enterprise system is the best route for the organisation. The system has to benefit the organisation in a definable, preferably measurable, way. Once it has been implemented, going back is extremely difficult because of the changes the new system brings (Bingi et al., 1999).

The cost of the system needs to be weighed against the intended gains (Davenport, 1998). If the company competes primarily on cost, the cost of an enterprise system may erode the company’s advantage. It can be difficult to define success in measurable terms, and the definition of success may even change during the lifetime of the project (M. L. Markus, Axline, Petrie, & Tanis, 2000). However, some sort of definition is needed in order to later determine whether the project has been successful and measure its worth.

2.3. Third-party integrator selection

Ideally, an experienced partner is needed to help with the implementation. This partner needs to be carefully chosen because they will often bring in the vendor (Binstock, 2010). Ideally, they should have experience in the company’s industry, but more importantly, it needs to be ensured that they have access to experienced staff and intend to use them. Consulting companies often hire new graduates and use them on projects, allowing them to learn on the job at the expense of the client (Grossman & Walsh, 2004).

2.4. Software package selection

An often-encountered problem is misalignment between the business and the chosen software package. Fundamentally, the structures embedded in the software may not match those embedded in the organisation. The software vendor draws on their own knowledge, experiences and norms when writing software (Soh & Sia, 2004), and decides what constitutes a best practice (Davenport, 1998). As such, the software may be based on operations in a country, region or industry that fundamentally differ to that of the organisation. The organisation has its own embedded structures (Soh & Sia, 2004), some of them involuntary (like laws and regulations imposed by government, and standards and guidelines by professional bodies), others voluntary (such as organisational experience and perceptions, and general preferences). These may not be accounted for by the software. Culture may also be an issue: Davison (2002), for instance, discusses how issues that arise in in Eastern culture, such as empowerment and the meaning of numbers, are not addressed in software that is produced in the West.

A package that best fits the company's business practices and processes should be selected. If there is a gap between the company's requirements and the features of the software, there are two options: customise the software or modify the organisation's practices.

Enterprise systems are designed for some degree of configuration, typically using optional modules and configuration tables (Davenport, 1998). There is also the option of integrating the software with external systems. But this may not be enough and there may be the need to customise the software by re-writing a portion of it. This is almost universally advised against (Panorama Consulting Group, 2010; Vogt, 2002), as it requires all the time, effort and cost of producing new software, can cause problems when integrating the system with external systems (such as those of customers and suppliers), and may cause additional problems when the vendor releases updates for the system (or, at the very least, the changes will have to be re-tested every time the vendor releases an update).

Modifying the business's practices to align them with the software is recommended by many authors (Benders, Batenburg, & van der Blonk, 2006; Davenport, 1998); others advise against it (Grossman & Walsh, 2004). The impact this may have on the organisation is discussed in section 2.8. Such changes can be beneficial, as they introduce best practices and other improvements that may streamline and enhance the company, but they may also be detrimental, negatively affecting workflow and competitive advantage (Davenport, 1998; Soffer, Golany, & Dori, 2005).

The ideal scenario is to choose a package with the smallest gap and highest degree of fit in order to minimise the need for either customisation or business process changes (Ngai et al., 2008). In practice, a little of both may be required.

2.5. Vendor selection

As important as the software package is the vendor that provides it. When the organisation buys the package it joins the software's network – the users and producers that "share a common interest in its destiny" (Damsgaard & Karlsbjerg, 2010), and want to protect their investments and ensure the software's evolution. The network should offer good long-term benefits as the organisation, network and package grow. The availability of adequate documentation and competent personnel skills are important parts of the network.

It is crucial to beware of vendor promises (Grossman & Walsh, 2004). The vendor's sales-people may unilaterally promise new features in future software versions and overstate the software's capabilities. The integration partner should be able to advise on this (Binstock, 2010), but as a general precaution all promises should be put in writing and the project team should make sure it understands the software's specifications.

