ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Information and Software Technology xxx (2007) xxx–xxx

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

Investigating Knowledge Management practices in software development organisations – An Australian experience

Aybüke Aurum *, Farhad Daneshgar, James Ward

School of Information Systems, Technology and Management, University of New South Wales, Sydney, NSW 2052, Australia

Received 27 September 2006; received in revised form 21 May 2007; accepted 23 May 2007

Abstract

This study, using both quantitative and qualitative methods, investigates current practice of Knowledge Management (KM) in Software Engineering (SE) processes in two Australian companies on the basis that they both claimed to apply KM practices in their software development work. It also describes the KM activities and KM process used in SE practice, and examines the enablers of KM process for SE in terms of leadership, technology, culture, process and measurement.

One of the main findings showed that software developers believe in the usefulness of knowledge sharing; however, their ability to utilise some of the KM systems was limited. The most commonly used systems included personal networks, informal networks, groupware and third-party knowledge. There is a need to formalise knowledge sharing of practices, while also supporting informal and ad-hoc knowledge sharing. While KM was considered to be important, the tools, techniques and methodologies currently employed for software development were inadequate to address effective management of knowledge in these organisations. In both organisations, a uniform model of the KM process did not exist. Among the four KM enablers, leadership was considered to be the most significant as top-down KM strategies were seemingly being pursued by management. Technology was also considered to be an obvious mechanism for KM, despite some of their current KM systems either being unsuitable or inaccessible. In addition, the crucial role that personal networks played in accessing tacit and implicit knowledge was seen as a key reason to foster a culture that encourages participants to share their knowledge with others.

Keywords: Knowledge Management; Software Engineering; Software process models; KM activities; KM process enablers

1. Introduction

In the information age, many organisations compete for shrinking profit margins in increasingly global markets. As a result, the prospect of redesigning and integrating business processes in order to enhance operational efficiencies has become a necessity [7,8,31,34]. It is now expected that organisations be flexible enough to challenge their accepted business practices, while at the same time build upon their previous experience. The collective body of knowledge offered by employees of these organisations has emerged as a key point of differentiation, providing a foundation upon which the quality of products and services can be improved [8,25,34].

Over the last 30 years, software has become an integral component in everyday life. Whether it be switching on a television, heating food in a microware or driving a motor vehicle, a large proportion of the activities we perform are dependent upon software. However, despite the visibility of the Internet and the penetration of desktop computers into the consumer market, software engineering (SE) remains a discipline that has yet to reach maturity. During the 1990s, increased consideration was given to the process used for software development, centred upon its potential to improve the quality of software delivered. The popularity of the Capability Maturity Model (CMM), ISO9000 series of standards and the Software Process Improvement and Capability dEtermination model, SPICE, is illustrative of

* Corresponding author.
E-mail addresses: aybuke@unsw.edu.au (A. Aurum), f.daneshgar@unsw.edu.au (F. Daneshgar), james.ward@optusnet.com.au (J. Ward).

0950-5849/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.infsof.2007.05.005

the effort software engineers have made to improve the robustness of their processes (chap. 16 [6]), [43].

Software development processes have always been knowledge-intensive [6,19,42]. For one thing, the increased complexity of SE project work has led to a greater reliance upon knowledge processes to solve problems [15,20]. Through collection of process and product data, careful measurement of planned software activities, trial and error, and feedback from customers and the environment in general, software organisations gain local experience [25,47]. It is the nature of many software projects that unless the latest knowledge is available, opportunities to resolve problems can be lost. Software development teams do not work in isolation, and are required to work in a cross-functional environment. In situations like this, effective knowledge sharing among software developers becomes a critical success factor, and KM becomes an enabler of organisational learning [45,47]. When applied to the SE development processes and projects, KM encourages software engineers to reference the lessons they have learned from previous projects, as well as enshrining the importance of sharing best practices amongst peers [7,18,21]. As a result, software engineers can improve their efficiency, and reduce unnecessary re-work. More importantly, managers are able to capture the domain knowledge that software developers acquire during their work.

There is currently a gap in literature concerning the KM process for SE, particularly that which relates to an Australian context. As a result, it is not possible to make a strategic impact analysis of KM as applied in SE processes in Australia. For example, it is not clear whether KM itself is treated as a strategic enabler of SE process, or whether it acts more as a subset of the SE process that maintains an infrastructure for activating various KM activities as needs arise and are recognised during the SE processes, or it stands somewhere between these two extremes.

This study aims to reduce this gap by evaluating current state of KM as applied to Australian SE organisations using an empirical investigation. This article is an extension of Ward and Aurum [52] research. Two SE organisations were selected and two SE projects were selected from each of these organisations. A total of 12 people from these four projects were interviewed in order to capture qualitative data about various aspects of KM practices in those projects. In addition, the participants were also asked to complete a detailed questionnaire in order to gain an understanding of the current KM practices within their company. During the data collection we also focused on both codifying and personalising knowledge strategies [19,22]. Specific objectives of this study are: (a) to determine current practices for KM in SE in Australian companies, (b) to describe activities that comprise the KM process for SE in these companies, (c) to identify high-level models of the KM process for SE and (d) to establish the impact of leadership, technology, culture and measurement as enablers of the KM process for SE in Australia. We believe that this study should be interest to practitioners for those managing, conducting and supporting software projects as well as academics whose studying software product development process from KM perspective.

The remainder of the paper is organised as follows: some of the related work in this area is presented in Section 2. The research method is described in Section 3. Section 4 presents the results, which are then further discussed in Section 5. Validation issues are discussed in Section 6 followed by conclusions and future work in Section 7.

2. Related work

The aim of this section is to identify both the aspects as well as the boundaries of this study in relation to the relevant existing studies. The review of literature on KM in general (Section 2.1) and in SE organisations (Section 2.2) in particular will achieve this goal. These discussions provide grounds for understanding and justifying the KM framework on which the current study is based (Section 2.3).

2.1. Knowledge and Knowledge Management

Knowledge is a broad and abstract notion. The definition of 'knowledge' is one that has attracted a significant amount of conjecture [13]. Knowledge has been defined as an 'understanding awareness, or familiarity acquired through study, investigation, observation or experience over time' [12]. It has also been conceived as 'justified personal belief that increases an individual's capability to take effective action' [1,2]. Nonaka [38] distinguishes between tacit and explicit knowledge. Explicit knowledge is stored in textbooks, software products and documents. Tacit knowledge is stored in the minds of people in the form of memory, skills, experience, education, imagination and creativity [40]. There are objections in the KM community relating to the classification of knowledge into tacit and explicit types. The idea is that some knowledge believed to be tacit can in fact be transformed into explicit knowledge. This type of knowledge is called implicit knowledge. In other words tacit knowledge is a type of knowledge that is hard to articulate, whereas implicit knowledge can be articulated but has not yet. The value of implicit knowledge is recognised by in the upper echelons of management within organisations, as it is considered to be critical to enabling employees to fulfil their roles. There is a common agreement that both implicit and explicit knowledge are important; however, implicit knowledge is more difficult to identify and manage.

In essence, the objective of KM is to transfer implicit knowledge to explicit knowledge, as well as to transfer explicit knowledge from individuals to groups within the organisation. Hence, KM is concerned with creating, preserving and applying the knowledge that is available within organisations. The recent interest in organisational knowledge has prompted the issue of managing the knowledge possessed by an organisation for its benefit. Such interest

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

has given rise to the creation of a new perspective for the KM that views KM as a discipline for identifying and leveraging the collective knowledge in an organisation to help the organisation compete [50,51]. One major issue therefore would be the effectiveness of Knowledge Management Systems (KMS) in enhancing effectiveness and productivity of knowledge workers in creating competitive knowledge. However, a review of the current relevant literature reveals contradictory results [4]. For example, Markus [35] found that the experience of those seeking knowledge influences their perceptions of the effectiveness of KMS. In contrast, Ko and Dennis [29] argue that more experienced workers are less likely to seek knowledge stored in a KMS, and are more likely to contribute knowledge they have internalised for others to use. Among other things, this article examines the validity of these arguments in relation to the selected organisations.

Wickramasinghe [54] professes KM is comprised of both objective and subjective elements, and hence a KMS must be able to cater for both. Based on the results of eight case studies, Wickramasinghe reported that the competitive environments that all these organisations were operating in were key factors in driving the adoption of KMS (Ibid). In addition, recruitment and retention of employees were identified as major priorities. All organisations believed KM provided them with a source of competitive advantage. Technology remained a powerful enabler, given the global reach of each organisation and the necessity to develop searching facilities. All organisations believed that they could articulate definitions of KM, although results were varying in terms of developing an enterprise-wide vision for KM. In addition, Wickramasinghe noted that the KMS implemented in each of the three case study organisations were only capable of supporting objective, or explicit knowledge, rather than subjective or tacit knowledge. Wickramasinghe concludes that rather than implementing KMS, the three case study organisations implemented organisational memory systems.

In another study, Balthazard and Cooke [8] explain that the availability, source and flow of information are the product of human processes rather than of information technology. While they recognise the importance of organisational culture in supporting KM, they argue that aspects such as measurement and organisational change are still areas that present challenges for organisations. They propose that a constructive culture will positively impact both individual and organisational outcomes that promote KM success. Alternatively, a defensive culture will negatively impact both individual and organisational outcomes that promote KM success. The importance of organisational culture has also been raised in many other studies. For example, Lee and Choi [33] point out an overall lack of adequate metrics in organisations for measuring the success of KM initiatives. They define a KM Performance Indicator (KMPI) that has five components that can be used in determining an organisation's Knowledge Circulation Process (KCP). These components are knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilisation and knowledge internalisation. They claim that KCPs are influenced by organisational culture. They further assert that the KMPI can measure the quality of organisational knowledge, and that it can also have a correlation to managerial performance (Ibid).

2.2. Knowledge Management in software engineering organisations

This section is presented in two parts. The first part discusses the role that KM is claimed to be capable of playing in SE development processes. In the second part, the actual application of the KM in SE organisations is discussed with particular attention given to the Experience Factory model as the most widely used model in SE organisations. These arguments will then lead to a discussion in Section 2.3 on the proposed KM framework for this study.

2.2.1. The role of KM in software development process

Regardless of the claims about ambiguities surrounding the definition of knowledge, as well as the importance and necessity of KM, software processes continue to evolve. This is because software developers posses highly valuable knowledge relating to the product development, the software development process, project management and technology in general. As a knowledge-intensive work, the software development process involves both explicit and implicit knowledge. This knowledge is dynamic and evolves with technology, organisational culture and the changing needs of organisation's software development practices. There are cases where this knowledge is created at irregular intervals and the value of its use can only be displayed over time. The use of the Internet facilitates the storage and utilisation of activities, thus improving the quality of the software development process. Experience also plays a major role in knowledge-related activities. Software development can be improved by recognising the related knowledge content and structure as well as the required appropriate knowledge, and performing planning activities. Disterer [20] provides insight into the collection of knowledge and experience from project work. He observes that the closing stages of a project are increasingly becoming a common point for project reviews to be conducted, and subsequently for knowledge associated with the project to be extracted. The profiles of the project team members are presented as effective means of conveying knowledge about projects, as well as linking specific knowledge to particular individuals.

One view of KM in SE suggests that KM can be conceptualised as a major component of Software Process Improvement (SPI) initiatives [36]. In a collaborative research, Mathiassen and Pourkomeylian [36] included definition of KM strategies early in SPI projects in order to ensure KM strategies include both personalised and codified approaches, as well as align them with those of the organisational strategy.

The relationship between software development processes and KM has also been examined, eliciting a distinction between formalising and automating software processes using KM [28]. It is argued that KM in SE assists software developers in defining software processes, pursuing a process-oriented approach and improving and adapting existing software processes for future use. Although it is accepted that software process models may be developed for use by an organisation, however they must be understood by software developers before they can be effective in managing knowledge. Lindvall and Rus (chap. 4, [6]) declare that organisations must facilitate both formal and informal knowledge sharing between software developers. They assert that KM complements existing approaches to software process improvement, rather than seeking to replace them. A number of options for implementing and using KM systems for SE are advanced, such as expert identification, the creation of KM champions, document management and using predictive modelling to direct decision-making [6].

