

TOWARDS MANAGING KNOWLEDGE FOR SOFTWARE PROCESS IMPROVEMENT

Bridget Meehan¹, Ita Richardson,²

Abstract. Can explicit and effective knowledge management be used as part of current initiatives to improve software processes in software companies? Is effective and explicit knowledge management the key to pushing companies to higher levels of software process maturity? The authors present research which takes the first step in exploring these possibilities.

AMS Mathematics Subject Classification (1991):

Key words and phrases: knowledge management, software process, software process improvement (SPI)

1. Background and rationale

Knowledge is fast becoming the key to survival and competitive advantage in the dynamic and increasingly unpredictable global markets of today [11, 14, 5]. This might sound like a cliché, but it is still a fact.

The knowledge that belongs to a company is known as its corporate knowledge. Corporate knowledge is the *explicit* and *tacit* knowledge a company has about processes, procedures, markets, products, and technologies, that enables it to generate profits and be competitive [5].

Skyrme [13] describes knowledge management as “the explicit and systematic management of vital corporate knowledge...it requires turning personal knowledge into corporate knowledge that can be widely shared throughout a company, and appropriately applied”.

Knowledge management is about: treating knowledge like any other corporate asset and managing it as such; promoting learning and innovation; nurturing collaboration, not competition, between employees; safe-guarding against the loss felt when an employee leaves; preventing repetition of work [16, 1, 5]. And ultimately, knowledge management is about improving the bottom line. To do all this, a company needs to explicitly and effectively manage certain

¹School of Computing and Mathematical Sciences, University of Ulster at Magee College, Northland Road, Derry BT48 7JL, Northern Ireland. Email: b.meehan@ulst.ac.uk

²Department of Computing Science and Information Systems, University of Limerick, Castletroy, Limerick, Ireland. Email: ita.richardson@ul.ie

knowledge-related processes. These processes are knowledge creation, knowledge storing and gathering, knowledge use, knowledge leverage, and knowledge sharing [15].

This research explores the potential of knowledge management within the context of software processes and their improvement. Software process improvement (SPI) is based on the principle that improving the software processes by which a product is developed (i.e. making processes more effective and efficient), leads to sustainable improvement in competitive factors such as quality, productivity, and time-to-market [3]. There are a number of SPI models available, such as the Capability Maturity Model (CMM), BootStrap, the Software Process Improvement and Capability dEtermination framework (SPICE), and the Software Process Improvement in Regions of Europe project (SPIRE), which can be used to guide companies in their SPI efforts. These models generally determine the effectiveness and efficiency of processes by measuring the level of their maturity. The more mature a process is, the more effective and efficient it is.

However, problems can be experienced when implementing current SPI models. For example, companies often experience quite a bit of difficulty in reaching the highest levels of process maturity. There are some examples of organisations that have reached the highest level, level 5, of the CMM [10], but these are relatively few. In addition, business environments, including software development environments, are constantly changing and unpredictable [6]. Ould [9] points out that change is the norm and stability the exception in today's software development environments. Therefore, SPI initiatives should focus on the ability to change over the ability to remain stable, which is not the case with most initiatives.

It is these problems which have turned the attention of the authors to the role of knowledge management in SPI initiatives. For example, it is effective knowledge management which is the key to coping with constantly changing and unpredictable environments [11]. So perhaps managing knowledge can overcome the problem of managing change in software development environments.

Although there are many parallels between knowledge management and SPI models, the authors believe that knowledge management covers a much wider spectrum of issues: tackling the issue of managing valuable tacit knowledge, and not just explicit knowledge; performing external research (i.e. an activity where employees pro-actively look outside their company for new technologies, practices, and ideas which can be used to improve processes within the company, and keep them up-to-date with current trends); using explicit coaching or mentoring systems which ensure that new or inexperienced employees learn about how processes are performed and that they learn how to perform these processes correctly; promoting formal and informal collaboration and communication between employees.

The rationale behind the research presented here, is that by addressing the above issues and thus managing knowledge more effectively and efficiently within

current SPI initiatives, greater benefits could be get from the initiatives and higher levels of software process maturity could be attained. The research does not go to the point of proving that this is the case. However, it serves as an important first step in this direction by investigating how knowledge may be currently managed in software processes. To take this first step, the following question is answered:

- How is knowledge being implicitly managed within software development processes in software companies? The word 'implicit' is deliberate here because the authors were not going to study companies with explicit or formal knowledge management initiatives in place. However, although not explicitly managed, the authors were of the belief that knowledge would at least be implicitly managed in some way.

The remainder of this article discusses how the authors went about answering this question, and discusses the results of the investigation.

2. Research method

To discover how knowledge is implicitly managed in the software development processes of software companies, the following six steps were followed:

1. **Implicit knowledge management was defined** for the purposes of the research. The definition developed by the authors was:
 - The absence of an explicit knowledge management initiative, and the extent to which any of the knowledge-related processes of creation, storing, sharing, and leverage are explicitly or implicitly performed.

This definition was based on the major knowledge-related processes listed in the literature [15] which must be performed if knowledge is to be managed. Each of the knowledge-related processes listed in the definition has also been defined individually by the authors. These definitions are given in the recommendations section and are described in terms of how they relate specifically to software development processes.