2.6. Infrastructure and other third-party vendors

Issues surrounding infrastructure requirements and integration with third-party software need to be addressed (Poon & Yu, 2010). This will require collaboration, as the vendor knows the software's demands on hardware and the network (Grossman & Walsh, 2004). Similarly, vendors of third-party are familiar with their products' needs.

2.7. Negotiation and legalities

Despite what may be said, vendor contracts can be negotiated. Grossman & Walsh (2004) mention several issues to be aware of:

- There should be a separate contract between the organisation and the system vendor, and between the organisation and third-party integrator.
- Ensure that all promises are put in writing.
- Include detailed specifications in the contracts.
- Examine limitation of liability clauses. The majority limit the vendor's liability to the amount that the organisation has aid to them; the problem is that total costs for a project, including internal costs (such as staff time and lost revenues), are usually far more.
- Deal with the issue of arbitration versus litigation. Vendors prefer to use arbitration to resolve disputes, but this may not be in the organisation's best interests. Linked to this is the issue of where litigation would take place – vendors prefer to work under their local law rather than the organisation's local law.
- Negotiate pricing. Despite the general view, vendors are prepared to be flexible. Not only is the cost negotiable, but so is the payment schedule.

- Address the subject of who owns software modifications. Though the organisation might pay for modifications, the company that makes them may own the copyright. The vendor may also have rights to these modifications.

2.8. Impact on the organisation

An enterprise system will have a major impact on many of the organisation's internal functions. These impacts need to be understood and planned for right from the beginning of the project (Binstock, 2010).

The decision to modify the company's processes to align itself with the software to be implemented can lead to major changes in the firm's organisation and culture. The ability to access operational and financial data in real-time can allow the company to streamline management structures, but at the same time it involves centralising the control of data and the standardisation of data, qualities more consistent with hierarchy (Davenport, 1998). Some executives use the implementation of an enterprise system to force change and introduce consistency in the company's procedures; others use it to break down hierarchy and allow more innovation and flexibility (Davenport, 1998; Le Loarne, 2005). Whatever model is chosen to be introduced, business processes should be re-engineered before the software is implemented in order to avoid inappropriate software modifications and gain the maximum benefit from the software (M. L. Markus et al., 2000). Special care should be taken to avoid implementing existing redundant or non-value-adding systems – the integrated environment of the new system means that the organisation should to look at how it can do business differently (Minahan, 1998; Umble & Umble, 2002).

An enterprise system can also be to introduce uniformity across regions (Davenport, 1998). But this needs to be carefully considered because of the differences in local markets and the possible erosion of local competitive advantage by introducing such uniformity. There are also local customer requirements and regulations to consider. The company's international strategy will determine the system's configuration and structure (Madapusi & D'Souza, 2005). A firm with a multinational strategy, which operates in and is sensitive to different national markets, would implement local versions of the system, allowing each region to manage its own information and link to headquarters through financial reporting structures. For a global strategy, which aims at global efficiency and cost minimisation, a central system that national units access through interfaces would be used. A transnational strategy is a hybrid, focussing on local responsiveness and global efficiency, and would have a mixed implementation.

When multiple legacy systems are replaced by a single monolithic one, data and workflow issues arise (Soh, Kien Sia, Fong Boh, & Tang, 2003). Data from different sections of the company need to be merged and consistency issues need to be dealt with. Data ownership and entry become issues because the different sections all access this common data. Given that data will flow far more easily throughout the company, the effects of inaccurate data can be magnified (Bingi et al., 1999).

3. During Implementation

3.1. Project Management

In a project as large and as complex as an enterprise system implementation, strong project management is required. “Excellent project management against a project plan with clear objectives, deliverables, and milestones ensures that the project is effectively planned and delivered” (Ngai et al., 2008). The goals, scope and schedule of the project need to be clear, and the deadlines realistic. Many organisations underestimate the amount of resources, time and external assistance they will require (Umble & Umble, 2002).

