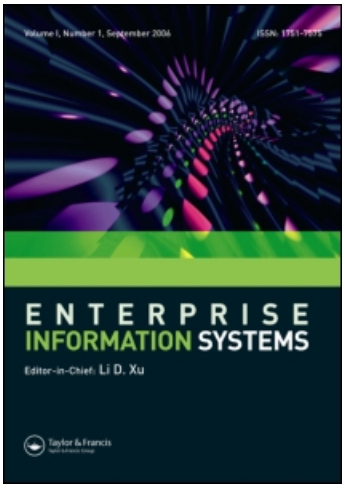

Enterprise Information Systems
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t748254467

A case study of exploiting enterprise resource planning requirements

Nan Niu[a]; Mingzhou Jin[b]; Jing-Ru C. Cheng[c]
[a] Department of Computer Science and Engineering, Mississippi State University, Mississippi State, MS, USA [b] Department of Industrial and Systems Engineering, Mississippi State University, Mississippi State, MS, USA [c] Information Technology Laboratory, U.S. Army Engineer Research and Development Center, Vicksburg, MS, USA

First published on: 24 January 2011

To cite this Article Niu, Nan , Jin, Mingzhou and Cheng, Jing-Ru C.(2011) 'A case study of exploiting enterprise resource planning requirements', Enterprise Information Systems, 5: 2, 183 — 206, First published on: 24 January 2011 (iFirst)
To link to this Article: DOI: 10.1080/17517575.2010.519052
URL: http://dx.doi.org/10.1080/17517575.2010.519052

Enterprise Information Systems
Vol. 5, No. 2, May 2011, 183–206

Taylor & Francis
Taylor & Francis Group

A case study of exploiting enterprise resource planning requirements

Nan Niu[a]*, Mingzhou Jin[b] and Jing-Ru C. Cheng[c]

[a]*Department of Computer Science and Engineering, Mississippi State University, Mississippi State, MS 39762, USA;* [b]*Department of Industrial and Systems Engineering, Mississippi State University, Mississippi State, MS 39762, USA;* [c]*Information Technology Laboratory, U.S. Army Engineer Research and Development Center, Vicksburg, MS 39180, USA*

(*Received 26 May 2010; final version received 24 August 2010*)

The requirements engineering (RE) processes have become a key to conceptualising corporate-wide integrated solutions based on packaged enterprise resource planning (ERP) software. The RE literature has mainly focused on procuring the most suitable ERP package. Little is known about how an organisation exploits the chosen ERP RE model to frame the business application development. This article reports an exploratory case study of a key tenet of ERP RE adoption, namely that aligning business applications to the packaged RE model leads to integral practices and economic development. The case study analysed a series interrelated pilot projects developed for a business division of a large IT manufacturing and service company, using Oracle's appl1ication implementation method (AIM). The study indicated that AIM RE improved team collaboration and project management experience, but needed to make hidden assumptions explicit to support data visibility and integrity. Our study can direct researchers towards rigorous empirical evaluations of ERP RE adoption, collect experiences and lessons learned for practitioners, and help generate more effective and mature processes when exploiting ERP RE methods.

Keywords: enterprise resource planning; requirements engineering; Oracle; application implementation method; exploratory case study

1. Introduction

The enterprise resource planning (ERP) system is an enterprise information system designed to integrate and optimise the business processes (BPs) and transactions in a corporation (Moon 2007, Xu 2007). ERP aims to enhance organisational cross-functional efficiency and effectiveness through the seamless integration of all the information flowing through a company (Davenport 1998). Major business drivers behind ERP implementations are: improving productivity, providing competitive advantage and satisfying customer demands.

The current generation of ERP systems provides packaged software solutions, as well as the reference models or process templates that claim to embody the current best business practices (Kumar and van Hillegersberg 2000). With pre-packaged

*Corresponding author. Email: niu@cse.msstate.edu

ISSN 1751-7575 print/ISSN 1751-7583 online

DOI: 10.1080/17517575.2010.519052
http://www.informaworld.com

ERP components, the functionality that is desired can be accessed immediately, and developed and tested by someone expert in that functionality.

Transitioning to the ERP mindset is not without risk, since an ERP transition is not only an information technology (IT) project but also a business decision. ERP systems are complex pieces of software. Consequently, many such implementations have been difficult, lengthy and over budget, were terminated before completion, and failed to achieve their business objectives even a year after implementation (Peterson *et al.* 2001). Thus, the effective use of ERP software requires a systematic method that facilitates the exploitation of the benefits of ERP functionality while guarding against the technical and business pitfalls. From the software engineering perspective, we can better understand and optimise the ERP system if the method is applied during requirements engineering (RE) activities (Basili and Boehm 2001).

RE is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used (Nuseibeh and Easterbrook 2000). Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies. Requirements are not only elicited according to stakeholder needs but also discovered based on technology innovations.

The conceptual picture of building a business application on top of a packaged ERP system is given in Figure 1. The picture involves two parties: the ERP vendor and the business application developer. Their respective perspectives are labelled outside and inside the parentheses in Figure 1. Blackbox and whitebox reuse aim to provide the organisation with increased software quality and productivity (Frakes and Kang 2005). It is crucial to determine the customised functionality and know how the adaptation will be accomplished. The unused part is a burden to the customer because the provided functionality is not actually needed. The grey-shaded area is unsupported because the desired functionality is not provided in the ERP package.

The real world picture is likely to be more complex than Figure 1. The intertwined connections between the modules present many risks for RE to assess. These include mismatches between business customer's needs and ERP features, interferences arising from unused components, and inconsistencies during integration of newly built portions. Therefore, the reuse benefits claimed for ERP systems

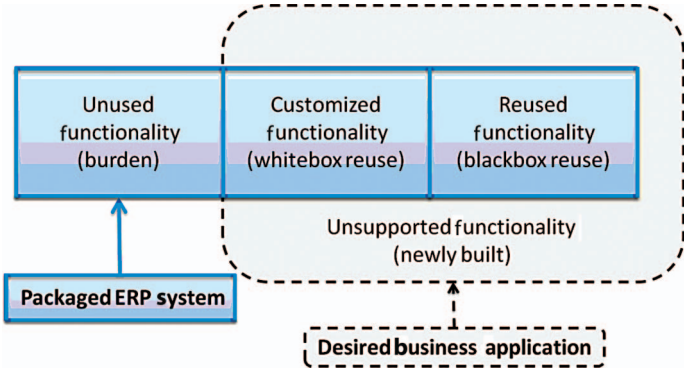

Figure 1. Exploiting ERP software in business application development.

have to be weighed against the cost of mitigating the risks. From Figure 1, it is clear that the fitness of ERP RE is concerned with reconciling the ERP system and the customer requirements. Fitness, in this context, has two criteria:

(1) *Choose what is fit*: procuring the ERP system that maximises the reuse to unused ratio;
(2) *Fit what is chosen*: tailoring the business application development so that customisation and extension are achieved.

The literature in ERP RE has focused mainly on the first aspect, namely how to select the most suitable ERP package. The effort includes matching ERP functionality to customer requirements (Rolland and Prakash 2001), modelling ERP package selection requirements (Maiden and Ncube 1998), employing use cases in a commercial off-the-shelf tender (Lauesen and Mathiassen 1999), guiding the acquisition and evaluation of off-the-shelf software (Ncube and Maiden 1999) and assessing ERP system's interoperability with legacy systems (Feblowitz *et al*. 1996). The fundamental issues of procuring software package requirements were outlined in Finkelstein *et al*. (1996).

In contrast, the RE literature has paid little attention to the second fitness criterion listed above, i.e. how to leverage the selected ERP package to construct high quality business applications. In practice, ERP providers are relatively easy to identify among market leaders, such as SAP AG, Oracle Corporation and Baan Co. Selection tends to be a less pronounced bottleneck as experience is accumulated and reported. The success of ERP projects, then, strongly hinges on how well they align the business requirements (BR) and application components to the selected ERP system. One solution is to take advantage of the RE guidelines defined in the chosen ERP package. However, we are aware of no empirical studies that investigate this ERP RE tenet, nor the scope of its applicability. The discipline remains under-explored. To address this gap, we conducted an evaluation of the underpinning tenet of ERP RE methods.

This article describes a case study that concerns the ERP applications, developed for a large IT manufacturing and service company, using Oracle's application implementation method (AIM). We examine the RE practices in three sequentially linked ERP projects. To allow for detailed evaluation, the projects adopted the same release of AIM. The adoption included not only artefacts but also processes. Thus, the study concentrated on merely the exploitation of an ERP RE method, and ignored assessing various ERP packages during the initial procurement phase. We set out to explore the tenet that using the chosen ERP method to frame the business application development would yield a better fitness of RE practice. We also wanted to gain insights into other aspects of the discipline. For example, what improvement is needed for AIM? How hard is it to adapt multiple applications within a generic ERP method?