2.2.2. Application of KM in software engineering organisations

In recent years, researchers have developed a number of descriptive process models for KM in SE. However, there have been relatively few attempts to analyse the activities performed in the KM process in actual SE practice, and this paper partially fills this gap. Nevertheless, results from existing studies into KM process models deliver an overview of the problem domain of KM in SE.

For instance, the Experience Factory (EF) is a framework for experience management, which is one of the earliest models to address the problem area of KM in SE [11]. This framework is deeply aligned with organisational structures and business processes. Basili et al. [9,10] acknowledge that for an organisation to implement the EF approach for KM, a number of potential barriers to success must be overcome. They argue that although the EF is aimed at establishing a learning organisation, it requires a significant investment of time and effort.

Johansson et al. [26] apply an Experience Engine approach to KM in SE, as a subset of the EF. They list problems identified with the EF approach, such as its experimental nature, the organisational restructuring it induces as well as its reliance upon an experience base containing extensive written documentation. Hellström et al. [23] completed a case study at Ericsson Software Technology (EST), a software engineering division within Ericsson. They interviewed those involved with the development of a decentralised program for KM, and closely aligned to the principles of the personalisation approach outlined by Hansen et al. [22]. EST also determined that both the physical context and the situational nature of knowledge exchanges were two aspects of KM not covered effectively by the EF approach. Hence, EST implemented a broker model, creating the role of knowledge broker to connect those who possess knowledge with those who seek it. As a result EST was able to establish an effective balance between the centralised directives engendered by bureaucratic structure and the decentralised, human interaction that provides insight into knowledge needs and deficiencies perceived by workers.

Dingsøyr et al. [18] provide an insight into problems faced by small to medium organisations in addressing KM in SE. They consider postmortem reviews and experience reports as two approaches suitable for collecting software development knowledge, particularly endorsing lightweight experience reports. The researchers conclude that lightweight postmortem reviews perhaps reveal more about software development practices, while experience reports mostly reveal client relationships and interactions.

Komi-Sirvio et al. [30] investigate a needs-based approach for conducting KM in SE. They conducted a number of interviews with employees from a company engaged in software development. Their initial results reveal failure of recommended approaches such as Lessons Learned databases and Data Transfer Days. A new approach implemented by the organisation studied was grounded in the principles of the EF, yet focused on supplying knowledge for a project's immediate needs. This just-in-time approach is indicative of the constraints placed upon software developers, being unable to devote significant amounts of time to packaging experience for later use.

Despite favorable views of existing KM models and frameworks specifically developed for SE, Ruggles [44] provides a negative view towards KM in SE organisations. He investigated the current ideas, attitudes and approaches, held by executives towards KM in 431 US and European software organisations. The results illustrate a lack of confidence that organisations possess in relation to the transfer and generation of knowledge, despite the latter activity being identified as a key process activities in KM strategies. Some of the general areas for improvement in relation to the KM process include mapping sources of internal expertise, establishing networks of knowledge workers as well as the creation of new roles for managing knowledge. It is suggested that organisations should abandon technology-centric strategies, and focus upon enablers such as leadership and culture (Ibid).

2.3. The proposed KM framework

In this study, a number of existing KM frameworks from the literature was reviewed and their capabilities were assessed upon their popularity as well as their potential applicability to the software development [5,11,24,31,39,41]. These frameworks were selected to provide a structured means of evaluating the KM process, through testing hypothetical descriptions as well as exploring underlying mechanisms that are often difficult to uncover. The main criteria for selecting a framework for this study were; (i) it is used most widely (chap. 5, [6]) and (ii) it is most consistent with the latest holistic frameworks, e.g., [5,31,41].

In this study, three existing KM models were used to establish a theoretical grounding for investigating the

KM process for SE. The first model was the SECI model developed by Nonaka and Takeuchi [39], which has been acknowledged as an important contribution to understanding the creation of different types of knowledge. The second model was the 'Experience Factory' [10,11], which is synonymous with KM in SE, and has been implemented by both academics and practitioners with varying degrees of success. The third model was the American Productivity & Quality Centre and Arthur Andersen model [5,14] for KM, which has formed the basis for benchmarking efforts in over 100 organisations globally. After studying these well-known models, a list of major KM enablers, KM process activities and corresponding KM systems were identified as illustrated in Table 1.

(i) **KM enablers**: these are organisational factors that foster the development of knowledge through a typical KM process [5]. In other words, these enablers are considered as being necessary for successful application of KM in the domain of SE. These enablers are used in the Knowledge Management Assessment Tool (KMAT) which assists organisation to determine of its management practice [41]. From Table 1, *leadership* evaluates the role of leadership in managing knowledge within the organisation. *Culture* is concerned with the climate of the knowledge sharing environment, as well as the culture in the organisation that promotes the knowledge sharing. *Technology* examines the technology aspect of KM practice. *Measurement* aims to find out if there are effective measures to indicate the success or failure of KM practice in organisation.

(ii) **KM activities**: these activities are adopted from Arthur Anderson [5]. The activities presented in Table 1 can also be found in many of today's SE environments. Each of the clusters of KM activities in Table 1 represents a closely related set of activities that have mentioned by various writers and more or less refers to the same concept in the mainstream IS literature. These generic categories are: 'knowledge creation', 'knowledge transfer' and 'knowledge application' [2,3]. This taxonomy also closely resembles that of Rus and Lindvall [45] who mention the three categories: 'activities related to the purpose of outputs', activities related to the scope of inputs' and 'efforts required to process inputs'. An additional activity called the 'knowledge organisation' has also been recognised as a KM activity by the early KM researchers however mainly as a 'focus' rather than as an 'activity': "KM focus is organised access to and retrieval of content" [3]. This activity however is receiving explicit recognition in many of the recent studies and therefore it is also included in Table 1. Definitions for the seven KM activities in Table 1 are provided in Appendix A.

(iii) **KM systems:** in an effort to match between KM activities with KM systems that support these activities, Laudon and Laudon [31] developed a framework that shows mapping between the above two. Based on this framework, the KM systems are presented in column 3 of Table 1. Definitions for the KM systems in Table 1 are provided in Appendix A.

Table 1
An Integrated Knowledge Management framework (adopted from [5,31,41])

Knowledge Management enablers	Knowledge Management activities	Knowledge Management systems
Technology, Measurement, Culture, Relationship	K Identification K Acquisition K Creation	Communities of practice Personal networks Organisational practices and routines
	K Organisation	Document management systems Expert system Organisational routines
	K Transfer	Training Informal networks Groupware
	K Application	New IT-based business processes (BP)
	K Adoption	New products and services New markets

Table 1 allows analysis of various aspects of both organisational learning and effectiveness of knowledge workers that use various technologies and under different cultural environments and leadership styles. As a result, KM can be defined as an integrated process incorporating a set of KM activities in order to create, store, transfer and apply knowledge on a knowledge business value chain using appropriate technologies and cultural environments. One other reason for adopting an integrated framework for this study is that while many software engineers are conscious of knowledge that they have acquired from personal experience, they often find it difficult to express what they actually know. The key challenge for many of software development organisations, therefore is not only to evaluate knowledge from an epistemological perspective, but more importantly in terms of its ability to address contextual and business-related problems such as managerial and organisational issues; and the selected framework satisfies these conditions. KM in this sense becomes an enabler of organisational learning that increases the ability of the organisation to learn from its environment and to incorporate knowledge into its business processes. This study uses the framework presented in Table 1 to investigate KM practice in SE companies that are investigated.

3. Research design

Software development requires knowledge that embeds in software product and knowledge that describe develop-

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

ment process as well as organisational processes. The basic principle in SE is that the overall quality of software can be improved when knowledge is made available and used proficiently. Furthermore, the need for further development of SE practices within organisations adds to the demand for systematic knowledge and skill management at all stages of the software lifecycle. Thus, developing effective ways of managing SE knowledge is of interest to software developers.

The primary objective of this study is to provide a more complete description of the KM process as applied by a sample of Australian organisations engaged in software development by adopting an empirical research method. Work in this area is in its infancy, and early studies have demonstrated conflicting opinions on the effectiveness of existing KMSs for SE organisations. As far as the authors are aware no investigation has been conducted by Australian SE industry or other Australian institutions for evaluating aspects of KM practice in Australian SE organisations and projects. The following research questions provided a focus for investigation:

- Research question 1 (RQ1): what is the current state of practices for KM in SE in a sample of Australian companies?
- Research question 2 (RQ2): what KM activities, if any, comprise the KM process for SE in these companies?
- Research question 3 (RQ3): what high-level models describe the KM process for SE in Australian SE companies?
- Research question 4 (RQ4): what are the impacts of leadership, technology, culture and measurement as enablers of the KM process for SE in each of these companies?

3.1. Research method

This research is conducted using two industry-based case studies, since case studies are "especially suitable for learning more about a little known or poorly understood situation" [32]. Using both a quantitative and qualitative approach as complementary epistemological orientations is helpful in triangulation, and offers better understanding of the findings. Thus, this research employed a mixed research methodology to conduct the case studies. Both qualitative (i.e., semi-structured interviews) and quantitative methods (i.e., a detailed questionnaire) of data collection were used for three reasons, namely triangulation, comparison and in depth understanding.

3.1.1. Semi-structured interviews

Semi-structured interviews were conducted to understand the process used for managing knowledge during software development and the interviewees' perception of how to manage SE knowledge during software development process. The interview questions primarily focussed on the current practice of KM in SE based on four elements. The elements namely types of knowledge, motivation for sharing knowledge, knowledge sources and KM systems [6,13,39].

The main purpose of these interviews was to collect qualitative data complemented and triangulated by the results collected from the questionnaire. Each software developer had to address issues arising from dynamic group work, the need to combine both generic and specific skills, and what knowledge to retain [6]. The interview sessions allowed each participant to clarify question phrasing, as well as to ensure that participant responses addressed the research objectives of this study. The interviews were held at suitable times for participants, with the duration ranging 45–60 min. They were conducted in private meeting rooms at the premises of the organisations involved. Each participant was given an overview the research, and detailing the participant's rights and responsibilities. Each participant then was asked a series of questions, with both the researcher and participant seeking clarification or more information wherever required. The proceedings of each interview were recorded verbatim electronically. Transcription of interviews was over 100 pages length.

3.1.2. Questionnaire

The questionnaire is based upon elements of existing KM models, as well as activities suggested as being components of the KM process used for SE. It consists of three sections: (1) background information about each participant and a nominated project, (2) major activities performed in the KM process for SE (i.e., the seven KM activities in Table 1) and (3) the KM enablers (i.e., the four enablers in Table 1) for the KM process in SE. Finally the questionnaire also included an appendix which provided the definitions of the terminology and KM activities that we used within the questionnaire. The main categories of the questionnaire items were between open-ended and close-ended questions. The questionnaire developed for this study was designed to contain a balance of these types. A pilot study was then completed, involving participants with SE experience. The results of the pilot were analysed, and prompted refinement of the research instrument before it was applied to the target audience in the two selected companies.

The researcher who conducted the interviews also administered the questionnaire to each participant, ensuring consistency in administering the research instrument, with the duration ranging 60–90 min in a private session, and only the researcher and the participant in attendance. Each participant was instructed to complete the questionnaire and to provide additional comments/feedback verbally at any stage.

3.2. Selection of case studies and data collection

In this study, the two software companies (Company A and Company B) were selected primarily on the basis that they both claimed to apply KM practices in their software development work. One of the authors who once worked in Company A earlier was also working in Company B at the time of investigation carried on.