The project needs to be actively managed (Binstock, 2010). Details are important. Progress needs to be assessed against the planned schedule and team members need to be regularly updated on this progress and any problems encountered. Scope is particularly important to manage, not only to avoid cost and schedule overruns, but also to ensure that it is not overly broad or ambitious. Cutting features due to time constraints can lead to expectations not being met (M. L. Markus et al., 2000; Motwani, Subramanian, & Gopalakrishna, 2005), but pushing out the schedule can also lead to problems and delays may in fact indicate that a larger problem exists (Binstock, 2010).

Despite the fact that consultants may have more experience in enterprise system implementation, the company should control each phase of the project (Somers & Nelson, 2001). Because of the importance of the project, top management should be involved throughout to provide direction, resolve conflicts, unify the organisation, and ensure that the implementation meets their vision (Bingi et al., 1999). Mid-level management should also be involved with detailed aspects of the implementation and should facilitate communication between the implementers and end users (Umble & Umble, 2002).

3.2. Multi-site issues

If the company has a complex organisational structure and is spread out all over the world, multi-site issues need to be addressed. There are four dimensions to be considered (M. L. Markus, Tanis, & van Fenema, 2000): business strategy, software configuration, the technology platform and the practical execution, all of which are interdependent.

Business strategy was discussed in section 2.8: the company can follow a multinational, global or transnational strategy.

Configuring enterprise software involves “one or many legal-financial entities and one or many operational entities” (M. L. Markus et al., 2000). For instance, the company might have multiple operational entities (e.g., sales and distribution units) but have only one set of financials.

Conceptually, the architecture of a multi-site technology platform can be viewed as having two extremes. In the one, the database and servers reside in a central location which is accessible via remote access; in the other, they are distributed to various locations. Centralised architectures are easier and cheaper to configure but issues to do with database size and performance, telecommunication costs and policies, maintenance costs, risk management, and the autonomy of local management may arise. Distributed architectures also have their own set of issues: data replication, response times and support costs.

Finally, the execution of the implementation, a complex issue, has to be dealt with. Multiple locations can mean dealing with different managerial reporting lines, spoken languages and cultures. Decisions can be made centrally or locally, and events can be organised to happen all at once in a “big-bang” approach or incrementally in a phased approach.

The big-bang approach is appealing because it involves an intense period of change that is short when compared to that of the phased approach, condensing the difficulty of the implementation into a short period of time and allowing for resources to be freed up more quickly (Kimberling, 2006). However, it is often rushed and many defects may remain hidden until the system is live and in use, even if thorough testing has been done (Vogt, 2002). Dealing with these defects while trying to run the system can be taxing.

The phased approach involves a step-by-step implementation, either according to functionality (different modules of the software) or operational site (Kimberling, 2006). It is safer because it makes fewer changes at once, thus minimising the number of potential defects that might be introduced, and it allows the implementers to learn as the project progresses (Vogt, 2002). However, it takes more time and can be draining on employees as it involves constant changes over a long period.

Both approaches have pros and cons, so the most appropriate one needs to be chosen for a particular implementation. A balance between the two may be best.

3.3. Change management and training

Given that the system will introduce major changes, resistance to it is inevitable. People naturally prefer to preserve the *status quo* because they understand it; a new system, particularly one as invasive as an enterprise system, can bring out fears about competency, job importance and job safety (Umble & Umble, 2002).

Resistance to change needs to be anticipated, identified and dealt with. Employees need to be convinced that the system will benefit the company and make their jobs easier. Implementing the system in a phased approach can help with this as it gives people time to adjust to the changes. Good communication of the strategic goals, expectations and deliverables, communicated by the firm's management, will also help dampen down resistance (Ngai et al., 2008; Umble & Umble, 2002).

Training is an important part of change management and should ideally begin before the project does, and continue during and after the implementation (Bingi et al., 1999; Umble & Umble, 2002). It aids employees in understanding the concepts of the system, which ensures their readiness and improves the chance of them accepting it (Mandal & Gunasekaran, 2003; Ngai et al., 2008). For training to be truly effective, it needs to cover not only the new system, but also the organisation's business practices and processes.