2. ERP RE

ERP systems have two important features: firstly, they facilitate a causal connection between a visual model of BPs and the software implementation of those processes and secondly, they ensure a level of integration, data integrity and security, which is not easily achievable with multiple software platforms (Davenport 2000). ERP

software is not limited to specific industry sectors: it can be configured for retail industries, mining companies, banks, to name a few demands.

The RE community has tacitly assumed that contractual build-from-scratch development is the normal case in which there is a clear interface between customer and developer (Potts 1995). By considering off-the-shelf development too, RE establishes its relation to the rest of software and system development in a different perspective. Standard off-the-shelf RE processes have recently become a key to conceptualising any integrated, corporative-wide solution based on packaged ERP software (Daneva 2002). The main ERP RE activities include (Daneva 2004):

- *Requirements elicitation*: finding, communicating and validating facts and rules about the business.
- *Enterprise modelling*: analysing and representing BPs and data.
- *Requirements negotiation*: validating process and data architecture, resolving process and data issues, and prioritising requirements.

These activities focus on the initiation, adoption and adaption stages of ERP implementations (Rajagopal 2002), and are in line with the key RE activities discussed in Nuseibeh and Easterbrook (2000). In addition to gathering customer-specific requirements, RE for ERP systems must take into account that these systems are available off-the-shelf, which implies that:

- a generic software solution has already been selected by the vendor and is embodied in the ERP system;
- the success of ERP installation is critically dependent on the customising process;
- a number of organisational requirements have been elicited already to support the decision of acquiring this ERP system.

It is argued that a complete requirements specification is not needed for ERP RE. Instead, initial incomplete requirements can be detailed to restrict the solution space and converge towards the needed functionality (Rolland and Prakash 2001). A generic RE model helps accelerate such a convergence in that it offers defined processes, suggests process stakeholders, specifies steps to accomplish tasks, indicates task dependencies and provides standard tool support for ERP RE (Daneva 2004).

Essentially, an ERP RE process is about matching, reconciling and integrating. An organisation starts with a general set of BP and data requirements, and then explores standard ERP functionality to see how closely it matches the organisation's process and data needs. Procuring a matched package is only part of the puzzle. ERP projects demand huge resources and effort. The investment required to implement an ERP system at large companies typically ranges from $50 million to more than $500 million (Davenport 1998). Once the project decision is made, the initial resources are allocated, and the ERP package is selected, there is little room for retreat. Among 42 Australian-based ERP implementations surveyed in Parr and Shanks (2000), only one abandoned the project. Most companies felt impelled to continue once committed.

ERP RE, therefore, must not only consider the choice itself but also take full advantage of the choice in ensuing business application development. One

straightforward way is to adopt the RE guidelines defined in the ERP package. Fortunately, most market-leading ERP systems, such as SAP and Oracle, have established and documented detailed RE processes in their products. If we neglect the pre-packaged method and practise the traditional custom-built RE norm (Potts 1995) in ERP projects, the reuse ratio is reduced and so is the benefit. The whole point of exploiting ERP software is lost. However, this does not mean that we should copy the packaged RE process to our practice without any change. Instead, we must adjust, customise and sometimes create new RE guidelines, just as we must adapt packaged ERP functionality in desired business applications.

3. Study context

Despite a high adoption rate of ERP systems for replacing legacy systems and for pursuing business success, over 90% of ERP implementations exceed both their budget and the allocated time frame (Parr and Shanks 2000). Given that RE is any ERP project's most expensive stage (Holland and Light 1999), the RE knowledge is not only needed but also vital to the field. Our study investigates the issues arising when an organisation makes the ERP RE model a live process.

The organisation in our study is one of the largest PC manufacturers and IT service providers in the Asia-Pacific region. Its headquarters is located in Beijing, PR China, and it has approximately 23,000 employees as of 2008. To honour confidentiality agreements, we will call it ZT. ZT's products include desktops, laptops and mobile phone handsets. ZT also provides IT integration and support services. ZT's ERP implementations began in the late 1990s for strategic, operational and technical reasons. Among the major motivations for ZT to adopt ERP were the need for a common IT platform and the desire for process improvement, data visibility, operating cost reductions, increased responsiveness to customers and improvements in strategic decision-making.

ERP projects range from multi-site, substantial-resource comprehensive mode to less-ambitious, small-scale incremental mode (Parr and Shanks 2000). ZT took an incremental and evolutionary adoption strategy to reduce risk. In particular, pilot ERP projects were launched in one of ZT's business divisions, allowing a trial-and-error inquiry to collect experience and assess feasibility.

ZT selected Oracle's AIM (1999) to implement its pilot ERP projects. Although the discussion of ZT's selection details is out of the scope of this article, selecting AIM was based primarily on technical and organisational factors. In particular, ZT had used Oracle's platforms (e.g. Primavera Portfolio Management 1.9) and products (e.g. Oracle Database 8.1.5) to successfully implement many software-intensive projects. ZT wanted to continuously leverage Oracle's technological strengths, especially at a methodology level by adopting AIM where a set of guidelines can be tailored and applied to ZT's specific business situations. From the organisational perspective, ZT felt that Oracle's high-quality corporation image would contribute positively to its internationalisation strategy.

AIM is Oracle's project management methodology and implementation strategy. AIM is based on multi-track methodology, technology modernisation, business process re-engineering (BPR), Oracle application development, organisational change and project management (Oracle AIM 1999). In essence, AIM incorporates two things. First, it is a methodology showing what tasks are required, what order they should be completed, and what resources are required. Second, it provides

deliverable templates for all the tasks that require them. Hence the hybrid of methodology with a deliverable template tool makes AIM a powerful product. Although it seems a disadvantage for AIM to have a large number of deliverable templates,[1] one typically does not use all the available templates but chooses those that will be useful to a particular project. Figure 2 shows AIM's process overview. Along with the handbooks and user guide, AIM comes with an HTML-based document and process querying tool that allows the user to navigate through various pre-defined document templates. Figure 3 shows a snap-shot of the query interface.

The vertical rounded boxes in Figure 2 represent the six AIM phases, which provide quality and control checkpoints to coordinate project activities that have a common goal. AIM tasks are organised into processes, which are labelled to the left of Figure 2. Each process represents a related set of objectives, resource skill requirements, inputs and deliverable outputs. Of particular interest to our RE study are the following processes:

- BP *architecture* addresses understanding of the organisation's BPs and aligns them with the business application requirements to be implemented. It helps determine process data of the ERP project.
- *Business requirements definition* (RD) documents the business needs that must be met by the implementation project. It helps identify scenarios and events that reflect the application requirements.
- *BR mapping* matches desired BR and standard functionality. It helps identify and resolve gaps between the application and the ERP package.

We selected ZT's pilot ERP projects as an ideal case study (Yin 2003) to explore off-the-shelf RE methods for a number of reasons. First, ERP development is a critical case in testing off-the-shelf software adoption. ERP projects demand considerable resources and commitment, and are key to enduring business success. Second, AIM is representative in ERP packages. This indicates that the lessons learned from our case are informative about the experiences of the typical situation. Third, ZT's ERP development represents both a revelatory and a longitudinal case, in that pilot ERP studies were seldom investigated but three sequentially linked ZT

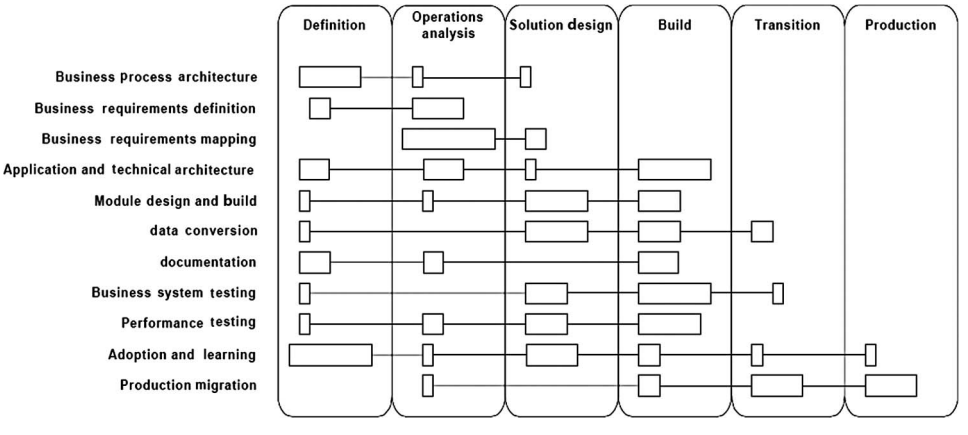

Figure 2. AIM process overview (Oracle AIM 1999).