Company A is a multinational organisation, and have a KM special interest group, with an area allocated on their intranet from which KM-related articles and contributions can be downloaded. Company B is also an Australian organisation. In this company there were sessions held where an employee, considered being an ‘‘expert’’, presented in his/her work other employees. Both organisations recognised the knowledge that each of their employees possessed. They both assigned time and resources to try and tap into this knowledge, with mixed results. Both of these companies have already invested significant amounts of time and resources in improving the quality of their software development processes, and therefore they would be ideal candidates for evaluating the effectiveness of a KM process as it is applied in SE.

Data were collected through the questionnaire and interviews. A total of 12 people (six people from each company) were participated in this study. Data were collected from two projects at each company. The same subjects who participated in interviews also filled out the questionnaires.

All participants were in software development roles in the projects examined which included one project team leader, six software developers, two programmers, one systems analyst and two consultants. Software developers were targeted because of their exposure to many aspects of the software development lifecycle, as well as their broad technical perspective. Their experience in software development ranged from 2 to 24 years.

3.3. Description of the companies and the projects

For reasons of confidentiality, the two companies are identified by the letters A and B, followed by a project number to represent each project within the respective company. Characteristics of these companies and the projects are provided in Table 2.

Table 2
Characteristics of projects

Project	Product	Customer	Locations	Customer access	No. of team members	Effort (PM)	Size	Priorities
A1	Enhancements to an Australian Internet banking website	External	Same site	Moderate	11–25	240	Large	Time/cost/quality
A2	Processing application for online home loan applications of an Australian bank	External	Same site	Moderate	1–10	28	Medium	Functionality
B1	Administration application for a large Australian government agency	External	Different sites	Moderate	26–100	967	Large	Cost
B2	Administration application for large Thai bank	External	Different sites	Hard	26–100	111	Large	Time

Company A is in the IT services industry, and has a global workforce of over 130,000 people operating in over 60 countries. Approximately 35% of the workforce in software development roles. Both of the projects studied produced customer-specific products for external clients. In both projects there was close interaction with the customer throughout the project duration.

- *Project A1*: the objective of the project A1 was to support and make enhancements to a large Internet banking website. Participants were required to produce code for releases scheduled every few months. The project had between 11 and 25 team members for its duration, with approximately 70% possessing technical qualifications. The total effort was approximately 240 person months. Thus, the project was classified as large.

- *Project A2*: in project A2, project team members were involved with developing an application to process online home loan applications for a major Australian bank. The customers were external customers, with customer access regarded as moderate. The project had between 1 and 10 team members for most of its duration, with approximately 90% possessing technical qualifications. The total effort was approximately 28 person months. Hence, the project was classified as being medium-sized.

Company B is also in the IT services industry, operating primarily in Australia and employing 2500 people. Approximately 40% work in software development roles. Both projects produced customer-specific products for external clients. In both projects there was close interaction with the customer throughout the project. Both time and cost were major priorities for both projects.

- *Project B1*: in project B1, project team members were involved in developing an administrative system for a large Australian government agency. The project team was located at a different site to the customer with customer access being classified as moderate. The total effort was approximately 967 person months. Therefore, the project was classified as being large.
- *Project B2*: in project B2, project team members were involved with developing a superannuation system for a financial institution in Thailand. The project team consisted of between 26 and 100 people, with approximately 90% of the project team possessing technical qualifications. The total effort was approximately 111 person months. Consequently, the project was classified as being large.

4. Data analysis and results

The role of KM as a success factor in large software projects is widely accepted [6]. Central to this role is storing project data as well as experiences and problems or issues noted by engineers into experience repositories or factories. This section focuses on the data captured from the two case studies during interviews and from the questionnaire, and provides a detailed analysis of these data.

4.1. Current practices for KM in SE

In analysing RQ1, this section examines the current KM practices in two companies. The objective was to assess the software developers' understanding and perception of KM, which in turn, would determine their commitment to the practice of KM. We wanted to assess whether or not there was a common/shared understanding of the KM among them, and in what they were doing.

Data for this question was captured during the semi-structured interviews. As a first step in this direction, interviewees were asked to define knowledge and KM. The second step towards understanding current KM practices was to assess the interviewees' level of awareness of the currently available knowledge sources as well as tools used, and techniques and methods applied. Although quite tempting, no direct reference was made at this stage to KM activities, as this is the subject of the next research question, described in Section 4.2. The data collected were then used as basis for describing the KM practices for SE in the two companies.

4.1.1. Company A

While participants from Company A believed that KM would ultimately improve the quality of the work they produced, it became apparent that the tools and techniques being used were inappropriate or ill conceived. Most participants affirmed that their colleagues were an invaluable source of acquiring knowledge (i.e., *Informal networks* in Table 1), although many observed that they relied on the extent of *personal networks* rather than the knowledge managed collectively by their company (i.e., *organisational practices and routines* in Table 1). It was noted that management has heavily promoted standards and process methodologies, with project documentation required to conform to these activities.

In project A1, participants were aware of the potential benefits associated with KM, despite facing difficulty in their software development processes. Most participants were able to provide appropriate definitions for both knowledge and KM, although there was some confusion regarding the distinction between information and knowledge. For instance knowledge is defined by the three participants as follows: "*Understanding of systems and technologies*," "*Relationship between concept and physical world*" and "*Skills, experience (i.e., "The Ropes") needed to understand something*."

Although *personal networks* for Knowledge Acquisition had not been formalised, participants disseminated knowledge by engaging in regular discussions with each other during coffee breaks, team meetings and through email (i.e., *Informal networks* and *Groupware* in Table 1).

In project A2, KM was positioned as an important component in the project plan. As far as the knowledge storage process of KM was concerned, all participants were conscious of organising their knowledge for future use. They were also conversant with the structure of project repositories and group directories. However, during the knowledge storage process, the intrinsic motivation of project team members was relied upon in order to have the knowledge produced from the project stored in a more explicit form. In addition, participants encountered difficulty when asked to define the concepts of knowledge and KM. As in project A1, a uniform process model for applying KM to software development work could not be identified.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

4.1.2. Company B

As in Company A, KM was considered to be an important component of the work of software developers. The influence of standards, particularly the ISO9000 series, on KM was visible also in the two projects examined. The absence of KM systems and the limited search facilities for repositories and archives highlights the relative immaturity of the KM process for SE in Company B.

In project B1, regular feedback was delivered to participants after reaching each milestone. The majority of participants were able to supply appropriate definitions for both knowledge and KM. For example one participant described knowledge as "*information a person has in his/her head or that can be asserted or interpreted based on past experience*" and KM as "*structuring and organizing information in a way that is easily accessible*."

The location of experts was heavily reliant upon the extent of *personal networks*. The amount of discussion and networking among project team members and peers was limited, perhaps as a result of the geographical separation of the software developers involved.

In project B2, confusion surrounded the definitions of both knowledge and KM. Most participants were able to identify and resolve problems using their existing knowledge, although searching facilities were limited. All participants were able to adapt existing knowledge in order to produce innovative solutions to problems. In addition, sharing knowledge with others was undertaken with a view to enhance the capabilities of the project team, ultimately focused on producing a quality product.

4.2. KM process activities

In analysing RQ2, this section examines the KM-related activities that are applied in SE contexts. One of the objectives of our study is to capture both explicit and implicit activities in order to understand the nature of the KM processes during software development in selected companies. We describe *implicit activities* as the activities performed by team members in an ad-hoc manner; they do not involve the sharing of knowledge about the task [46] and ultimately lead to implicit or tacit organisational knowledge. Performing activities explicitly during the project lifecycle plays an important role in the prevention of such conditions. We define *explicit activities* as those activities that follow the stated software development process in a visible and accountable manner to the whole project team [46]. Explicit activities provide retrievable information through developing explicit knowledge that can be used as new components in the construction of organizational knowledge. Furthermore, they increase the potential problem-solving ability of a development team and also facilitate decision-making activities during the software development process, through providing the possibility of knowledge transfer and knowledge integration; the latter has a significant role in reducing software development challenges [49]. Thus, capturing explicit and implicit activities during the project life cycle is a way of better understanding the nature of the KM process on the one hand, and obtaining an implicit evaluation of the level of explicit knowledge within the company, on the other.

In order to examine the relationship between the practice and the research, each participant was asked whether the seven KM activities provided in the questionnaire were performed explicitly, implicitly or not performed at all. They were also required to assign a percentage to each activity in relation to the total effort for KM in SE to be. Fig. 1 summarises the distribution of explicit and implicit KM activities in four projects that we investigated in two case studies.

4.2.1. Company A

At Company A, all seven KM activities were being performed. The results indicate that most participants performed Knowledge Adaptation implicitly. Knowledge Organisation and Knowledge Distribution activities were mainly performed explicitly, whereas the remaining activities were performed both implicitly and explicitly in each of the projects examined.

In project A1, the majority of participants understood Knowledge Identification and Knowledge Organisation to be performed explicitly. Both Knowledge Creation and Knowledge Application were performed implicitly. Knowl-

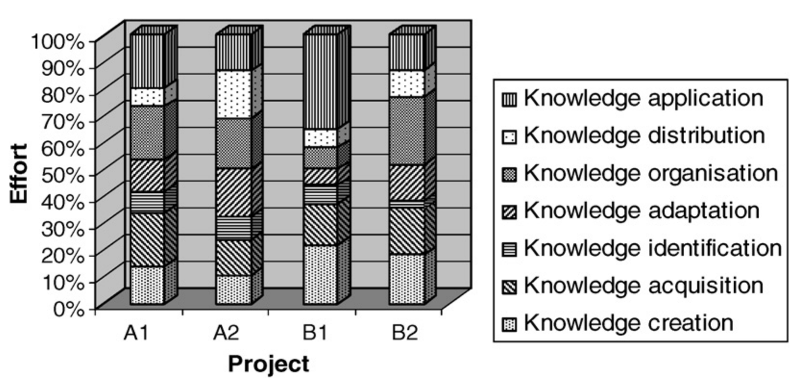

Fig. 1. Average efforts invested on seven KM activities in four projects.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

edge Acquisition, Knowledge Adaptation and Knowledge Distribution were performed both implicitly and explicitly.

In terms of the total amount of effort allocated to each activity, both Knowledge Organisation (20%) and Knowledge Application (20%) required the most effort. Knowledge Acquisition (19%) was closely behind, while Knowledge Identification (8%) and Knowledge Distribution (7%) required the least amount of effort.

In project A2, most activities in the proposed KM process model were performed explicitly. The responses of participants revealed that Knowledge Acquisition, Knowledge Identification, Knowledge Organisation, Knowledge Distribution and Knowledge Application were all performed explicitly. Knowledge Adaptation was performed implicitly by all participants, while Knowledge Creation was performed both implicitly and explicitly.

In terms of the total amount of effort allocated to each activity, Knowledge Organisation (18%) and Knowledge Distribution (18%) required the most effort. Knowledge Adaptation (17%) followed closely behind, while Knowledge Identification (9%) required the least amount of effort.

4.2.2. Company B

At Company B, all seven of the proposed KM activities in this study were performed. The results indicate that most participants performed Knowledge Creation and Knowledge Application explicitly, whereas Knowledge Identification and Knowledge Organisation were mainly performed implicitly. The remaining activities were performed both implicitly and explicitly in the projects analysed.

In project B1, Knowledge Creation, Knowledge Acquisition and Knowledge Application were performed explicitly. All participants performed Knowledge Identification, Knowledge Adaptation, Knowledge Organisation and Knowledge Distribution implicitly. In relation to the total amount of effort allocated to each KM activity, Knowledge Application (35%) required the most amount of effort. Knowledge Creation (22%) and Knowledge Acquisition (15%) were also allocated a significant amount of the total effort invested in the KM process for SE.

In project B2, Knowledge Organisation and Knowledge Distribution were performed explicitly. The remaining activities were performed implicitly, implying that the visibility of the KM process used for SE in project B2 was limited. That is, KM process activities were embedded within the context of the software development lifecycle followed in this project. In relation to the total amount of effort assigned to each KM activity, Knowledge Organisation (26%), Knowledge Creation (18%) and Knowledge Acquisition (17%) required the most amount of effort. Knowledge Identification (3%) required the least amount of effort.