Anyone who uses the system or has some connection with it, including top management, should be trained so that they understand how the system integrates with the company's operations and to ensure they are able to take full advantage of its capabilities (Umble & Umble, 2002). If employees do not understand how to use the system, they will bypass it or use it inefficiently, and full benefit of the system will not be realised.

The cost of training must not be underestimated – it is often as expensive as the system itself (Grossman & Walsh, 2004) and needs to be included in the project's budget and plan. It is also takes a significant amount of time because the system is complex and cannot be taught quickly, hence the need for continual training (Bingi et al., 1999).

Like training, change management must not end with the project, as the issues faced during it may still exist afterwards (Binstock, 2010).

3.4. Testing

Given the complexity of enterprise systems, testing is an essential issue. But it is often neglected. When the schedule gets tight it, like scope and training, gets cut down. The temptation to do so needs to be resisted because the consequences of errors and the cost of correcting them are far greater once the system is in use.

There are a number of areas where testing is likely to be deficient (M. L. Markus et al., 2000): cross-module integrations, interfaces with legacy systems, modifications to the enterprise software, and unusual scenarios and scenarios involving the input of invalid data. It must not be presumed that third-party work, such as that from the integrator, has been properly tested and is flawless. Another often-neglected area of testing is stress testing (Grossman & Walsh, 2004). The system has to be able to function under the load of a live work environment. Stress testing will reveal flaws in not only the software but the network and infrastructure too.

3.5. Data management

Data conversion is a critical factor when an enterprise system is replacing legacy systems. The new system's data requirements and formats need to be determined, and the mapping and conversion of this data is an important task in the project plan (Ngai et al., 2008). Often data in legacy systems is of poor quality because of historical issues, such as extemporisation by, the users who have specialised knowledge and are able to interpret it (M. L. Markus et al., 2000), with little thought having been given as to whether it has been inputted properly. However, in an environment where information is common and shared, such data can cause problems. Users need to be trained on how to correct flawed data.

4. Post-implementation

4.1. Post-implementation audit

A post-implementation audit is performed to determine how successful the implementation is. Nicolaou (2004) proposes five areas that should be looked at:

1. Overall project scope and planning: review how well the system fits in with the organisation's vision and how effective the project planning was, and evaluate the implemented information infrastructure.
2. Driving principles for project development: review these principles and determine whether they justified the project.
3. Misfit resolution strategies: review the strategies used to deal with software and organisation alignment issues.
4. Attained benefits: evaluate if and how well the expected benefits were realised.
5. Learning: review the effectiveness of training and knowledge transfer between project team members and other users.

Users tend to have high expectations of enterprise systems. But flaws and defects in the system can diminish this enthusiasm, especially during the period immediately after implementation (Vogt, 2002). For this reason, user feedback alone is not a good enough measure of success. A dampening of enthusiasm and the fact that it will take users time to adapt to and learn the system needs to be taken into account when analysing the project's success.

Metrics can help ascertain how successful the project is. By establishing baseline measures while determining the business case, it is possible to measure performance before and after implementation (Kimberling, 2007). This will assist in identifying areas that are under-performing and opportunities for improvement.

4.2. Support

Disbanding the project team immediately upon project completion is not advisable (M. L. Markus et al., 2000). Instead, the team members should be kept together to provide support and assist with enhancements. They will have become valuable experts while working on the project and their expertise should not be lost.

The exact nature of the support structure needs to be decided. The team members could either become permanent members of an enterprise system team (a centralised structure) or they could return to their functional roles but still assist with enhancements (a decentralised structure) (Gallagher & Gallagher, 2010). It may be difficult to keep some members, as not everyone enjoys being in a supporting role (M. L. Markus et al., 2000).

4.3. Refinement and enhancement

The implementation is unlikely to be perfect. Not only is it likely that new issues will be discovered, but it is also likely that as users become accustomed to the system and accept it they will think of ways to improve it (Musaji, 2004). Requests for changes need to be managed (Berchet & Habchi, 2005): each request needs to be examined (looking particularly at who it originated from, whether it is a duplicate of another request, and whether the requested functionality already exists), assigned a priority, and analysed for feasibility.