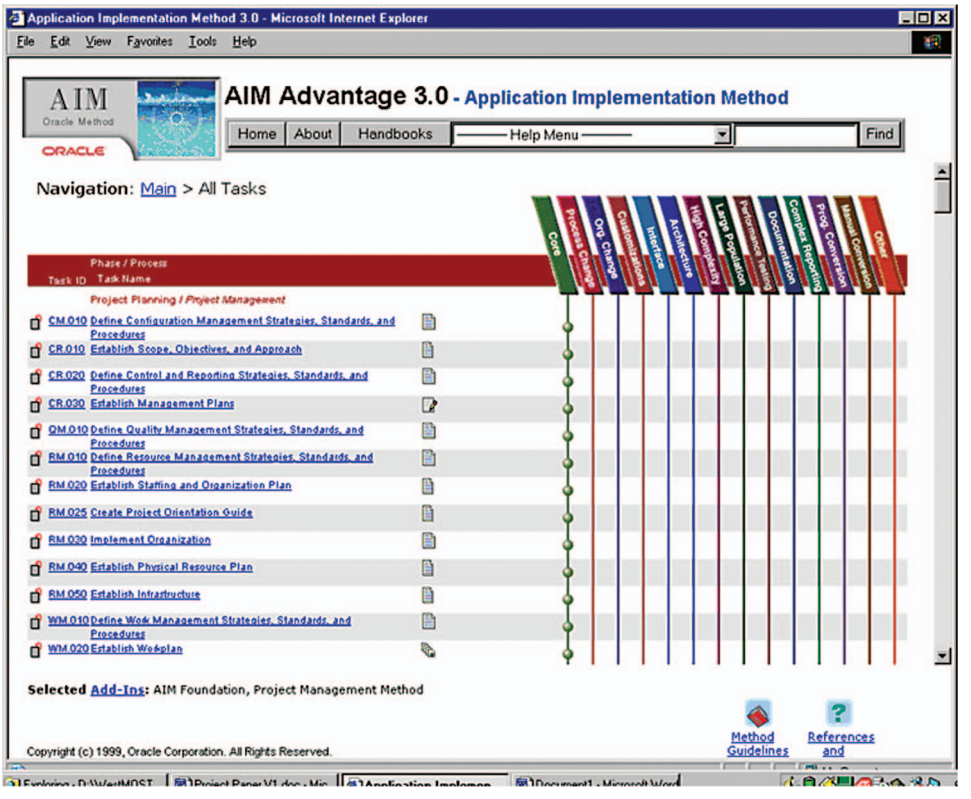

Figure 3. AIM document and process querying tool.

projects were included as units of analysis in our study design (see Section 4.3 for details). Finally, ZT's business division under study was highly cooperative and generous with regards to our research, so we anticipated a high degree of access to key stakeholders and projects' data.

4. Methodology

A case study is an empirical inquiry that investigates a phenomenon within its real-life context; (Yin 2003) covers all aspects of the case study method – from problem definition, design and data collection, to data analysis and composition and reporting. Case studies rely more on qualitative analysis than statistical analysis over a large number of instances to connect cause and effect. In software engineering, they are particularly suited to studies in which the researcher has little control over the key variables (Easterbrook *et al*. 2007).

We used an *exploratory case study* (Yin 2003) as the basis for our research design. Exploratory case studies are ideal for analysing what is common and/or different across cases that share some key criteria. They are appropriate for preliminary studies in which it is not yet clear which phenomena are important, or how to measure these phenomena. In our case, we were particularly interested in understanding how the ERP RE method would affect the business application development process. While the ERP RE suggests some specific benefits, we know

little about the issues arising when organisations make the standard ERP RE a live process, and we know less about how to make such a process work better. In other words, not enough is known about how exactly ERP RE is best deployed, nor how the expected benefits arise. For these reasons, it would be premature to try to measure the cost/benefit trade-off. For this study, our intention was to explore *how* a pre-defined ERP RE model affects business application development.

4.1. Research question

Our underlying research question is to examine whether *'Aligning ERP projects to the AIM RE model leads to integral RE practices and economic application development.'* This arises from the underpinning off-the-shelf development philosophy: if the chosen ERP RE were not fully exploited, the business application development would remain separated and costly. In order to address the research question, we consider the following practice areas.

- *AIM RE improves ZT's team collaboration experience*. Because AIM explicitly defines phases and processes, as shown in Figure 2, thus allowing team members to carry out tasks according to their specialisation and background, collaboration should be better facilitated and guided.
- *AIM RE improves ZT's project management experience*. Because business and design decisions must be explicitly recorded in AIM RE, intra-project management should be easier. Because AIM supports all of Oracle's application product families such as customer relationship management (CRM), finance and human resources, inter-project management should be enabled.
- *AIM RE improves ZT's data visibility and integrity*. Because AIM's uniform IT infrastructure and Oracle's expertise in data management, organisation-wide data sharing, data consistency and cross-business data integration should be effectively achieved.
- *AIM RE improves ZT's ability to map application-specific requirements*. Because the gaps between business application requirements and pre-defined ERP functionality can be identified and resolved in AIM RE, concerns that are specific to particular applications should be adequately addressed.
- *AIM RE improves ZT's flexibility of building customised applications*. Because application extensions are built-in notions in AIM, all kinds of ERP project should be customised with relatively small effort.

The list is by no means an exhaustive set of criteria that must be assessed during the adoption of any off-the-shelf ERP RE model, but represents the perspectives that help us to refine and operationalise the underlying research question. We believe that if AIM RE improves ZT's practice in these areas, then the claimed benefits will be demonstrated.

4.2. Study design

Our underlying research question concerns the benefits of aligning ERP projects, especially multi-ERP developments, to the AIM RE model. To investigate this, we chose three ERP projects from ZT's pilot repository: product lifecycle management

(PLM) system, customer service and support (CSS) system and CRM system. Table 1 summarises some basic characteristics of these projects, all of which were developed on top of AIM 3.0.0 (Oracle AIM 1999).

As described in Section 3, these projects were launched in one of ZT's business divisions as pilot trials of an incremental ERP adoption process. PLM was used within the designated business division to enable easy tracking of the entire lifecycle of a software product from its conception, through design and development, to deployment, upgrade and disposal. PLM integrated stakeholders, data, processes and business systems, and provided a product information backbone for the company. CSS aimed to improve customers' experience by providing an integrated tool to solve customer problems. CSS was part of a broader system, CRM, whose goal was to manage and nurture ZT's interactions with customers and sales prospects. CRM covered not only customer service and technical support but also sales-related activities such as online campaigns and offers. CRM, including CSS, had cross-divisional interfaces but was deployed within ZT's corporation boundary. No cross-organisational ERP attempt was made in these pilot projects.

It is worth pointing out that these three ERP projects were sequentially-linked in ZT's business division, as indicated by the project duration listed in Table 1. It was ZT's intention to assign similar management and development personnel to consolidate experience from these interrelated projects. All the three developments were Java-based, so the effects of technical environment were overshadowed by BPs.

In terms of size, PLM was the smallest and CRM was the largest, as indicated by the project duration and the team formation in Table 1. In all projects, one or two Oracle certified consultants participated in AIM training and phase monitoring. This was due to ZT's intention of making the AIM adoption a live process. ZT hired the consultants to further handle around 20% of the PLM implementation. The reliance on consultants was decreased in CSS and CRM. It was the project leader's responsibility to build sufficient connections to stakeholders from sales, marketing and customer services. Notably, a business analyst with good interpersonal skills and working knowledge of BP change and application functionality joined the CSS and CRM projects. This was key to facilitate and improve team collaboration. Similarly, a customer relation manager oversaw and coordinated the aspects in CRM. Time pressure was not a serious concern in all three projects because ZT initiated them as experimental trials. However, following the AIM recommendation (4 weeks for RE cycle completion) was intended.

In our study, the data were collected by means of semi-structured interview and artefact analysis (Lethbridge *et al*. 2005). The interviews involved project leaders, developers, AIM consultants and the formats ranged from structured surveys to open-ended questionnaires. See Appendix for the structured interview questions. Artefacts in our analysis included project management report, business baselines, functional requirements, preliminary conceptual architecture, meeting minutes, etc. Due to geographic constraints, most data collection was done via asynchronous communication such as e-mails and Web forms. A few teleconferences and on-site visits were held during our study. We took the actual project duration into account because some experience or answers were only obvious in hindsight.