4.3. Descriptive KM process models

In analysing RQ3, this section constructs high-level descriptive models of the KM processes for SE using the data collected from the two case studies. The results are drawn from the questionnaire which required the participants to describe the KM process and the activities in each phase of a specific project they played a part in. High-level process models are constructed for each project and then compared with models in the existing literature. Examples of the process models identified by the participants are illustrated in Figs. 2–5. The left hand side of the figures list the phases of the KM process (i.e., terminology used in the organization) in a particular project. The right hand side of the figures display the KM activities (in rectangular boxes) performed at each phase (i.e., the seven KM activities provided in the questionnaire). Solid and dashed lines are used to illustrate whether the activities are performed explicitly or implicitly. Note that the size of the boxes is of no importance. The order of the boxes shows the progression of the activities in each phase.

The results so far of this study indicate that KM activities in SE were embedded within the context of the system development lifecycle model embraced by each company. Although each company had a process improvement initiative, the results suggest that process improvement was driven more by commercial necessity rather than a desire to leverage the knowledge they possess. Company A, in order to retain one of the biggest customers, wanted to get to CMMI Level 5. Both companies operated with a large number of consultants, who were mostly sent out to client sites where they could be billed for their time. Both companies had far fewer staff at head office, performing administrative work that was not billable. When it comes to making money, this came from billing out consultants at high rates, rather than having these people work internally on process improvement activities.

Since the application of KM principles in SE remains an emerging area, it was anticipated that participants might encounter difficulty when asked to describe the components of a KM process that they apply in their software development work. However, most participants were able to associate the KM activities proposed in this study with various stages of the system development lifecycle in their projects.

Each project of each company used a different KM process for its SE. Therefore, a single high-level descriptive model of the KM process for SE could not be formulated. While this is a reflection of the relative immaturity of KM in SE, nevertheless several high-level models were produced for each project. Each model depicts the phases that comprise the system development lifecycle models being used in one project, with a mapping to KM activities that are proposed in this study.

4.3.1. Company A

In project A1, the first participant concurred with the list of activities presented in this research as suitable components of a KM process model used for SE in Company A. The remaining two participants disagreed with this assessment, and considered KM activities to exist within the context of system development lifecycle model prescribed by their company.

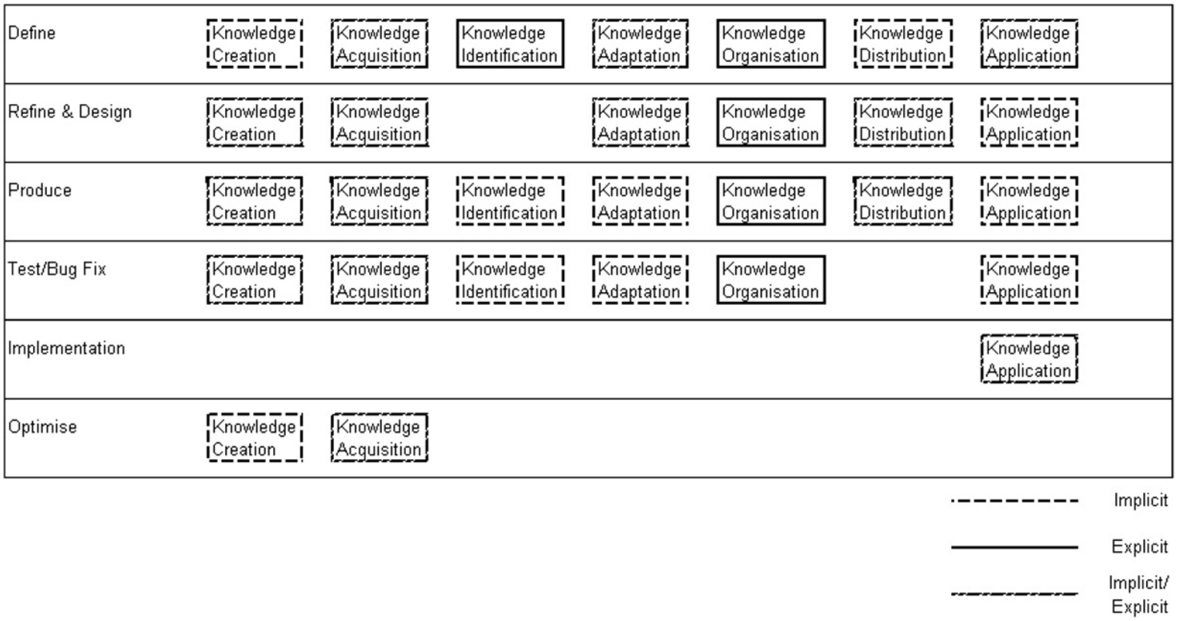

Fig. 2. KM process model for project A1.

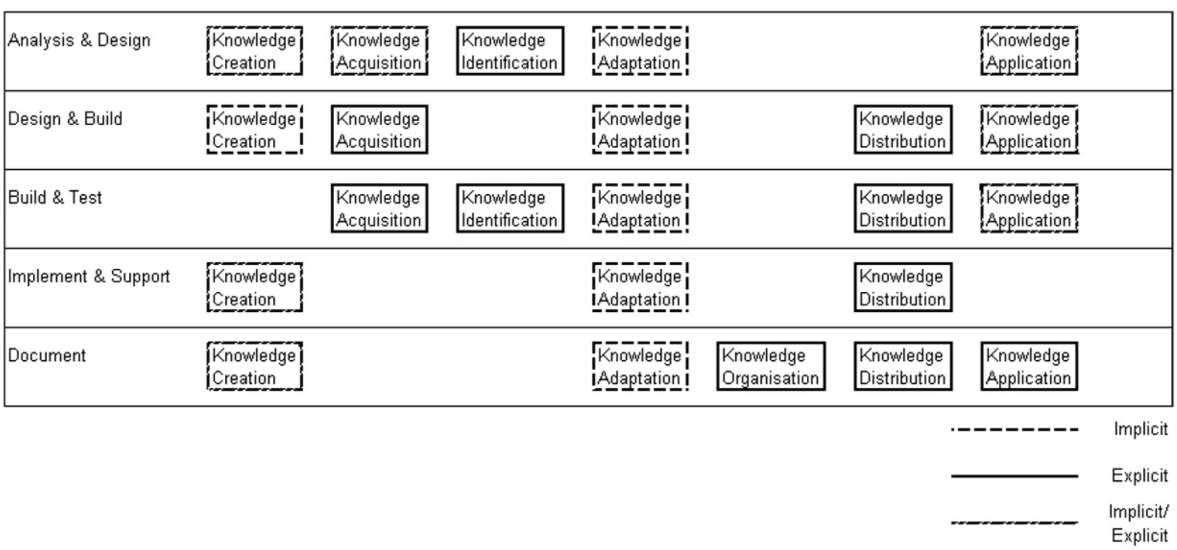

Fig. 3. KM process model for project A2.

The high-level model for project A1 depicts six main phases (see Fig. 2). In both the Define and the Produce phases, all activities identified in this study are performed. It is interesting to note that in every phase in which Knowledge Organisation is performed, it is performed explicitly. In addition, Knowledge Acquisition is performed both implicitly and explicitly in every phase in which it is performed. The responses suggest that the high-level model presented below is predominantly linear, closely resembling a waterfall model for software development.

In project A2, every participant considered the KM process to be embedded within the context of the system development lifecycle followed in his/her project (see Fig. 3). The high-level model for A2 illustrates five main phases, with multiple KM activities being conducted in each. In contrast to the high-level model constructed for project A1, a phase relating to documentation is represented. Some participants acknowledged that this phase was being iteratively conducted throughout the entire software development process.

An activity that is performed in every phase of the high-level model presented for project A2 is Knowledge Adaptation. In addition, Knowledge Creation, Knowledge Distribution and Knowledge Application are performed in all but one of the phases depicted. On the other hand, Knowledge Organisation is performed only during the Document phase.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

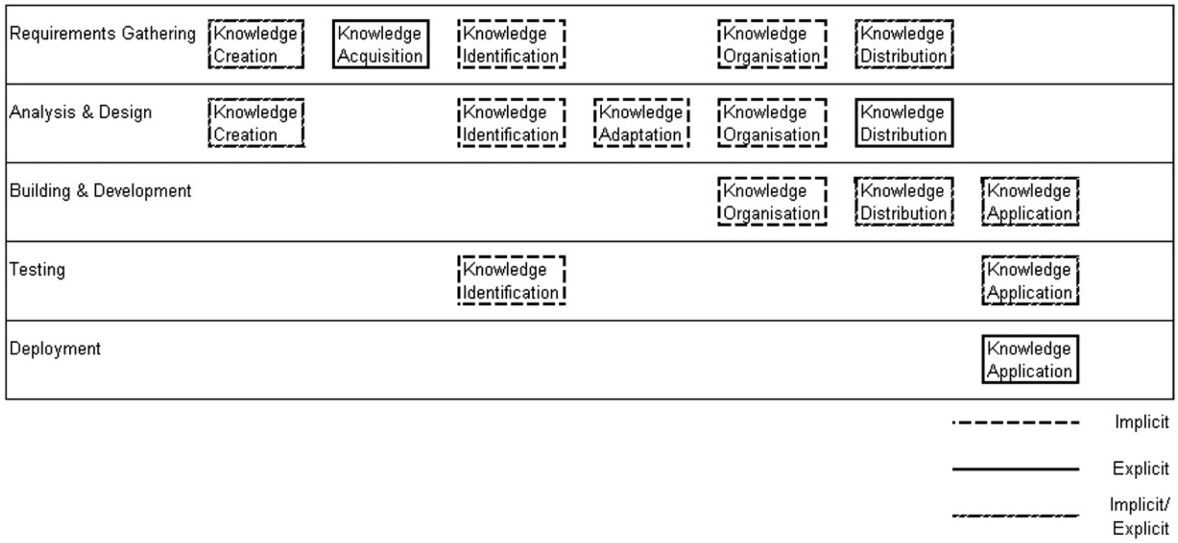

Fig. 4. KM process model for project B1.

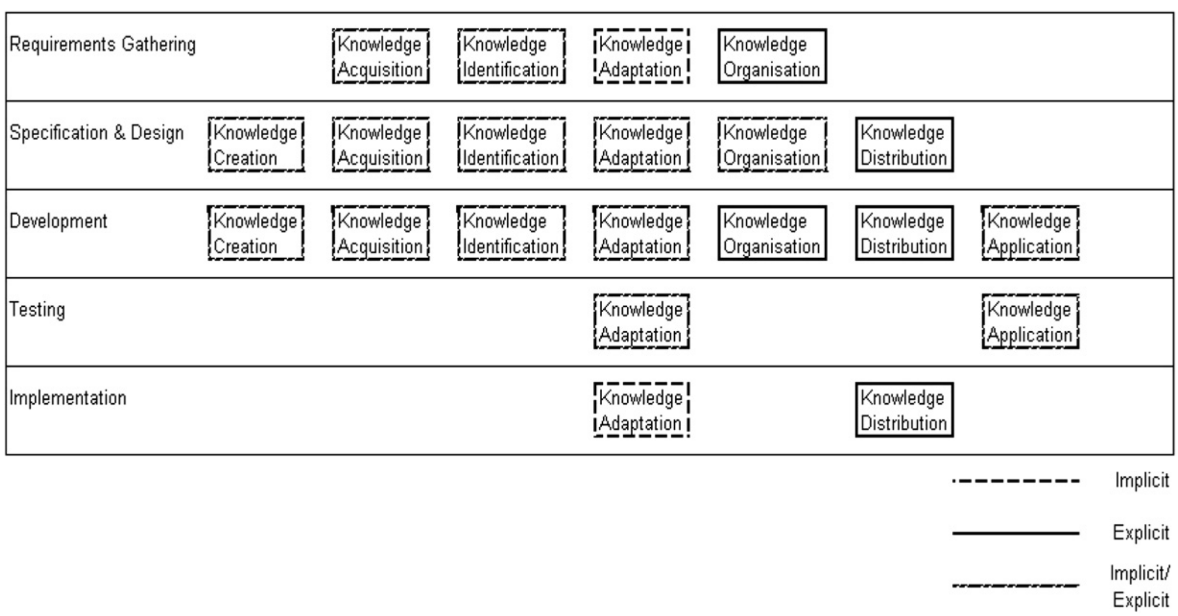

Fig. 5. KM process model for project B2.