It is possible to include functionality to monitor usage of the system in terms of which functionality is used, how often and by whom (Musaji, 2004). This will aid in identifying unused sections of the system, as well as users who are having problems and require assistance.

Enhancements and refinements need to be made using the same methodology and level of strictness that were applied during the implementation in order to avoid introducing new problems and breaking existing functionality (Musaji, 2004). This includes having proper planning, documentation and testing.

5. Conclusion

The implementation of an enterprise system is a complex task and there are a variety of technical and business factors that need to be considered before, during and after such a project.

One of the key factors is the parties involved. The organisation needs to be committed to the project and the project team needs to be strong and have a variety of expertise. The third-party integrator and vendor need to be carefully selected, as they play integral roles throughout the life of the project.

Another key factor is the chosen software package. Ideally, it needs to be closely aligned with the company's business policies and processes. If it is not, either it must be customised or the business processes have to be changed. Realistically, both may have to happen, but while extensive business process re-engineering is acceptable, massive customisation is widely advised against because of a number of complications it can cause. If process re-engineering does take place, the project team and organisation's management have to be aware of the impact it could have on the organisation and be prepared to deal problems that arise from it. Change management is critical: given the current turbulence and uncertainty in the business world, a lack of it can create a major obstacle in the successful implementation of a system (Panorama Consulting Group, 2010).

Effective and efficient project management is critical during the implementation. The organisation has to accept full responsibility for and actively manage every phase in the implementation. Deadlines and budgets need to be adhered to, and issues to do with implementing the system at multiple sites planned for and dealt with.

Training is vital to the success of an enterprise system because "without adequate training, a system can never be used properly, nor can it ever achieve the returns that were projected" (Grossman & Walsh, 2004). It should begin while the system is being implemented and continue afterwards in order to maintain standards.

In the post-implementation phase of the project, an audit of the project will help to determine how successful it was and review issues that arose during it. The project team should not be disbanded during this phase, but rather maintain and enhance the system, as well as provide support for the system's users.

As Bingi et al. (1999) wrote, implementing an enterprise system "is a careful exercise in strategic thinking, precision planning, and negotiations with departments and divisions". It is vital that organisations be aware of the factors discussed in this paper and consider them carefully, as dealing with them correctly will help to ensure the successful implementation of the system and the full realisation of the benefits it offers.

6. References

Benders, J., Batenburg, R., & van der Blonk, H. (2006). Sticking to standards; technical and other isomorphic pressures in deploying ERP-systems. *Information & Management, 43*(2), 194-203.

Berchet, C., & Habchi, G. (2005). The implementation and deployment of an ERP system: An industrial case study. *Comput.Ind., 56*(6), 588-605.

Bingi, P., Sharma, M. K., & Godla, J. K. (1999). Critical issues affecting an ERP implementation. *Information Systems Management, 16*(3), 7.

Binstock, A. (2010). *ERP gone bad: Lessons from real-world failures.* Retrieved February 27, 2011, from http://www.infoworld.com/d/applications/erp-gone-bad-lessons-real-world-failures-150

Damsgaard, J., & Karlsbjerg, J. (2010). Seven principles for selecting software packages. *Commun.ACM, 53*(8), 63-71.

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. *Harvard Bus.Rev., 76*(4), 121-131.

Davison, R. (2002). Cultural complications of ERP. *Commun.ACM, 45*(7), 109-111.

Ehie, I. C., & Madsen, M. (2005). Identifying critical issues in enterprise resource planning (ERP) implementation. *Computers in Industry, 56*(6), 545-557.

Gallagher, K. P., & Gallagher, V. C. (2010). An exploratory study of organizing structures for post-implementation ERP. *System Sciences (HICSS), 2010 43rd Hawaii International Conference on,* 1-10.