In our evaluation, we used qualitative methods (Richards 2005), coding (relating answer sections to proper subject matters under testing) and categorising (classifying answers to be positive or negative), when analysing collected data. Qualitative

Table 1. Characteristics of ERP projects developed by AIM (release 3.0).

	PLM (product lifecycle management)	CSS (customer service and support)	CRM (customer relationship management)
Project duration	December 2000–April 2001	January 2002–August 2002	May 2004–March 2005
Project scope	Intra-division	Inter-division and intra-corporation	Inter-division and intra-corporation
Development environment	JSP, Perl-CGI, IIS	J2EE, Apache	J2EE, EJB, Apache
AIM recommended RE cycle	4 weeks	4 weeks	4 weeks
Actual RE cycle	4 weeks (including 1 week AIM training)	5 weeks (including 1 week AIM training)	5 weeks (including 1 week AIM training)
Team formation and breakdowns	7 people (1 project manager, 2 Oracle consultants, 4 developers)	10 people (1 project manager, 1 business analyst, 1 Oracle consultant, 7 developers)	13 people (1 project manager, 1 business analyst, 1 customer relation manager, 1 Oracle consultant, 9 developers)

research seeks to make sense of the way themes and meanings emerged and patterned in the data records built up from observations, interviews, surveys and other research media. It is particularly suitable in our study since our collected data were records of observation and interaction that were complex and contextualised.

5. Findings and observations

In this section, we report the results of studying ZT's pilot ERP projects that attempted to make an off-the-shelf RE model become a live process in business application development. We present facts, observations, lessons learned, and organise our discussion according to the practice areas presented in section 4.2.

5.1. AIM RE improves ZT's team collaboration experience

In the three projects we studied, the amount of project stakeholders' involvement was directly related to the results: no involvement yielded no results, while passive involvement yielded poor results. The most critical success-enabler was the strong commitment of upper management to the ERP projects, viewing them as business change initiatives rather than self-possessed software projects. As pointed out by the business analyst in the CSS project, *'AIM changed the way we service our customers because it imposed its own logic on our technology and culture'*.

ZT effectively made AIM training a mandatory activity to project management teams and staff engineers in its business division. This not only raised awareness and engagement but also helped discover the personnel who demonstrated interest in the upcoming BPR. In all the PLM, CSS and CRM projects, it turned out these early-involved trainees would likely to identify multi-component data conflicts and cross-divisional process mismatches, while passive or skeptical training participants routinely overlooked such integration requirements.

In ZT's CSS and CRM projects, large numbers of users were trained in order to deploy the systems, largely through a 'train the trainer' approach. Training lasted 1 week in each of the three pilot projects, as shown in Table 1. After CSS was completed, ZT initiated a project of developing company-specific online training materials, mainly brief readings and multiple choice quizzes. The training for the CRM project, therefore, consisted of 1-day online training (including passing a preliminary exam) and 4-day traditional consultant onsite training. While the initial feedback about online training was mixed, ZT wanted to continue experimenting with this approach to further assess its applicability and effectiveness.

ZT's experience recommended the active involvement of a certified (external) consultant into the project. However, in PLM, a collaboration mechanism was missing and sometimes the result was frustration. The project leader was reluctant to prioritise application-specific requirements for fear that standard ERP package would drop the lower-ranked functionality. Meanwhile, the consultant was reluctant to measure capacities or set up project milestones due to inaccurate information. To improve the teamwork experience in CSS and CRM, ZT assigned a business analyst with good interpersonal skills, and made the external consultant assume a collaborative role from the beginning of the projects. Our study stressed that collaboration issues could be dealt with effectively only by joining consultant's knowledge of the ERP package with the business analyst's knowledge of the organisation.

5.2. AIM RE improves ZT's project management experience

The most important feature by far and indeed the *guts* of AIM are the documentation templates it provides (Oracle AIM 1999). AIM's project management method, together with the documentation templates, makes AIM a powerful tool for assisting team members in running projects smoothly. Often, just finding the project documentation is difficult and once this process is completed, people find that the documentation is inadequate. This makes it very difficult to support such implementations and any new users of such software have to rely on learning from previous users whose availability is often in question.

ZT's pilot trials leveraged AIM's templates and tools to support project management. Such templates and tools could only succeed if the project teams used them properly and made them a routine part of the process. However, there are more than 150 templates defined in AIM 3.0.0, which come in a variety of formats such as MS Word documents, MS Excel Spreadsheets, and MS Project Files (Oracle AIM 1999). Making use of every template in ZT's projects was clearly an overkill. One strategic solution that ZT developed was to adopt generic project management documents and ignore Oracle configuration documents. ZT found that AIM RE templates (BP, BD, and BR, cf. section 3) relating to BR scenarios, high-level gap analysis and requirements and data mappings were particularly useful. Some other templates, according to ZT's CSS project manager, *'would require over-customisation to be usable'*. The tendency to over-customisation was expensive and consumed much internal resource, so ZT chose to *'un-customise customisations'* by not following some AIM templates.

Meanwhile, because AIM RE tools and templates were new to ZT, introducing them in parallel with the RE process was considered practical. The strategies included making project teams aware of the dependencies between AIM and the documentation templates, choosing a small subset of available templates that suited the application, gaining consensus on what tools and templates to use, and designing training scenarios of how to use the tools and templates to support RE activities.

AIM uses Oracle's Project Management Method (PJM) to provide a framework in which projects can be planned, estimated, controlled and tracked in a consistent manner. This consistency is required in ZT's business environment, where projects implemented packages and developed application extensions in order to satisfy the BR. Each task within PJM is assigned to a PJM life-cycle category. Figure 4 shows PJM and its relationship with AIM, where PJM life-cycle categories (project planning, phase planning, phase control, phase completion, and project completion) are integrated into the project plan at appropriate project and phase levels. ZT's experience of using PJM to manage CSS and CRM was positive in that the defined activities and life-cycle categories provided a repeatable process and streamlined the work breakdown structure.

5.3. AIM RE improves ZT's data visibility and integrity

Despite Oracle's expertise in data management, AIM 3.0.0 did not require conceptual data models as mandatory project deliverables, partly because having data models was implicitly assumed in AIM. As a result, all ZT's ERP projects in our study slowed down due to insufficient data specification and modelling in the RE

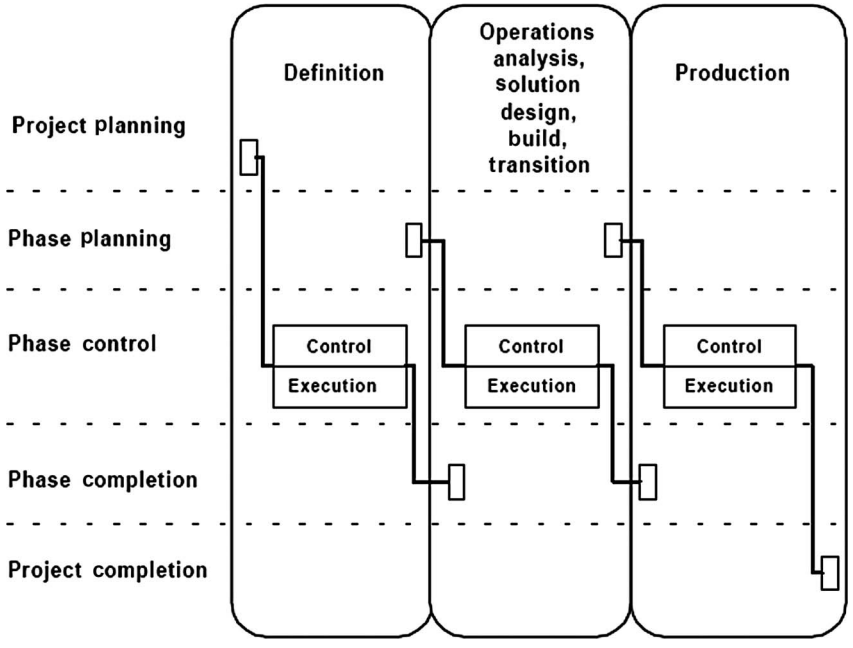

Figure 4. Using PJM to manage an AIM project.

stage. The lesson learned was to make the hidden assumption explicit, and an ERP RE methodology needs to address such important data management issues as data specification, data migration, data cleaning and data integrity.

To improve data visibility, ZT extended the AIM framework as illustrated in the process flow diagram in Figure 5. The logical extensions are highlighted in grey. The newly added database components acted as central repositories for project artefacts. The extensions illustrated a direct integration between requirements, feature-set mapping and configuration stages of RE. Moreover, a greater amount of configuration and requirements reuse was supported by the extended framework.