4.3.2. Company B

As shown in Fig. 3, in project B1, all participants considered the KM process to be embedded within the software development lifecycle adhered to in their project. The high-level model for project B1 (Fig. 4) illustrates five main phases. Knowledge Identification, Knowledge Adaptation and Knowledge Organisation are all performed implicitly in this model. The remaining activities are performed both implicitly and explicitly.

It is interesting to note Knowledge Creation is only performed in the earlier phases of the model, along with Knowledge Acquisition and Knowledge Adaptation. Alternatively, Knowledge Application is performed only in the latter phases of the model.

In project B2, all participants considered the KM process to be embedded within the software development lifecycle model followed in their project (see Fig. 5). As with project B1, the high-level model for project B2 illustrates five main phases.

In the Development phase, all activities identified in this study were performed. Knowledge Distribution is performed explicitly in every phase of this model, while Knowledge Organisation is performed mainly explicitly. The remaining activities are performed both implicitly and explicitly. It is interesting to note that Knowledge Adaptation is performed in every phase of this model, while the remaining activities appear primarily in the first three phases of this model.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

4.4. KM enablers

KM essentially brings together three organisational resources – people, processes and technologies – to enable the organisation to use and share information more effectively. KM enablers, in Table 1, essentially imply that leadership, technology, culture and measurement must be addressed when applying a KM process in organisations, rather than considering it in isolation.

In analysing RQ4, this section evaluates the impact of KM enablers on the KM process for SE. Data is collected through the questionnaire, which required each participant to respond questions relating to factors in the KM process for SE used on his/her project. In addition, each participant was asked to rate each of the *importance* and *applicability* of each enabler of the KM process for SE using a Likert scale. This metric was used to establish an overall perspective of how each enabler was viewed by the participants, and to solidify links with activities in the KM process for SE (see Fig. 6).

4.4.1. Company A

At Company A, leadership emerged as having the most significant impact upon KM in SE. A high-level of support could be found also for both technology and culture, although there was greater variance in the ratings participants assigned to them. Leadership was assigned an average rating of 4 for both its importance and applicability to the KM process for SE. Although half of the participants assigned a rating of 2 or 3 for the importance of technology, a high rating for its applicability was consistently recorded. For culture, a similar variance was displayed with each of the participants assigning a rating of 3 for both its importance and applicability. The ratings assigned for measurement suggest that it is the least appreciated of the four enablers, for its importance and applicability to the KM process for SE.

4.4.2. Company B

At Company B, leadership, technology and culture emerged as significant enablers of the KM process for SE. In terms of importance, all three were consistently given high ratings. However, participants were divided about the applicability of culture. Measurement had the lowest ratings for applicability, despite every participant's rating it highly for its importance to the KM process for SE. This dichotomy may be explained by the difficulty many participants expressed when asked to identify appropriate measures that could be used to determine the knowledge they possess. The lower ratings for applicability may also be a product of inappropriate measurement strategies adopted by management, generating cynicism towards their use to ascertain job performance.

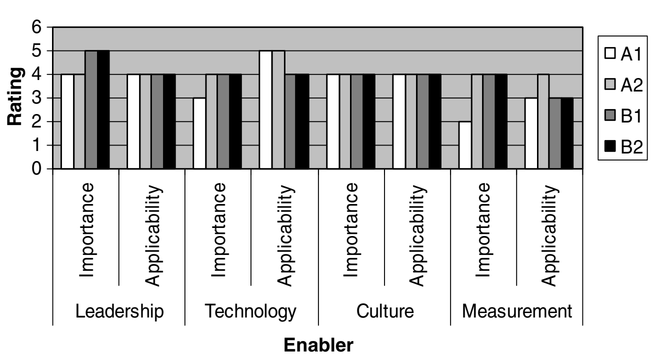

Fig. 6. Four enablers in four projects from Company A and Company B.

5. Discussion of results

This section presents discussion of findings from questionnaire by using supporting quotations from the structured interviews with software developers.

5.1. Current practices for KM in SE

This section discusses the current practice of KM in SE based on four elements focussed on our interviews, namely types of knowledge, motivation for sharing knowledge, knowledge sources and KM systems.

5.1.1. Types of knowledge

In both companies, the type of knowledge applied for software development in the projects examined was primarily implicit. Success stories were not frequently shared between project team members. However, a number of structures, particularly in Company A, had been established for storing explicit knowledge, such as group directories, archives and repositories (i.e., *document management systems in Table 1*). Despite the accessibility of explicit knowledge, the third participant from project A1 highlighted the difficulty in converting implicit knowledge to explicit knowledge: "*To be honest, I'm not sure how you'd formally put it down in a way that you could reference it*." One of the participants in B1 project, who had been working about two years in Company B, was unsure of the existence of formal archives and repositories, as well as the ability to match an individual with the knowledge he or she possesses "*I don't think we've got any kind of directory that ties people in, technically, with their skill set*." The third participant from project B2 commented that the particular system used for this project mandated individuals with specialist expertise: "*Unless you know that certain programs and files related to each other, you're never going to know*." In short, the predominant knowledge type in both companies was tacit and implicit and the process for locating individuals with knowledge relied upon the scope of personal networks.

5.1.2. Motivation for sharing knowledge

In both companies, most participants claimed that their primary motivation for sharing knowledge with other team members was the desire for everyone to be able to perform their duties at a similar level. In Company A, the majority

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

of participants believed career progression was associated with the ability to continually move between different projects. The second participant from project A2 asserted: "*Whenever I was able to become redundant, I could find a better job.* At Company B, most participants maintained that their primary motivation for sharing knowledge was to ensure all project team members were able to perform their duties at a similar level. While the third participant from B1 pointed out that there is always to something to gain by sharing knowledge, the first participant from project B2 commented that knowledge can be a factor in delivering more stimulating work. Some participants discussed the significance of obtaining appropriate KM tools, as well as the explicit recognition of KM activities in project schedules. The third participant from project A2 sensed these two factors would engender increased motivation to engage in KM activities.

5.1.3. Knowledge sources

In Company A, the majority of participants from project A2 preferred a '*Learn by Doing*' approach to acquiring knowledge, rather than attending formal *training* (see Table 1) programs. The knowledge gained from participating in formal *training* programs was considered to be irrelevant unless it could be applied immediately. Furthermore, both the time and cost of such programs meant they were offered infrequently to the participants. Finding subject-matter experts was an informal process, with an onus being placed upon participants and their managers to leverage personal networks. The third participant from project A1 maintained that colleagues were the most useful source of knowledge (*informal networks* and*personal networks*), although questioned their ability to organise and distribute knowledge with others. In Company B, *personal networks* were seen as the primary mechanism for transferring knowledge between project team members. The use of third-party knowledge was common, with the main sources being the Internet and magazines, overshadowing project repositories and KM systems maintained such as in Company B. Most participants were comfortably using this knowledge. The third participant from project B2 declared that third-party knowledge could be applied without major adaptation, depending upon the validity of its source. There was also some confusion as to whether project repositories and archives were accessible organisation-wide.

5.1.4. KM systems

In Company A, all participants were aware of repositories and archives containing knowledge from earlier projects. New employees were introduced to the main group directories as part of their induction programs, underlining the importance assigned to maintaining the currency and relevance of knowledge bases. Company A had also tried to foster online communities, to assist employees in communicating with others sharing the same interests. The first participant from project A2 described the process for submitting a question about the preparation of test scripts to a global KM community maintained by Company A as follows "*...we had some templates, but then we decided to post a message to the GSMS community. In less than an hour, we had around 20–30 responses*."

The responses of participants from Company B reveal that KM systems, e.g., *document management systems*, were not widely available. There was frustration with the avenues for communicating feedback. There was optimism about the effect a KM system could have on knowledge re-use, although it remained largely untracked. In project B1, a software development framework developed for a similar project was re-used. The first participant from project B1 stated: "*What we do now is create a framework for what we did from the last project. We're using, basically, the same framework now*." It can be argued that the use of a KM system would eliminate the risk of re-using knowledge without also accounting for the idiosyncrasies of each project.

In summary, findings were similar to previous studies [13,15] and showed that most of SE knowledge was tacit and contextual in nature which was difficult to capture and store. In terms of knowledge management strategy, our findings complement Hansen's [22] work. In other words, knowledge was managed mostly documenting and storing carefully (i.e., the codification strategy) in Company A, whereas it was mainly managed through personal networks (i.e., personalisation strategy) in Company B. Although, participants believed the usefulness of knowledge sharing, they were not able to utilise some of the KM systems presented in Table 1, e.g., formal training. Among those systems, the personal networks, informal networks, groupware and third-party knowledge such as Internet and magazines were the most commonly used systems. On the other hand, expert systems, BPM systems, new product and services technologies, and new market technologies were not present in these companies yet at the time that they were investigated. There was a need to formalise knowledge sharing of practices, especially in Company B, while also supporting informal and ad-hoc knowledge sharing. A detailed discussion on knowledge sharing can be found in (chap. 4, [6]).

5.2. KM process activities

This section focuses on the application of seven KM activities in two companies.

5.2.1. Knowledge Creation

Knowledge Creation was performed both implicitly and explicitly across all projects examined. All 12 participants recognised that this activity was being performed in his/her software development project. This is encouraging, implying that there is general consensus that software development does produce a large amount of knowledge. A medium amount of effort was allocated to this activity, ranging from 10% to 25% of each person's software development time.

An A1 participant concluded that project reviews did not always produce a balanced assessment of positive and negative outcomes: *"...you rarely have these insightful comments like 'A really, really good way would have been this.' It would just be a case of 'this screwed up. What did we do wrong? Let's make sure we don't do it again."* One of the B1 participant expressed a preference for discussing problems with colleagues at the time these problems are discovered, rather than investigating more explicit knowledge sources: "*Just, because that's the first place you go to another team member*."

At both Company A and Company B, team meetings were acknowledged as crucial opportunities for team members to present new ideas, to offer advice as well as to commit to processes and methodologies. The first participant from project B1 portrayed team meetings as being pivotal to the creation of new knowledge: "*You talk about problem-solving, but there's also things like guidelines. We talk about how we should do this, how we should do that. And, from that, we do produce documents*."

5.2.2. Knowledge Acquisition

Knowledge Acquisition was performed both implicitly and explicitly across all projects examined. Each participant recognised that this activity was being performed during the course of his/her software development project. In projects A2 and B1, this activity was performed mainly explicitly, while in project B2 it was performed mostly implicitly. A medium amount of effort was assigned to this activity, ranging between 10% and 20% of each person's software development time. At Company A, team meetings were seen as ideal opportunities to acquire knowledge from others in a project. For software developers, code reviews were seen as even more useful, since more specific knowledge generally emerged. At Company B, feedback about project status and observation of problem areas was ordinarily encouraged. The principal forum, as in Company A, was the team meeting. In addition, colleagues were regarded as being the most valuable source of knowledge. It could not be determined whether this perception was because colleagues offered knowledge that was easily applicable to project-specific problems, or because of the ease of knowledge transfer.

5.2.3. Knowledge Identification

Knowledge Identification was performed both implicitly and explicitly across all projects examined. In project B1, all participants performed this activity implicitly. In projects A1 and A2, this activity was performed explicitly by most participants, while in project B2 this activity was performed mainly implicitly. A small amount of effort was assigned to this activity, ranging between 0% and 10%.