Grossman, T., & Walsh, J. (2004). Avoiding the pitfalls of erp system implementation. *Information Systems Management, 21*(2), 38-42.

Kanaracus, C. (2010). *Biggest ERP failures of 2010.* Retrieved February 21, 2011, from http://www.infoworld.com/d/applications/biggest-erp-failures-2010-741

Kimberling, E. (2006). *ERP's big bang theory.* Retrieved March 6, 2011, from http://it.toolbox.com/blogs/erp-roi/erps-big-bang-theory-11954

Kimberling, E. (2007). *The importance of ERP post-implementation audits.* Retrieved March 6, 2011, from http://it.toolbox.com/blogs/erp-roi/the-importance-of-erp-postimplementation-audits-14971

Le Loarne, S. (2005). Working with ERP systems—Is big brother back? *Computers in Industry, 56*(6), 523-528.

Madapusi, A., & D'Souza, D. (2005). Aligning erp systems with international strategies. *Information Systems Management, 22*(1), 7-17.

Mandal, P., & Gunasekaran, A. (2003). Issues in implementing ERP: A case study. *European Journal of Operational Research, 146*(2), 274.

Markus, M. L., Axline, S., Petrie, D., & Tanis, S. C. (2000). Learning from adopters' experiences with ERP: Problems encountered and success achieved. *Journal of Information Technology (Routledge, Ltd.), 15*(4), 245-265.

Markus, M. L., Tanis, C., & van Fenema, P. C. (2000). Enterprise resource planning: Multisite ERP implementations. *Commun.ACM, 43*(4), 42-46.

Minahan, T. (1998). Enterprise resource planning: Strategies not included. *Purchasing, 125*(1), 112. Retrieved from http://www.pages.drexel.edu/~arinzeob/ERPSCM/ERP3.pdf

Motwani, J., Subramanian, R., & Gopalakrishna, P. (2005). Critical factors for successful ERP implementation: Exploratory findings from four case studies. *Computers in Industry, 56*(6), 529-544.

Musaji, Y. (2004). ERP postimplementation problems. *Information Systems Audit and Control Association (ISACA) Journal (JOnline), 4*, 2011/03/06. Retrieved from http://www.isaca.org/Journal/Past-Issues/2005/Volume-4/Pages/JOnline-ERP-Postimplementation-Problems.aspx

Ngai, E. W. T., Law, C. C. H., & Wat, F. K. T. (2008). Examining the critical success factors in the adoption of enterprise resource planning. *Computers in Industry, 59*(6), 548-564.

Nicolaou, A. I. (2004). Quality of postimplementation review for enterprise resource planning systems. *International Journal of Accounting Information Systems, 5*(1), 25-49.

Panorama Consulting Group. (2010). *Panorama consulting group 2010 ERP report*. USA: Panorama Consulting Group.

Poon, P., & Yu, Y. T. (2010). Investigating ERP systems procurement practice: Hong kong and australian experiences. *Information & Software Technology, 52*(10), 1011-1022.

Soffer, P., Golany, B., & Dori, D. (2005). Aligning an ERP system with enterprise requirements: An object-process based approach. *Computers in Industry, 56*(6), 639-662.

Soh, C., Kien Sia, S., Fong Boh, W., & Tang, M. (2003). Misalignments in ERP implementation: A dialectic perspective. *International Journal of Human-Computer Interaction, 16*(1), 81-100.

Soh, C., & Sia, S. K. (2004). An institutional perspective on sources of ERP package–organisation misalignments. *The Journal of Strategic Information Systems, 13*(4), 375-397.

Somers, T., & Nelson, K. (2001). The impact of critical success factors across the stages of enterprise resource planning implementations. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 8 - Volume 8,* 8016.

Umble, E. J., & Umble, M. M. (2002). Avoiding ERP implementation failure. *Industrial Management, 44*(1), 25.

Vogt, C. (2002). Intractable ERP: A comprehensive analysis of failed enterprise-resource-planning projects. *SIGSOFT Softw.Eng.Notes, 27*(2), 62-68.