Following the improved process shown in Figure 5 to study the data requirements of customer-related functions, ZT was able to standardise about 80% of the data used in its CRM project. From CSS to CRM, ZT went from using 12 data schemes for customer support to using just one, and it standardised all data about technical and service reporting. While some country-specific customer data, such as accounting and tax information, have not been fully standardised, ZT achieved a remarkable degree of data visibility and integrity.

While logically extending AIM by making explicit the undocumented built-in assumptions, ZT recognised the importance of involving a qualified data architect in the ERP projects. The responsibilities of the data architect included defining a data dictionary and an interface specification. Although not attempted in ZT's pilot projects, extending AIM and involving data architects could effectively coordinate cross-organisational ERP systems in that data semantics, utility, and communication were better modelled and aligned (Daneva and Wieringa 2006).

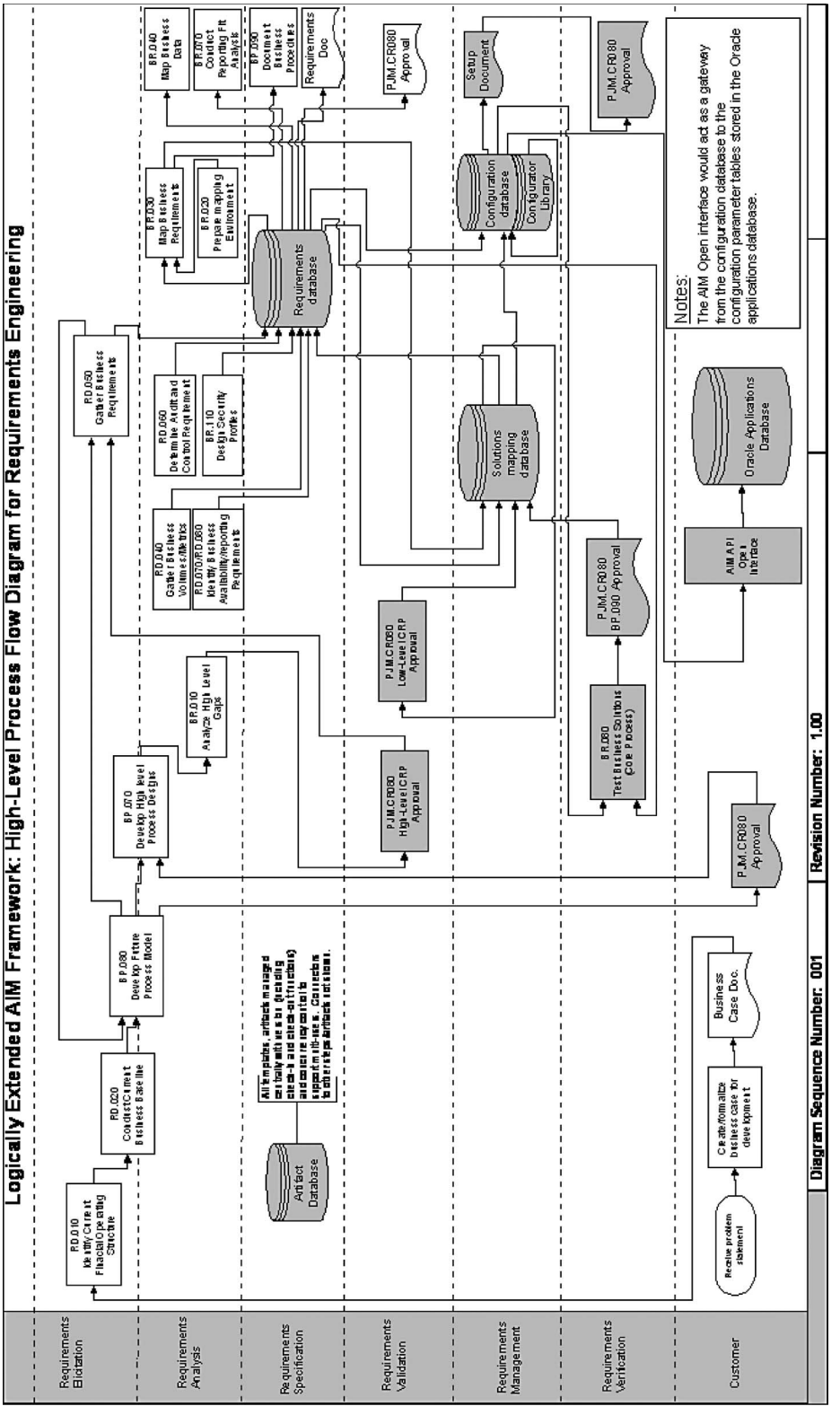

Figure 5. Logical extension to AIM emphasising data visibility.

5.4. AIM RE improves ZT's ability to map application-specific requirements

Studies suggested that, on average, for any ERP project, few process or data components were reusable at the 80–100% level (Holland and Light 1999, Davenport 2000, Daneva 2002). The conceptual picture shown in Figure 1 captured ZT's experience that business applications must customise processes, data flows, and components to complete the fit. But doing so without upfront analysis could lead to an overwhelmingly expensive undertaking. Poorly mixing fully reused, customised and newly created requirements resulted in poor transaction code with unexpected and unlocalised side effects (Daneva 2004).

Identifying and resolving gaps between BR and ERP functionality were defined in AIM's BR process, which ZT found useful. The resolving methods included documenting workarounds, creating alternative possibilities, using application extensions and changing the underlying BP (Oracle AIM 1999). The ERP functionality reuse rate was estimated to be slightly over 50% in each of ZT's pilot projects.

Although AIM provided several methods, application specifics and management experience played a key role in resolving requirements mismatches. In the CRM project, ZT had to throw away a few reuse options because the target area had not been standardised in AIM. Besides, ZT realised that they had to upgrade customisations whenever the vendor upgraded base functionality. Marketing hype aside, when project leaders better understood ERP reuse and customisation risks, they would be less inclined to unnecessary adaptation and more likely to reprioritise the application-specific requirements.

5.5. AIM RE improves ZT's flexibility of building customised applications

In ZT's trials, the biggest gain was probably developing three different kinds of ERP applications, PLM, CSS and CRM, using AIM. On one hand, standard ERP functionality and process were bound early in the package to guide blackbox reuse. On the other hand, AIM allowed late binding (customisation) through just-in-time requirements determination (Potts 1995).

The decision of ZT's multi-project development concerned the selection of ERP modules such as financials, controlling, asset management, production planning, human resources, and so forth. ZT also needed to decide the module composition process and how to reshape the BP. Not all companies wish to make massive changes to their BPs (Parr and Shanks 2000). ZT realised that a certain degree of BPR was inevitable, particularly when legacy systems were involved. However, and interestingly, experienced experts were adamant that ERP implementation should be accompanied by as little BPR as possible (Parr and Shanks 2000). This confirmed ZT's incremental ERP adoption strategy and AIM's flexibility.

Although the literature suggested that a complete set of requirements was not needed for ERP RE (Rolland and Prakash 2001), ZT discovered that the tension towards completeness was important for trade-off analysis and preventing requirements leakage (Daneva 2004). In PLM, creeping seriously affected the quality of the resulting requirements document: signed-off specification was neither consistent nor complete. These risks were invisible until project's later stages and resulted in expensive rework. In CSS and CRM, ZT stated some standard AIM criteria for completing the business blueprint and had stakeholders accept them.

The improvement of the CRM project also included the use of AIM reference models to monitor scope and check traceability through requirements artefacts.

Figure 6 summarises the survey results in a 5-point Likert scale: 1 represents 'strongly disagree' and 5 represents 'strongly agree.'[2] We aggregated the responses across the three pilot projects, and used boxplots (Tukey 1977) to depict the answers to specific questions raised in our questionnaire (cf. Appendix, Part 2). The survey aimed to use an ordinal scale to depict ZT's project members' opinions about our research questions. We invited 17 people to participate in the survey. We collected 12 completed surveys, among which five responded to more than one pilot ERP project. The respondents covered the ERP projects' major roles, ranging from project manager and business analyst to Oracle consultant and ERP project developer. We accounted for both quantitative and qualitative methods when analysing the data (ratings, free-form answers, interview transcripts, etc.). We concluded that, in ZT's case, AIM RE improved team collaboration and project management, but data visibility and integrity should be greatly improved. ZT's AIM training was considered effective, and the ERP pilot projects were completed on time and within budget. It is not clear whether AIM RE improved ZT's ability to map application-specific requirements or its flexibility of building customised applications.