At Company A, some participants encountered difficulty comprehending the nature of this activity, and doubted this activity was being performed in their normal work as software developers. The first participant from Company A, when encountering the term, remarked: "*Knowledge Identification ...I don't think we do that at all*." The second participant from project A2, in discussing process improvement suggestions, noted: "*At least this project had documented roles and responsibilities. You just have to refine the structure of what everyone has to do*." The third participant from project A1, when asked about identifying current software development problems using knowledge gained from previous work, stated: "*Mostly, it's just stuff I've stored in my head. I'm pretty hopeless with documenting things like that*."

At Company B, participants mentioned that areas for process improvement were usually identified by individuals, and then raised and addressed in forums such as team meetings. Most participants were comfortable with the reuse of knowledge from earlier projects, in order to identify current problems. However, this knowledge was often tacit or implicit.

5.2.4. Knowledge Adaptation

Knowledge Adaptation was performed both implicitly and explicitly across all projects examined. In project A1, this activity was mainly performed explicitly. In project A2, all participants performed this activity implicitly. In projects B1 and B2, this activity was performed mainly implicitly. A small amount of effort was assigned to this activity, ranging between 5% and 20%.

The majority of participants from both Company A and Company B stated a preference for generic knowledge, citing its applicability to multiple situations. This preference may arise from the variety of different business problems that software developers are asked to consider when working for an IT services organisation. The third participant from project A2 asserted that knowledge remained useful only within the context of a specific project: "*I would say it is very limited knowledge that isn't useful to...anything in the future*."

Most participants considered updating a knowledge base to be an onerous activity, and one that was not assigned high priority in the projects they were involved with. The second participant from project A1 noted that updating the knowledge base with a particular knowledge item was left to those directly responsible for this item: "*The most you might do...I point them at your reference, and say 'Have we considered doing this?*" A key inhibitor identified by participants in relation to updating a knowledge base is time, exacerbated by constant movement between projects and geographic locations. The third participant from project A1 commented: "*Most of the time, you'll find that you don't get the chance to do it, whether you want to or not*." The third participant from project B2 asserted that updates and changes to address identified problems with existing knowledge was cumbersome: "*Yeah, I would suggest it to the people who happened to have access*."

5.2.5. Knowledge Organisation

Knowledge Organisation was performed both implicitly and explicitly across all projects examined. In project A2,

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

all participants performed this activity explicitly. In projects A1 and B2, this activity was performed mainly explicitly, while in project B1 it was performed implicitly by most participants. A medium amount of effort was assigned to this activity, ranging between 5% and 30% per person. Many participants felt remote access was either too slow or not possible. The constant travel of consultants working in a technology services organisation was identified as a reason for the perceived difficulty in accessing archives and repositories.

At Company A, directory structures were closely linked to the use of specific method. Despite the presence of standardised directory structures, the first participant from project A1 was critical of the approach toward organising knowledge that the project method advocated. While the use of standards may have resulted in the conversion of some implicit to explicit knowledge, the third participant in project A1 presented a conflicting view: "*Pretty much 99% of the knowledge that I have gained from this project is just in my head*." At Company B, many documents from earlier projects were made available, although the searching facilities were primitive. In addition, many participants were unsure as to the existence of archives and repositories. The second participant from project B1 stated: "*...as far as I'm aware of, there's no central repositories with any kind of searching facilities.*"

While standards mandated that specific project documentation be produced and stored, the results hint that participants retained a significant amount of tacit and implicit knowledge from projects they had previously worked on. Although documents from earlier projects were made available, they were often difficult to locate due to the absence of a centralised search engine.

5.2.6. Knowledge Distribution

Knowledge Distribution was performed both implicitly and explicitly across all projects examined. In projects A2 and B2, this activity was performed mainly explicitly, while in project B1 it was performed mainly implicitly. In project A1, this activity was performed both implicitly and explicitly. A small amount of effort was assigned for this activity, ranging between 5% and 20% per person. The majority of participants assigned less than 10% effort for this activity. Each participant was able to identify multiple sources of knowledge about policies and procedures governing software development in the company. Some popular sources included the Internet, magazines and colleagues.

The responses of participants from Company A suggest that the primary motivator for sharing knowledge was overwhelmingly career progression. The second participant from project A1 commented: "*[But] the not altruistic answer is if they know it, I don't have to do it*." An additional reason for sharing knowledge was recognition of the risk of having important knowledge residing in a few individuals only. The third participant from project A2 professed: "*I always say this...if I get hit by a car tomorrow and I can't come to work, that's it*." At Company B, participants cited more intrinsic motivators when explaining their reasons for sharing knowledge. The first participant from project B2 stated that sharing knowledge contributes to better quality software: "*Sharing means others can solve problems faster and are less likely to be frustrated, resulting in a more efficient and happy working environment*."

5.2.7. Knowledge Application

Knowledge Application was performed both implicitly and explicitly across all projects examined. A medium amount of effort was assigned to this activity ranging between 10% and 35% per person. In projects A2 and B1, this activity was performed mainly explicitly, while in projects A1 and B2 it was performed mainly implicitly. In project B1, 35% of the total effort assigned to KM activities was for Knowledge Application, emphasising the important role it was seen to play in software development work.

Each participant was comfortable with re-applying his or her existing software development knowledge in a range of different scenarios. The level of satisfaction with third-party knowledge was generally high, with many a participant willing to conduct extensive searches until locating a knowledge item suitable for their purposes.

Most participants were able to conceive benefits associated with a system to track knowledge re-use by project team members. However, such a system did not exist in either organisations. In describing some of the potential benefits of such a system, the second participant from project A1 stated: "*Well, I guess it would allow the organisation to put a value on its knowledge. And, also to measure risk. They can see a gap between the knowledge they know about and the knowledge they ought to know about*." The first participant from project A1 highlighted the overhead involved in establishing systems to track knowledge re-use: '*The reason I say that is that if you apply that method to tracking, you may find that nine out of ten items are not reusable. One is reusable, but you spend effort in tracking all ten of them*."

In summary, Knowledge Organisation was perhaps the most prominent of these activities, being performed explicitly in both projects assessed in Company A. In Company B, there was much variance in both the manner in which KM activities were performed, as well as in the effort assigned to them between projects. This suggests that participants were at an earlier stage in understanding the KM process for SE than participants in Company A. Overall, results from both the interviews and the questionnaire reveal that KM activities are being performed inconsistently across both projects within each organisation, as well as between organisations. The results indicate that while a number of KM activities are being performed in software development projects, a robust KM process that suits the challenges presented by SE has yet to be standardised.

5.3. KM process models

In both companies, there was a medium level of KM process awareness, with a number of KM systems made available for participants to use. It can be argued that in both companies software developers had a limited appreciation for phases other than those concerning implementation. This may have limited their ability to conceptualise some KM activities being performed explicitly.

In Company A, project documentation was organised in a standardised directory structure within archives and repositories. The above two factors may have influenced participants consider Knowledge Organisation to be performed explicitly in each phase of the descriptive process models for projects A1 and A2 as shown in Figs. 2 and 3. Although Knowledge Organisation seemingly makes a smaller contribution to the KM process in project A2, the Document phase is claimed to be performed iteratively throughout each of the previous phases in its model. The third participant from project A1 conferred a sense of how the KM process was interwoven with the particular software development lifecycle followed by projects in Company A.

In Company B, the KM phases depicted in the high-level models for both projects B1 and B2 are analogous (see Figs. 4 and 5). While this information is to be expected at an organisation that is ISO9000 certified, at the same time the activities performed in each of the similar phases differ between projects. This contradiction again underlines the immaturity of KM in SE, as well as suggesting that implement in a KM process perhaps imposes a considerable overhead on project management.

As in Company A, fewer KM activities are performed in the latter phases of the high-level models presented. The lack of iteration in the high-level models also contrasts with the three models presented from literature. This lack of iteration suggests that software developers have a limited appreciation for phases involving testing and documentation. A by-product may be that this perception limits their ability to conceptualise some KM activities being performed explicitly in particular phases of the lifecycle followed by their project. The third participant from project B2 pointed out that the awareness of KM activities proposed in this study being performed by project team members.

While neither of the two organisations followed a standard process model for KM in SE, both were conscious of the role of ISO9001 and CMMi (Capability Maturity Model® Integration) compliance in terms of securing new and existing customers. The high-level KM process models (Figs. 2–5) allow the following useful observations:

- The developers have their own terminology, thus the questionnaire had to be designed in a way that it captures this information while allowing us to identify the common KM activities illustrated in Table 1.
- A generic KM process model for each software company cannot be produced, as the KM activities depend upon the nature of the projects. Hence, individual KM process models for each project had to be constructed. This result implies the high degree of dependency between the KM process and its context that prevents the use of a single model for each company.

5.4. KM enablers

This section discusses the enablers of KM process for SE in terms of leadership, technology, culture, process and measurement.

5.4.1. Leadership

As a major KM enabler, leadership practices influence organisational strategy in terms of how the knowledge is defined and knowledge assets are used and how KM is linked to organisational management. In Company A, a variety of roles were responsible for managing software development knowledge produced from projects. The second participant from project A1 cited the role of Quality Champion as an individual, such as business analyst, project manager or team leader, who is already performing another role within the organisation. Although unsure of whether a specific role for KM had been created, the third participant from project A2 assumed that each project team member was responsible for managing his or her own knowledge. A culture of implicit rewards for knowledge sharing was demonstrated in Company A, with the first participant revealing that the quantity of contributions was a key determinant in gaining recognition.

On the other hand, in Company B, the responses of participants denote that leaders have a significant influence upon the KM process applied for software development. In spite of this, participants supported the link between KM and improved business performance established by their leaders. The second participant put it succinctly "*You can pick up previous knowledge for new projects. More knowledge you have would help you over your competitors*." While managers were active in promoting KM, many participants believed it was ultimately their responsibility to ensure the knowledge they gained from project work was preserved and made accessible to others.

In both companies, although most participants could not identify specific roles that were responsible for managing knowledge in their projects, project managers were considered to the most likely to be held accountable. Almost no participant was able to specify a KM champion for his/her organisation, either at project-level or on a larger scale.

5.4.2. Technology

KM requires the introduction of certain criteria to decide which knowledge is most critical for the organisation, and to govern the factors, conditions and technologies that guide the activities of Knowledge Creation, integration and use [17]. Most participants sensed that technology was important to the KM process, despite it not being considered as applicable. The first participant from project A2 described technology as a system used to locate experts within the organisation while as the second participant from project A1 indicated that technology made sharing tacit and implicit knowledge with others problematic. The responses of participants demonstrate widespread cynicism about the applicability of KMSs. The third participant from project A1 noted that these systems were unwieldy and ineffective at delivering knowledge suitable for resolving specific problems.

The first participant from project A1 affirmed a preference for the Internet over KMSs maintained by Company A "*What you want is 'How to do this,' 'How to do that.' Those are very rare*." This preference may be driven by the plethora of Internet search engines that deliver an enormous amount of information to its users. This comment conveys also the difficulty in creating a KMSs that delivers content that is both general and specific, in a readily accessible format. The third participant from project A2, while satisfied with the use of technology to facilitate KM in SE, acknowledged: "*It's limited in that you can't just walk over to them and ask them a question, obviously*."

All participants at Company B expressed their disappointment concerning the absence of specific KMSs in their organisation. The responses of participants also highlighted the inaccessibility of project archives or repositories. Nonetheless, the second participant from project B1 described the importance of technology to the completion of project work "*And then, even things like our project, where we are split geographically, it's a backbone*." One B1 participant further reinforced the central role technology was required to play in software development projects. One B2 participant acknowledged that KM systems would be more advantageous than email in tracing knowledge used in a software development project.

5.4.3. Culture

Cultural support for KM in software development context can be described in terms of the encouraging of knowledge sharing, promoting open climate for the free flow of ideas, enhancing feedback from post mortems and so on. Most participants regarded annual performance reviews as ineffective measurement tools that had failed to institute a beneficial dialogue for knowledge sharing between manager and developers. The feedback delivered was considered to be too general, focusing on personality rather than performance in specific projects. A preference was exhibited for feedback to be delivered to software developers mostly during, or immediately after a project. A detailed discussion on application of post mortems in software companies can be found in [18,48].