We considered having three sequentially linked units of analysis was important because ZT was able to learn from its own mistakes. According to ZT's business analysts, the ERP system's assumptions, or so-called best practices, both coincided and conflicted with ZT's business needs. In this context, a misfit occurs when the company's BR do not match the capabilities of the ERP package. Misfit can be caused by many factors, e.g. company-, industry- or country-specific requirements (Soh *et al.* 2000). Figure 7 summarises ZT's misfit resolution strategies. ZT's experience showed that which strategy to adopt directly related to the management's risk tolerance level.

- *BPR*. This requires the greater degree of organisational change. It was in ZT's interest to rework some of its processes, such as customer care and retention, to fit the ERP system's requirements. BPR results in 'tight coupling' so that the benefits of integration and control associated with ERP implementation can be achieved (Ciborra 2000, Srivardhana and Pawlowski 2007).

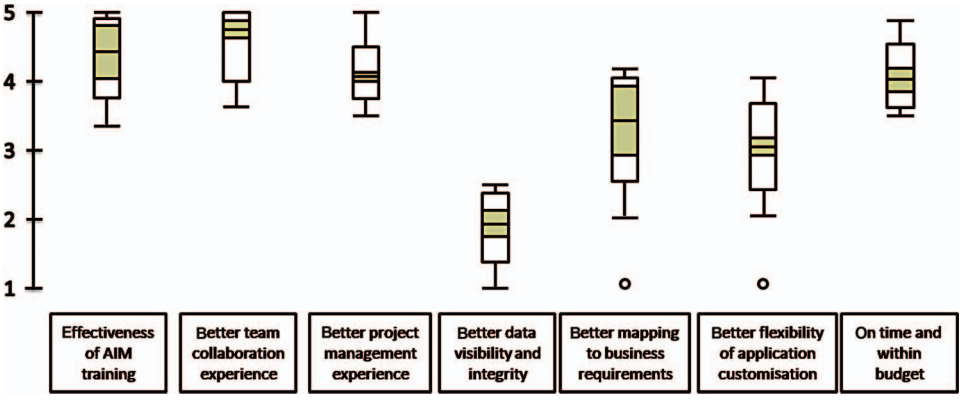

Figure 6. Ratings of evaluation questions (cf. Appendix, Part 2).

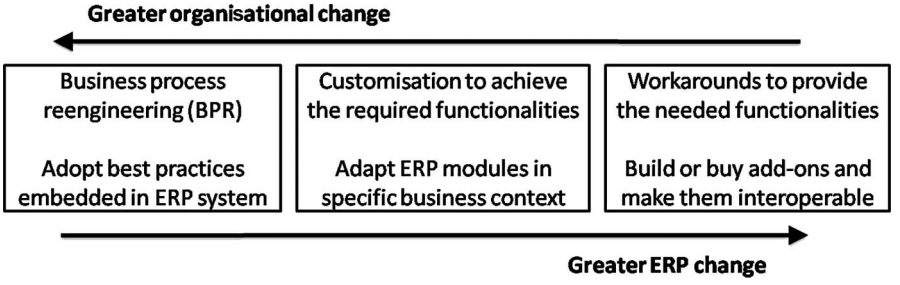

Figure 7. Misfit resolution strategies.

- *Customisation*. ERP systems are designed to be customisable, e.g. SAP R/3 offers so many options in 10,000 tables that implementation is often extremely complex (Scott and Kaindl 2000). ZT, for example, spent much time in customising the reporting module's format and content in its ERP projects.
- *Workarounds*. This requires the greater degree of ERP package change. It is argued that most customers inevitably find at least 20% of their needed functionality is missing from the standard ERP package (Scott and Kaindl 2000). Various workarounds, such as developing in-house proprietary modules or purchasing tailor-made functionality, often provide solutions. The workarounds are also known as 'loose coupling', which may compromise the benefits of process and data integration that should result from ERP implementation (Soh *et al*. 2003). Similar to the experience reported in (Liang *et al*. 2004), ZT found the lack of appropriate cost control modules in Oracle's ERP package. As a result, ZT created a separate module but faced the challenge of maintaining the module's interoperability with standard ERP modules.

In summary, our study not only confirmed some claimed benefits of adopting standard off-the-shelf RE methods but also identified implicit assumptions of AIM and pointed out deviants from ERP RE practices in the literature. In terms of our underlying research question, we felt that our exploratory case study *supported* the claim. Based on ZT's experience gained and lessons learned, we suggest that, when possible, stick with the ERP vendor's architecture approach to better manage complexity and support requirements reuse. The application team could use the ERP framework to develop the first blueprint version, which reflects basic requirements and is unlikely to be met with disagreement. More sophisticated requirements could be gradually incorporated and controversial issues could be systematically addressed in the incremental adoption process. Finally, it is crucial to keep the requirements baseline evolving along with the stakeholders' changing desires and needs.

6. Threats to validity

Several factors can affect the validity of our exploratory case study: construct validity, internal validity, external validity and reliability (Yin 2003).

Construct validity concerns establishing correct operational measures for the concepts being studied (Yin 2003). The main constructs in our case study are the concepts of 'AIM RE' and 'economic application development'. Regarding the first,

we chose AIM's BP, BD, BR processes, together with reference models and documentation templates defined in these processes, to represent the AIM RE model. By default, each ERP vendor offers an architecture concept that defines their package's underlying building principles, structures the business reality (in data and process views, at least), and provides conceptual modelling language for each view's components (Davenport 2000). Therefore, we do not feel choosing processes and models to represent this concept is a serious limitation.

As for the second construct, our interpretation of economic application development was intentionally broad. We considered ZT's five practice areas to help translate this construct into observable phenomena. We also asked ZT's employees to respond to the 'on time' and 'within budget' aspects of the ERP project. The most direct measure of 'economic' would come from project's budget, other allocated resources, and return on investment figures. However, per organisation's policy, these cannot be disclosed. Our best measure, therefore, came from the subjective opinion of study participants.

Internal validity concerns establishing a causal relationship, whereby certain conditions are shown to lead to other conditions (Yin 2003). Our major source of data for this study was the interviews we conducted with project teams from the partner company. In most cases, surveys and questionnaires were carried out asynchronously. Due to geographic constraints, we were unable to observe the actual development on site. Participants in our current study may have omitted important facts when answering questions, or we may have misinterpreted them. Observational field studies (Lethbridge *et al.* 2005) will help uncover these problems, and we plan to carry them out in further studies. Another threat to internal validity is the personal bias of the respondents due to the fact some survey questions are stated in a relative way. We reformulate some questions (Appendix, Part 2) in an absolute way with inverse Likert-scale polarity in order to guide future investigation.

External validity concerns establishing the domain to which a study's findings can be generalised (Yin 2003). The results of this study might not generalise beyond the AIM RE model. AIM was chosen by the partner company, and we had little control over it. Other ERP RE methods, such as Accelerated SAP (ASAP) or Baan's Dynamic Enterprise Modelling, might direct business application development in other dimensions and avoid some of the problems associated with AIM. Further case studies are needed to examine the effects of other methods. We also note that the pilot projects that incrementally adopted ERP package were chosen in our study. It is not clear whether the 'big bang' ERP adoption projects would exhibit similar characteristics. This warrants further analysis.

Reliability concerns demonstrating that the operations of a study can be repeated with the same results (Yin 2003). We expect that replications of our study should offer results similar to ours. Of course, the characteristics of selected off-the-shelf software and business application projects under study will differ from our reports, but the underlying trends should remain unchanged.

7. Related work

Packaged ERP systems have been universally accepted by the industry as practical solutions to achieve integrated enterprise information systems. Recent literature reviews of ERP, such as Moon (2007), pointed out that case studies were valuable contributions to knowledge by investigating a company's ERP implementation

experience with real data and observations. In order for the results from our exploratory case study to be generalisable, we examined several practice areas (including RD, communication, documentation, prioritisation, and misfit resolution), and addressed threats to validity.

A key premise of ERP systems is the underlying, sometimes unstated, but often implicitly promoted notion that the reference models in ERP systems embody best business practices (Kumar and van Hillegersberg 2000). Because the embedded business models typically reflect a bias towards Western practices, cultural and country differences are uncovered when ERP systems are adopted in Asian (Soh *et al*. 2000), and in China in particular (Liang *et al*. 2004, Martinsons 2004). Specifically, the grounded categorisation of misfit types suggests the specific data, functional (access, control, operational), and output issues to be focused on by Western ERP vendors (Soh *et al*. 2000). Although some local Chinese ERP vendors, such as UFSoft and HJSoft, had price advantage (Liang *et al*. 2004), ZT partnered with Oracle not only for Oracle's technological strengths and high-quality corporation image but also for ZT's internationalisation strategy.

One of ZT's experiences gained was on consulting and training. As pointed out in Liang *et al*. (2004), many foreign ERP vendors often fail to localise their strategies to capitalise on their financial strengths and technological advances. In its haste to gain market share, SSA, for instance, expanded too quickly to keep up with adequate consulting service and customer support due to the shortage of qualified technical personnel. Consequently, many ERP implementation failures resulted (Liang *et al*. 2004, Xu 2008). ZT was fully aware of the situation and tried to hire the *'most qualified consultants possible'*. Besides, ZT strategically reduced the reliance on consultants as projects progressed, and started to develop company-specific training materials to complement traditional training classes. Such company-tailored training helped ZT's stakeholders to discover a more complete, consistent and correct set of requirements.

ZT not only regarded its incremental ERP adoption strategy as feasible but also recognised the benefits gained through small-scale changes might not be fully realised. At the other end of the spectrum, 'big bang' adoption may be suitable for organisations that consider the risk is worth taking. Quantum Corporation, for example, undertook a 'big bang' implementation of its new ERP software (Markus *et al*. 2000). The company shut down its operations worldwide for 8 days while transitioning over to its new systems and processes. This highly risky maneuver made sense in light of the company's business strategy and organisational model, which required the worldwide capability to promise product availability to customers. Another organisational strategy ZT used throughout the pilot ERP projects was to assume centralised, not distributed, control. This allowed ZT to closely monitor requirements evolution and ensure sufficient resource (re-)allocation.

RE helps to identify and guide fulfilment of stakeholder goals and BR. It also helps to address budget and effort issues (Choi *et al*. 2007, Andersson and Linderoth 2008, Li *et al*. 2008) and promotes user-centred design (Vipola 2008). Three groups of stakeholders play important roles in ERP RE. Users own the business and organisational requirements, IT engineers know about the existing IT infrastructure and ERP vendors define best practices into reusable software packages. The viewpoints of different stakeholders are often competing and conflicting. A dilemma in ERP RE is to understand whether the organisation's requirements or ERP packages' imperatives or both (Kumar and van Hillegersberg 2000). One solution is

enterprise modelling that builds a 'conceptual value chain' to promote interoperability of different organisational units (Scheer and Habermann 2000, Johansson and Newman 2010). Another promising work is to explicitly model stakeholder *goals* for BP configuration (Lapouchnian *et al*. 2007, Graml *et al*. 2008). By capturing alternative process configurations and enabling tailored process deployments, applications built on top of ERP packages are able to accommodate changing business priorities and varying client preferences.

Alignment between ERP systems and BR has attracted many research efforts. A case study based on five companies revealed the following effective alignment strategies: functional expertise, knowledge integration, liaison mechanisms, project governance and the scope and integration of enterprise-wide processes (Sumner 2009). While our study echoed the importance of knowledge integration and project governance, other strategies were not reported by ZT's participants. Compared to general ERP solutions, industry-oriented ERP (IERP) system is designed for enterprises belonging to a specific industry sector and often supports specific business needs which are not covered by existing software packages (Izza 2009). For instance, ZT might choose an IERP designed for PC manufacturing for its PLM, CSS and CRM projects. Such an IERP incarnates more industry-specific requirements, eliminates redundant modules and functions and maintains a moderate scale (Wu *et al*. 2009). IERP also requires the vendors to benchmark best practices, including RE practices, within a specific industry sector. Alternatively, ZT might investigate how resilience, a crosscutting concern, could affect their multiple ERP projects (Wang *et al*. 2010, Zhang and Lin 2010).

As more companies have paid attention to the RE activities for embracing ERP systems, the questions on the value of ERP RE seem to be investigated more often and rigorously (e.g. Daneva 2002, 2004, Daneva and Wieringa 2006, Niu and Easterbrook 2008). This is an indication that the practices and understanding of ERP RE have matured enough to warrant some serious reflections on its fundamental questions. The exploratory case study reported in this article tested a key tenet of ERP RE adoption and shared our experience. More importantly, our study helps to point out some important issues and open up a few new research avenues.

8. Conclusion

This case study was set up to investigate the effects of ERP RE in business application development. We found that aligning ERP projects to the standard RE model led to integral practices (e.g. improved team collaboration and project management) and economic development (e.g. building three customised applications and completing the projects on time and within budget). We also uncovered implicit assumptions of the ERP RE model (e.g. having additional artefacts and processes to better support data visibility and integrity), identified deviants from common practices in the literature (e.g. eliciting towards a complete set of requirements, as opposed to using an incomplete set of requirements (Rolland and Prakash 2001), turned out to be more desirable for ZT) and reported threats to validity.

Our future work includes quantifying the cost-benefit of ERP RE, which may require the use of a process framework and multi-dimensional criteria to evaluate ERP projects (Teltumbde 2000). Furthermore, the pilot projects in our case study

aimed for 'accelerated' ERP implementation. It is argued that accelerated methods offer a minimal fit due to the short-time vision and the selection of limited functionality (van Everdingen *et al*. 2000). It would be interesting to examine flexible ERP implementations that allow the organisation to easily handle a wide variety of business procedures. Moreover, we are interested in identifying the common and varying elements in ERP RE models, e.g. Oracle's AIM, ASAP and Baan's Dynamic Enterprise Modelling. Lastly, we plan to investigate how software componentisation could overcome the monolithic character of current ERP systems and increase the adaptability to BR (Klaus *et al*. 2000).

Understanding and optimising the mechanics behind a standard RE process on a practice-by-practice basis is the essence of ERP RE adoption. Our experience highlights issues that are common when running a standard process in a relatively immature organisation that operates in a dynamic business environment. Our study can direct researchers towards rigorous empirical evaluations of ERP RE model adoption, and can help practitioners collect experience from the lessons learned. The community that addresses similar issues, on the whole, is likely to generate more effective and mature RE processes – with more predictable results and better visibility – when exploiting a generic off-the-shelf RE method.

Acknowledgements

We are grateful to all the management and staff at the partner company for their invaluable collaboration. We thank Robert Brassard, Jim Hoover and Steve Easterbrook for careful comments on earlier drafts of this article. We also thank the anonymous reviewers for their constructive suggestions.

Notes

1. More than 150 templates are defined in AIM 3.0.0.
2. We realise that when using Likert-scale questions it is best to have some with inverse polarity, i.e. some questions where respondent's agreeing means that we are dissatisfied. We also realise that the current questions are stated in a relative way. This might cause biases because each respondent has to compare AIM to their prior experience and each person may have quite different prior experience. The use of absolute questions may yield data that would have less personal bias. For the above reasons, we reformulate some questions (Appendix 1, Part 2) to guide future investigation.

References

Andersson, A. and Linderoth, H.C.J., 2008. Learn not to learn – a way of keeping budgets and deadlines in ERP projects? *Enterprise Information Systems*, 2 (1), 77–95.

Basili, V.R. and Boehm, B., 2001. COTS-based systems top 10 list. *IEEE Software*, 34 (5), 91–95.

Choi, J., Ashokkumar, S., and Sircar, S., 2007. An approach to estimating work effort for enterprise systems software projects. *Enterprise Information Systems*, 1 (1), 68–87.

Ciborra, C., 2000. *From control to drift: the dynamics of corporate information infrastructures*. Oxford: Oxford University Press.

Daneva, M., 2002. Using maturity assessments to understand the ERP requirements engineering process. *In*: *Proceedings of International Requirements Engineering Conference (RE'02)*. Washington, DC: IEEE Computer Society, 255–262.

Daneva, M., 2004. ERP requirements engineering practice: lessons learned. *IEEE Software*, 21 (2), 26–33.

Daneva, M. and Wieringa, R.J., A requirements engineering framework for cross-organizational ERP systems. 2006. *Requirements Engineering*, 11 (3), 194–204.

Davenport, T.H., 1998. Putting the enterprise into the enterprise system. *Harvard Business Review*, 76 (4), 121–131.

Davenport, T.H., 2000. *Mission critical: realizing the promise of enterprise systems*. Boston, MA: HBS Press.

Easterbrook, S., *et al*., 2007. Selecting empirical methods for software engineering research. *In*: Shull, F., Singer, J., and Sjøberg, D.I.K., eds. *Guide to advanced empirical software engineering*. London: Springer, 285–311.

Feblowitz, M., *et al*., 1996. ACME/PRIME: requirements acquisition for process-driven systems. *In*: *Proceedings of International Workshop on Software Specification and Design (IWSSD'96)*, Washington, DC: IEEE Computer Society, 36–45.

Finkelstein, A., Spanoudakis, G., and Ryan, M., 1996. Software package requirements and procurement. *In*: *Proceedings of International Workshop on Software Specification and Design (IWSSD'96)*. Washington, DC: IEEE Computer Society, 141–145.