The third participant from project A1 expressed concern about the KM culture that management were attempting to cultivate, in making contributions to KM systems mandatory "*And, forcing them down throats and forcing people to submit stuff to them is really the wrong way to go about it, I think*." In contrast, some participants mentioned that Company B had arranged information sessions, in which an employee possessing skills in a particular area would give a presentation on a topic of interest to others in the organisation. It is likely that these sessions were arranged by management to facilitate KM, particularly through enhancing communication channels among peers.

One of the participant in project B1 pursue an idea that if KM activities are explicitly outlined in project plans, it would lead an effective knowledge sharing. The point was that KM should not be seen purely as an altruistic task that a software developer works on in isolation from a project. Software developers are constantly busy, and hence it would make sense to develop something like an Experience Factory to perform KM "on the go". There were currently no explicit reward systems for an employee of Company B who effectively managed his/her own knowledge, or the knowledge of others. Nonetheless, the second participant from project B1 asserted that implicit rewards were offered by management "*[So] if management realises that you have been contributing a lot of knowledge to others, then quietly they give you a bonus*." However, the third participant from project B1 argued that financial rewards would not necessarily lead to an increase in KM activities and make the projects easier.

In providing a perspective on whether an association between knowledge and power had been formalised, mixed responses were recorded. The first participant from project A2 believed the link between knowledge and power was detrimental to quality, increasing risk as a by-product of some individuals becoming irreplaceable, whereas the second participant from project A2 stipulated how knowledge brings power.

5.4.4. Measurement

Measurement practices include how organisation quantifies its knowledge capital and resources. This research also focused on the KM measurement as an enabler for influencing the outcomes of the SE project as a result of evaluating various aspects of the projects' KM initiatives. The responses of participants reveal that Company A has invested limited resources in developing appropriate measures for evaluating the impact of KM in SE. Most participants in project A also encountered difficulty when asked to advance suitable measures for evaluating the knowledge of individuals. In contrast, an A2 participant believed that measurement for KM in SE was tied to annual performance reviews, as well as the content of online CVs. All participants at Company B encountered difficulty when asked to examine the relationship between measurement and KM. One B1 participant believed that measuring knowledge was problematic; possibly because tacit knowledge is not easily evaluated using traditional quantitative measures. Another B1 participant argued that by updating CVs on project completion, the organisation was able to determine the current level of knowledge of each of its employees.

Mixed responses were recorded when each participant at Company B was asked whether his/her true value to the company could be measured based upon the amount of knowledge s/he possessed. An occasional participant professed that some additional qualities also contributed to his/her true value. One B1 participant asserted that becoming an expert can lead to numerous recommendations.

In summary, our findings showed that among the four enablers studied in two companies, leadership emerged as having the most significant impact upon KM in SE in Company A and leadership, technology and culture were the most important enabler for Company B for managing knowledge (Fig. 6). Although participants acknowledge the importance of KM systems, they reported that there were no KM systems available for use software developers in Company B, whereas e-room, Intranet were among the commonly used KM systems in Company A. There were several measures (e.g., annual review or peer review) used to evaluate the knowledge possessed by software developers within the both organisations. However, majority of the participants believe these metrics were not appropriate to measure the knowledge and measurement was regarded the least important enabler. In other word, in both organisations measurement was rated lowest in terms of both its importance and applicability to the KM process for SE.

It can be argued that leadership was considered to be most significant because top-down KM strategies were seemingly being pursued by management. As Klint and Verhoef [27] discuss, KM in SE must be addressed from a bottom-up, operational perspective as well as a traditional top-down, managerial perspective. Since all participants were software developers, technology was considered to be an obvious mechanism for KM, despite current KM systems either being unsuitable or inaccessible. In addition, the crucial role that personal networks played in accessing tacit and implicit knowledge was seen as a key reason to foster a culture that encourages participants to share their knowledge with others. Although most participants were unable to name any specific measures that could be applied for KM in SE, there was strong support for the feedback delivered to software developers to be more regular and to include more effective measurement of tacit/implicit knowledge.

6. Validity threats

Four different types of threats to validity have been addresses in this study [53]. These are explained in the following sections.

6.1. Internal validity

Internal validity assesses the causal relationship between treatment and outcome. While all participants in this study were software developers, the diversity arising from educational backgrounds and industry experience was documented. The questionnaire used was developed with close reference to existing models and literature relating to KM in SE, and was piloted multiple times. It was also administered in interviews, with some lasting up to 90 min. Some participants provided shorter responses as more time elapsed. Many participants were conscious of the need to assign less than 60 min to an interview, given project constraints. Other participants may have attained a greater appreciation of the intended purpose of each section, hence enhancing their responses.

6.2. External validity

External validity measures the ability to generalise the findings of a study beyond the conditions that are particular to it. The participants selected may not adequately reflect the diversity of opinion present in the population examined. In this study, software developers were targeted, given their role within SE and their appreciation for process-orientation. Thus, it was felt that this sample would convey a realistic perspective of the KM process applied for SE in today's Australian companies. Hence this research cannot, and should not, be taken as representative of the Australian software engineering industry as a whole. This issue highlights the need for replication of this type of study. Both sample size and timing also influence the external validity of a study. The small sample size used may indicate that any conclusions drawn from this study are not generalisable outside the context of the IT services industry in Australia. The nature of this study is predominantly exploratory; with an emphasis upon describing the KM process used in SE practice in the period 2004–2005. It is expected that future research will reveal more advanced thinking about the KM process in industry, hence altering the generalisations that can be made.

This threat is concerned with the ability to generalise the findings beyond the actual study. In this study, selection bias and sample size are considered to have influence on the external validity. To avoid a selection bias, the participants of this study are selected according to their roles within the software team and based on their experience. However, the small sample size is a major threat to the external validity of this study. It may have affected the conclusions we have drawn and hence this research cannot, and should not, be taken as representative of the Australian software engineering industry as a whole. It highlights the need for replication of this type of study. Finally, skills and experience levels of team members and managers as well as differences in team cohesiveness and synergy from project to project are factors which have been ignored in this study. These can potentially confound our analysis and indeed present limitations to our study.

6.3. Construct validity

Construct validity is concerned with both the design of a study, as well as any social threats that may be present. This study was carefully designed, with the design being piloted and a detailed analysis of constructs such as knowledge and KM undertaken. The three models underpinning the research were also explained, establishing a connection with the objectives of this study. In addition, multiple case studies were completed to present multiple perspectives of KM in SE. The researcher conducted a number of interviews in two separate organisations, involving participants from different projects. Two social threats relevant to this study are hypothesis guessing and experimenter expectancies. With hypothesis guessing, a 'Hawthorne Effect' may transpire, introducing bias into the data collected. The researcher may also have particular expectations prior to commencing the study, either consciously or unconsciously. As a result, the researcher may interpret the data collected to satisfy these expected outcomes.

6.4. Conclusion validity

Conclusion validity assesses factors that affect the ability to draw correct conclusions about the relationship between treatment and outcome. Both question wording as well as the construction of a research instrument can influence the conclusions drawn from a study. From the pilot study, phrasing of questionnaire items was improved significantly. The questionnaire was divided into three main sections, with numerous subheadings outlining its logical structure to assist participants. The reliability of treatment implementation is also a significant notion to consider. In this study, care was taken to ensure that verbal questions asked in interviews were asked consistently across all interviews conducted. Other considerations are random irrelevancies in experimental settings and random heterogeneity of subjects. A standard environment was maintained for all interviews conducted, occurring in meeting rooms at the premises of each organisation. All participants were software developers, therefore limiting random heterogeneity in the group.

7. Conclusion and future work

The management of knowledge is regarded as a main source of competitive advantage for organisations [6,13]. To keep software companies competitive in the market, developing effective ways of managing software knowledge is of interest to software developers.

The objective this study was to explore the area of KM in SE, from an Australian perspective. The intention was to articulate current practices for KM in SE, as well as describing and modelling the KM activities and KM process used in SE practice. In addition, we examined enablers of the KM Process in SE as leadership, technology, measurement and culture, to establish an environment in which the KM process for SE could be applied. This study used the framework presented in Table 1 to investigate the above research objectives. Three concluding comments from our findings are the followings:

- The state-of-the-art practice for KM in SE was very immature in the companies that were investigated. Regular project feedback and discussion were encouraged while the ability to employ technology to search for and locate knowledge sources is a reality. A "learn by doing" approach emerged as the preferred way to create and acquire knowledge; while a link between KM improved business performance was made. We believe that while organisations accept the importance of KM, their attempts to address this issue are generally inconsistent and ad-hoc. We argue that in order to keep software organisations competitive in software business, especially in growing global software development environment, organisations need to follow more systematic approach managing their knowledge such as by focussing equally on both codification and personalisation strategies as suggested by several researchers [16,37]. Furthermore they need to take an approach that fits their organisational culture as well as specific business objectives and strategies.
- Software developers, in general, found it difficult to explain what they do, how they do something, how they solve particular problems using their knowledge. We also wanted to assess their understanding/perception of KM, which in turn, would determine their commitment to the practice of KM. It is argued that SE practitioners have a rudimentary understanding of activities in the KM process, hence complicating the ability to generate a descriptive model of the KM process used in SE. Overall, we conclude that while KM was considered to be important by participants, the tools, techniques and methodologies currently employed for software development were inadequate to address effective management of knowledge in these organisations. In both organisations, a uniform model of the KM process did not exist. However, participants were able to recognise KM activities being performed in their projects.
- Overall, leadership and technology emerged as the most significant enabler of the KM process for SE. For the leadership enabler, many participants believed it was ultimately their responsibility to ensure that the knowledge gained from project was preserved and made accessible to others. For the culture enabler, promoting knowledge sharing within organisations was considered crucial for software developers and the most useful feedback was seen the one provided during project rather than after the project completion. In regards to the technology enabler, participants believed that technology is useful in locating and transferring knowledge. Personal/informal networks, groupware and document management systems (Table 1) were the most frequently used enablers. As for the measurement enabler, it was recognised as being important, however, most participants encountered difficulty when asked to describe applicable measures that could be used to accurately measure personal knowledge.

- Finally, when we examine the relationship between KM activities and KM systems presented in Table 1, the finding showed that there was not always one-to-one mapping between KM activities and KM systems used in those activities. In other words, some KM systems were utilised in more than one KM activity. For instance personal networks could also be used not only during Knowledge Identification, acquisition and creation but also during knowledge transfer.

The need to deliver quality software has increasingly is still a subject of concern in software business. The management of knowledge and expertise has always provided the basis for systematic software development and process improvement. However, without a repeatable process for managing this knowledge and expertise, it can easily be dismissed as intangible. Authors of this article believe that KM provides a structured framework to assist software developers articulate the rules that they apply, the actions that they take and the social traditions of the domain that they operate within; which in turn triggered this investigation.

This study was designed to present an Australian perspective of KM in SE. We acknowledge that the findings from this study may not reflect the practices of organisations in Australian software industries, given the limited number of case studies examined. However, it provides a preliminary understanding for academia and software industry, on the aspect of 'supporting software development processes' from the KM perspective. This study also highlights the need for more investigations into the KM process and state-of-the-art practice for applying KM in SE, as means of identifying critical areas of concerns for practitioners. Furthermore, this study will benefit all stakeholders within the existing SE organisations that plan to embark in this new technology and cultural environments for their KM in SE processes.

It is hoped that a high-level descriptive process model for KM in SE will provide a foundation for the development of prescriptive process models for KM in SE. Future work needs to examine the KM process for SE in other industries and contexts, as well as the perspectives of other roles associated with software development. This research can also be extended to link KM more closely with software process improvement research. Further research could be carried out to determine how KM activities could be more closely integrated with software process improvement approaches such as CMMi and SPICE.

Finally, there is of course a great need for more studies in the field to generate more individual studies that in turn can be used in analysis of a series of case studies. In this way, a body of knowledge can be built that increases both academic and practitioners' general understanding of the impact of KM enablers in SE. Further studies are needed to investigate this issue.

Acknowledgement

We would like to thank Prof. Daniel M. Berry at University of Waterloo, Canada, for reviewing this paper and providing very useful comments during his visit to School of Information Systems, Technology and Management, University of New South Wales, Australia.

Appendix A. Knowledge Management activities

(a) *Knowledge Creation*. The generation of new knowledge in organisations.
(b) *Knowledge Acquisition*. Acquiring knowledge from either internal or external sources, including the process used to acquire this knowledge.
(c) *Knowledge Identification*. Determines what knowledge is applicable to a particular situation, whether knowledge items are reliable as well as finding and locating knowledge sources.
(d) *Knowledge Adaptation*. The modification of knowledge collected from previous experience and making it applicable to a specific issue or problem faced.
(e) *Knowledge Organisation*. Ensures that knowledge is stored and maintained for future reference by others.
(f) *Knowledge Distribution*. Focuses on sharing knowledge, either formally or informally.
(g) *Knowledge Application*. The use of knowledge for a particular purpose.

Appendix B. Knowledge Management systems

(a) *Communities of practice*. Groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis.
(b) *Personal networks*. A dynamic collection of interconnected heterogeneous personal devices, not only the local devices centered around the person, but also personal devices on remote locations such as devices in the home network, the office network and the car network.
(c) *Organisational practices and routines*. A set of dispersed knowledge in organisation that allow coordination and integration in volatile environments.
(d) *Document management systems*. Software that manages documents for electronic publishing. A document management system may support multiple versions of a document and may be able to combine text fragments written by different authors.
(d) *Expert systems*. Also known as Knowledge-based Expert Systems. These systems mimic the reasoning of experts whose knowledge is assumed to be deep in a narrow domain.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005

(e) *Training*. A planned process to help modify the attitude, knowledge or skill behavior of an individual through a learning experience.
(f) *Informal networks*. The web of relationships that people use to exchange resources and services. Informal networks are distinct from formal networks in that they are not officially recognised by organisations. The content of their exchanges can be work-related, personal, or social.
(g) *Groupware*. A software system that facilitates communication and collaboration between and among organisations.
(h) *Business Process Management (BPM) Systems*. A software tool intended to increase process efficiency by improving information flows between people as they perform business tasks.
(i) *New products and services technologies*. These are specifically system that support product life cycle with the aim of identifying the right time for new products and services.
(j) *New markets technologies*. Specialized knowledge-based system for identifying new markets both at company and industry levels.

References

[1] M. Alavi, D.E. Leidner, Knowledge management systems: issues, challenges and benefits, Communications of the Association for Information Systems (CAIS), vol. 1, Article 7, 1999.
[2] M. Alavi, D.E. Leidner, Knowledge management systems: emerging views and practices from the field, in: Proceedings of 32nd annual Hawaii international conference on system sciences, Maui, Hawaii, USA, 1999, 11 p.
[3] M. Alavi, D.E. Leidner, Knowledge management and knowledge management systems: conceptual foundations and research issues, MIS Quarterly 25 (1) (2001) 107–136.
[4] J. Andrade, J. Ares, R. García, S. Rodríguez, A. Silva, S. Suárez, Knowledge management systems development: a roadmap, Lecture Notes in Computer Science 2773 (2003) 1008–1015.
[5] A. Andersen, The Knowledge Management Tool. Developed jointly by Arthur Andersen & The American Productivity and Quality Center (APQC), Chicago, Arthur Andersen, 1997, 14 p. Available from: <http://www.apqc.org/km/>.
[6] A. Aurum, R. Jeffery, C. Wohlin, M. Handzic, Managing Software Engineering Knowledge, Springer, Germany, 2003.
[7] H. Baetjer Jr., Software as Capital: An Economic Perspective on Software Engineering. IEEE Computer Society, Los Alamitos CA, 1998.
[8] P.A. Balthazard, R.A. Cooke, Organisational culture and knowledge management success: assessing the behavior – performance continuum, in: Proceedings of the 37th Hawaii International Conference on System Sciences, 2004, pp. 97–64.
[9] V.R. Basili, P. Costa, M. Lindvall, M. Mendonca, C. Seaman, An experience management system for a software engineering research organisation, in: Proceedings of 26th Annual NASA Goddard Software Engineering Workshop, 2001, pp. 29–35.
[10] V.R. Basili, M. Lindvall, P. Costa, Implementing the experience factory concepts as a set of experience bases, in: Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering, Buenos Aires, Argentina, 2001, pp. 102–109.
[11] V.R. Basili, G. Caldieram, H.D. Rombach, Experience factory, in: J.J. Marciniack (Ed.), Encyclopedia of Software Engineering, John Wiley and Sons, Hoboken, NJ, USA, 1994.
[12] W.R. Borg, J.P. Gall, M.D. Gall, Applying Educational Research: a Practical Guide, Longman Publishing Group, New York, USA, 1993.
[13] T.H. Davenport, L. Prusak, How Organizations Manage What They Know, Harvard Business School Press, Boston, Massachusetts, 1997.
[14] M. de Jager, The KMAT: benchmarking knowledge management, Library Management 20 (7) (1999) 367–372.
[15] K.C. Desouza, Barriers to effective use of knowledge management systems in software engineering, Communications of the ACM 46 (1) (2003) 99–101.
[16] K.C. Desouza, Y. Awazu, P. Baloh, Managing knowledge in global software development efforts: issues and practices, IEEE Software 25 (5) (2006) 30–37.
[17] A.J. DiBella, E. Nevis, How Organizations Learn, Jossey Boss, San Francisco, 1998.
[18] T. Dingsøyr, N. Brede Moe, O. Nytro, Augmenting experience reports with lightweight postmortem reviews, Lecture Notes in Computer Science 2188 (2001) 167–181.
[19] T. Dingsøyr, H.K. Djarraya, E. Royrvik, Practical knowledge management tool use in a software consulting company, Communications of the ACM 48 (12) (2005) 97–103.
[20] G. Disterer, Management of project knowledge and experiences, Journal of Knowledge Management 6 (5) (2002) 512–520.
[21] T. Dybå, An empirical investigation of the key factors for success in software process improvement, IEEE Transactions on Software Engineering 31 (5) (2005) 410–424.
[22] M.T. Hansen, N. Nohria, T. Tierney, What's your strategy for managing knowledge? Harvard Business Review, March–April (1999) 106–116.
[23] T. Hellström, U. Malmquist, J. Mikaelsson, Decentralizing knowledge: managing knowledge work in a software engineering firm, Journal of High Technology Management Research 12 (3) (2000) 25–38.
[24] C.W. Holsapple, K.D. Joshi, Description and analysis of existing knowledge management frameworks, in: Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999, 15 p.
[25] J. Jashapara, Knowledge Management: an Integrated Approach FT (Financial Times), Prentice Hall, Pearson Education Limites, Essex, England, 2004.
[26] C. Johansson, P. Hall, M. Coquard, Talk to Paula and Peter – they are experienced, in: Proceedings of the Workshop of Learning Software Organizations, Kaiserslautern, Germany, 1999, pp. 69–76.
[27] P. Klint, C. Verhoef, Enabling the creation of knowledge about software assets, Data and Knowledge Engineering 41 (2–3) (2002) 141–158.
[28] R. Kneuper, Supporting software processes using knowledge management, in: S.K. Chang (Ed.), Handbook of Software Engineering and Knowledge Engineering, vol. II, May 2002, World Scientific Publishing, 2002. (ISBN: 981-02-4974-8), 2002, S579–608.
[29] D. Ko, A.R. Dennis, Who Profits from Knowledge Management? A Case of Experience versus Expertise. Hawaii International Conference on System Sciences, Hawaii, January 2004.
[30] S. Komi-Sirvio, A. Mantyniemi, V. Seppanen, Towards a practical solution for capturing knowledge for software projects, IEEE Software 19 (3) (2002) 60–62.
[31] K.C. Laudon, J.P. Laudon, Managing Information Systems: Managing the Digital Firm, Pearson/Prentice Hall, Upper Saddle, NJ, USA, 2006.
[32] P.D. Leedy, J.E. Ormrod, Practical Research – Planning and Design, 8th edition., Pearson Merrill Prentice Hall, New Jersey, 2005.
[33] H. Lee, B. Choi, Knowledge management enablers, processes, and organizational performance: an integrative view and empirical examination, Journal of Management Information Systems 20 (1) (2003) 179–228.
[34] H.C. Lucas Jr., Information Technology and the Productivity Paradox: Assessing the Value of the Investment in IT, Oxford University Press, New York, USA, 1999.
[35] M.L. Markus, Toward a theory of knowledge reuse: types of knowledge reuse situations and factors reuse success, Journal of Management Information Systems 18 (1) (2001) 57–94.

[36] L. Mathiassen, P. Pourkomeylian, Managing knowledge in a software organisation, Journal of Knowledge Management 7 (2) (2003) 63–80.
[37] S. Mukherji, Knowledge management strategy in software services organisations: straddling codification and personalisation, IIMB Management Review 17 (3) (2005) 33–41.
[38] I. Nonaka, A dynamic theory of organizational knowledge creation, Organization Science 5 (1994) 14–37.
[39] I. Nonaka, H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, New York, USA, 1995.
[40] M. Polyani, The Tacit Dimension, Routledge & Kegan Paul, London, UK, 1966.
[41] S.P. Raub, B. Sthapit, Towards a taxonomy of approaches for measuring organizational knowledge, Research and Practice in Human Resource Management 9 (1) (2001) 139–155.
[42] P.N. Robillard, The role of knowledge in software development, Communications of the ACM 42 (1) (1999) 87–94.
[43] T. Rout, Aspects of Process Assessment and Improvement, 6th International SPICE Conference on Software Process Improvement and Capability dEtermination (SPICE'2006), May 3rd–5th, Luxembourg, 2006.
[44] R. Ruggles, The state of the notion: knowledge management in practice, California Management Review 40 (3) (1998) 80–89.
[45] I. Rus, M. Lindvall, Knowledge management in software engineering, IEEE Software 19 (3) (2002) 26–38.
[46] E. Sadrei, A. Aurum, G. Beydoun, B. Paech, A Field Study of the Requirements Engineering Practice in Australian Software Industry, International Journal Requirements Engineering Journal 12 (2007) 145–162.
[47] K. Schneider, J.-P. von Hunnius, V. Basili, Experience in implementing a learning software organization, Computer 19 (3) (2002) 46–49.
[48] T. Stålhane, T. Dingsøyr, G.K. Hanssen, N.B. Moe, Post mortem – an assessment of two approaches, Lecture Notes in Computer Science 2765 (2003) 129–141.
[49] A. Tiwana, An empirical study of the effect of knowledge integration on software development performance, Information and Software Technology 46 (13) (2004) 899–906.
[50] G. von Krogh, Care in knowledge creation, California Management Review (40:3) (1998) 133–153.
[51] G. von Krogh, A. Back, A. Seufert, Putting Knowledge Networks into Action, Springer, Berlin, 2005.
[52] J. Ward, A. Aurum, Knowledge Management in Software Engineering – Describing the Process, in: Proceedings of Australian Software Engineering Conference, ASWEC'04, 13–16 April, Melbourne, Australia, 2004, pp.137–146.
[53] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering – An Introduction, Kluwer Academic Publishers., London, UK, 2000.
[54] N. Wickramasinghe, Integrating e-commerce and knowledge management – what does the Kaiser experience really tell us? International Journal of Accounting Information Systems 3 (2) (2002) 83–98.

Please cite this article in press as: A. Aurum et al., Investigating Knowledge Management practices in ..., Inform. Softw. Technol. (2007), doi:10.1016/j.infsof.2007.05.005