Frakes, W.B. and Kang, K., 2005. Software reuse research: status and future. *IEEE Transactions on Software Engineering*, 31 (7), 529–536.

Graml, T., Bracht, R., and Spies, M., 2008. Patterns of business rules to enable agile business processes. *Enterprise Information Systems*, 2 (4), 385–402.

Holland, C. and Light, B., 1999. Success factors model for ERP implementation. *IEEE Software*, 16 (3), 30–36.

Izza, S., 2009. Integration of industrial information systems: from syntactic to semantic integration approaches. *Enterprise Information Systems*, 3 (1), 1–57.

Johansson, B. and Newman, M., 2010. Competitive advantage in the ERP system's value-chain and its influence on future development. *Enterprise Information Systems*, 4 (1), 79–93.

Klaus, H., Rosemann, M., and Gable, G.G., 2000. What is ERP? *Information Systems Frontiers*, 2 (2), 141–162.

Kumar, K. and van Hillegersberg, J., 2000. ERP experiences and evolution. *Communications of the ACM*, 43 (4), 22–26.

Lapouchnian, A., Yu, Y., and Mylopoulos, J., 2007. Requirements-driven design and configuration management of business processes. *In*: *Proceedings of International Conference on Business Process Management (BPM'07)*. Berlin, Heidelberg: Springer-Verlag, 246–261.

Lauesen, S. and Mathiassen, M., 1999. Use cases in a COTS tender. *In*: *Proceedings of International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'09)*. Berlin, Heidelberg: Springer-Verlag, 115–130.

Lethbridge, T.C., Sim, S.E., and Singer, J., 2005. Studying software engineers: data collection techniques for software field studies. *Empirical Software Engineering*, 10 (3), 311–341.

Li, L., *et al*., 2008. Assessing the effects of manufacturing infrastructure preparation prior to enterprise information systems implementation. *International Journal of Production Research*, 46 (6), 1645–1665.

Liang, H., *et al*., 2004. Why Western vendors don't dominate China's ERP market. *Communications of the ACM*, 47 (7), 69–72.

Maiden, N. and Ncube, C., 1998. Acquiring COTS software selection requirements. *IEEE Software*, 15 (2), 46–56.

Markus, M.L., Tanis, C., and van Fenema, P.C., 2000. Multisite ERP implementations. *Communications of the ACM*, 43 (4), 42–46.

Martinsons, M.G., 2004. ERP in China: one package, two profiles. *Communications of the ACM*, 47 (7), 65–68.

Moon, Y.B., 2007. Enterprise resource planning (ERP): a review of the literature. *International Journal of Management and Enterprise Development*, 4 (3), 235–264.

Ncube, C. and Maiden, N., 1999. Guiding parallel requirements acquisition and COTS software selection. *In*: *Proceedings of International Symposium on Requirements Engineering (RE'99)*. Washington, DC: IEEE Computer Society, 133–140.

Niu, N. and Easterbrook, S., 2008. Exploiting COTS-based RE methods: an experience report. *In*: *Proceedings of International Conference on Software Reuse (ICSR'08)*. Berlin, Heidelberg: Springer-Verlag, 212–216.

Nuseibeh, B. and Easterbrook, S., 2000. Requirements engineering: a roadmap. *In*: *Proceedings of the Conference on the Future of Software Engineering*, New York, NY: ACM, 35–46.

Oracle, AIM, 1999. *Oracle application implementation method (release 3.0.0)*. CA, USA: Oracle Corporation.
Parr, A.N. and Shanks, G., 2000. A taxonomy of ERP implementation approaches. *In*: *Proceedings of Hawaii International Conference on System Sciences (HICSS'00)*. Washington, DC: IEEE Computer Society, 7018.
Peterson, W.J., Gelman, L., and Cooke, D.P., 2001. *ERP trends*. The Conference Board Report, R-1292-01-RR. New York, NY: The Conference Board.
Potts, C., 1995. Invented requirements and imagined customers: requirements engineering for off-the-shelf software. *In*: *Proceedings of International Symposium on Requirements Engineering (RE'95)*. Washington, DC: IEEE Computer Society, 128–130.
Rajagopal, P., 2002. An innovation-diffusion view of implementation of enterprise resource planning systems and development of a research model. *Information and Management*, 40 (2), 87–114.
Richards, L., 2005. *Handling qualitative data: a practical guide*. London: Sage Publications.
Rolland, C. and Prakash, N., 2001. Matching ERP system functionality to customer requirements. *In*: *Proceedings of International Symposium on Requirements Engineering (RE'01)*. Washington, DC: IEEE Computer Society, 66–75.
Scheer, A.-W . and Habermann, F., 2000. Making ERP a success. *Communications of the ACM*, 47 (4), 57–61.
Scott, J.E. and Kaindl, L., 2000. Enhancing functionality in an enterprise software package. *Information and Management*, 37 (3), 111–122.
Soh, C., Kien, S.S., and Tay-Yap, J., 2000. Cultural fits and misfits: is ERP a universal solution? *Communications of the ACM*, 43 (4), 47–51.
Soh, C., *et al*., 2003. Misalignments in ERP implementation: a dialectic perspective. *International Journal of Human-Computer Interaction*, 16 (1), 81–100.
Srivardhana, T. and Pawlowski, S.D., 2007. ERP systems as an enabler of sustained business process innovation: a knowledge-based view. *Journal of Strategic Information Systems*, 16 (1), 51–69.
Sumner, M.R., 2009. How alignment strategies influence ERP project success. *Enterprise Information Systems*, 3 (4), 425–448.
Teltumbde, A., 2000. A framework for evaluating ERP projects. *International Journal of Production Research*, 38 (17), 4507–4520.
Tukey, J., 1977. *Exploratory data analysis*. Reading, MA: Addison-Wesley.
van Everdingen, Y., van Hillegersberg, J., and Waarts, E., 2000. ERP adoption by European midsize companies. *Communications of the ACM*, 43 (4), 27–31.
Vipola, I.H., 2008. A method for improving ERP implementation success by the principles and process of user-centred design. *Enterprise Information Systems*, 2 (1), 47–76.
Wang, J.W., Gao, F., and Ip, W.H., 2010. Measurement of resilience and its application to enterprise information systems. *Enterprise Information Systems*, 4 (2), 215–223.
Wu, S.L., Xu, L., and He, W., 2009. Industry-oriented enterprise resource planning. *Enterprise Information Systems*, 3 (4), 409–424.
Xu, L., 2007. Enterprise information systems. *Enterprise Information Systems*, 1 (1), 1–2.
Xu, X., 2008. Effects of two national environment factors on e-commerce functionality adoption: a cross-country case study of a global bank. *Enterprise Information Systems*, 2 (3), 325–339.
Yin, R.K., 2003. *Case study research: design and methods*. 3rd ed. CA, USA: Sage Publications.
Zhang, W.J. and Lin, Y., 2010. On the principle of design of resilient systems – application to enterprise information systems. *Enterprise Information Systems*, 4 (2), 99–110.

Appendix. ERP projects survey and interview questions

Part 1: Background information

(1) What ERP project(s) have you participated in?
(2) What was your role in each project?
(3) What were the main business drivers for the ERP implementation?

Part 2: Evaluation

(For the questions listed in this part, please use the following scale to rate your answer: 1 – Strongly disagree, 2 – Disagree, 3 – Neither agree nor disagree, 4 – Agree, 5 – Strongly agree.)

(1) The effectiveness of AIM training was effective, in terms of quantity and quality.
(2) Doing AIM requirements engineering improved team collaboration experience.
(3) Doing AIM requirements engineering improved project management experience.
(4) Doing AIM requirements engineering improved data visibility and integrity.
(5) Doing AIM requirements engineering enabled better mapping to business needs.
(6) Doing AIM requirements engineering enabled more flexible application customisations.
(7) The ERP project developed by adopting AIM requirements engineering was completed on time or within budget.

Note: We reformulate the following questions in an absolute way with inverse polarity in order to guide future investigation.

(1) The team collaboration experience when adopting AIM requirements engineering should be greatly improved.
(2) The project management experience when adopting AIM requirements engineering should be greatly improved.
(3) The data visibility and integrity when adopting AIM requirements engineering should be greatly improved.
(4) The ability to map to business needs when adopting AIM requirements engineering should be greatly improved.
(5) The flexibility of building customised applications when adopting AIM requirements engineering should be greatly improved.

Part 3: Lessons learned

(1) Common mistakes checklist
(2) Symptoms
    - Lack of strategic alignment?
    - Lack of stakeholder involvement?
    - Poor planning?
    - User resistance?
    - Insufficient training?
    - Culture misfit?
(3) Recommendations for the future