2. **Three small, Irish software companies were chosen.** A small company can be defined as an independently owned firm employing fifty employees or less [4]. The three companies chosen for the research were ComTech Inc., LogiBytes Ltd., and DesignIT Solutions Ltd.

ComTech Inc. is a software development company with fifty employees. It was established in 1992 to develop software for a very specific market. LogiBytes Ltd. is a software house with fifty-two employees. Established in 1994 it is a dynamic, leading edge organisation, developing software for various markets, particularly the banking and financial sectors. Established in 1992, DesignIT

Solutions Ltd. has thirty employees. The company is a multi-media design agency, specialising in the formulation of new media strategies. Many forms of new media are explored, from Web technology to CD-ROM.

3. **Two software development processes were chosen** for the study: Software Requirements Management and Software Design.
4. **Employees where chosen from each company to help with the investigation.** Employees were chosen on the basis of how long they had been with the company and how much software development experience they had. They needed to be employees in their company for at least three years, and needed to have experience of performing the Software Requirements Management process, and the Software Design process.
5. **The chosen employees from the chosen companies were interviewed.** Nine interviews were conducted in total, three for each company:
 - Interview One in each company gathered general knowledge-related data about the company.
 - Interview Two in each company gathered knowledge-related data about the Software Requirements Management process.
 - Interview Three in each company gathered knowledge-related data about the Software Design process.

Interviews were designed to be as open-ended as possible, and used a combination of the unstructured interview technique and interview guidelines [7]. Being unstructured, interviews allowed candidates to discuss issues at length. Interview guidelines covered all the questions the authors needed to ask, and were used more as a reference to help the authors keep interviews on track.

Questions were constructed according to Patton's question typology, and therefore were singular and avoided yes or no answers, encouraging longer, more elaborate answers. Care was also taken to avoid leading questions. The questions focused on how the software requirements management and software design processes were performed in the companies, and on the knowledge in these processes. At no time was knowledge management explicitly mentioned. Examples of questions that were asked include:

- What public documents are available about possible problems with processes and their solutions?
- In what way do you get the opportunity to pass on what you know about the process with others?
- When problems occur in the process, in what way are you expected to solve them?
- What in your opinion could be done to improve the process?

6. **The data gathered during the interviews were analysed** using the constant comparative method [7]. This method is non-mathematical and analytical, examining the meanings of people's words and actions. It describes and interprets the data. It was felt that such a method suited the needs of the research.

The constant comparative method involves categorising the data, once it they have collected. The authors identified four categories within the data, categories that were consistent with knowledge management activities. The categories identified were those of knowledge creation, knowledge storing, knowledge sharing, and knowledge leverage. Definitions for these categories were developed by the authors to specify exactly what they meant in the context of software development processes, and were an outcome of the findings from the research. The definitions for these categories are given in the recommendations section of this paper.

3. Recommendations

The data gathered from the interviews with the three companies were analysed according to the knowledge-related categories of knowledge creation, knowledge storing, knowledge sharing, and knowledge leverage (as mentioned in the research method section). Recommendations for more effective and more explicit knowledge management are shared in this section for two of the four knowledge-related categories identified by the authors (i.e. knowledge creation and knowledge sharing). The recommendations are a result of the findings from the data.

Throughout this section, the word *processes* will be used when discussing both of the processes studied in the research, namely the Software Requirements Management process and the Software Design process. In addition, the Software Design process will be referred to as the *design process*, and the Software Requirements Management process will be referred to as the *requirements process*.

3.1. Knowledge Creation

3.1.1 Introduction

Knowledge creation within software processes happens when employees learn how processes are performed, tacitly or explicitly, from other employees and from appropriate documentation, if it is available. Knowledge creation also happens when employees learn through research they have performed themselves, for example external research, as described previously in this paper.

3.1.2 Recommendations for explicit, effective Knowledge Creation

In the companies studied, formal training on the processes, in the form of training courses, does not seem to be necessary. The consensus is that the

processes have to be learned on-the-job. However, it is the approach to this on-the-job learning that determines whether it works or not. Consider the following approach:

- Throwing inexperienced employees in at the *deep-end*, and not providing them with any support or guidance from more experienced employees.
- A *lack of documentation* to which employees can refer on how the processes should be performed. For example, process descriptions or examples of how the processes were performed in the past.

When this approach to on-the-job learning is followed it means that inexperienced employees have nothing to learn from, and have no way of knowing how the processes should be performed. There are no guidelines on how the processes should be performed, no one to help them through the processes. When this happens, employees have no way of knowing if they are performing the processes correctly or consistently with other employees in the company. To be effective, on-the-job learning should be approached in the following way:

- New or inexperienced employees must be supported by experienced employees through the use of a *mentoring system*, however informal. Through such a system, new or inexperienced employees are assigned to more experienced employees whom they can observe at work, and from whom they can learn. Mentors should give 'their' employees real, but less complex work to do at first, to help them build up some confidence when performing the processes. The notion of mentoring is one of the patterns for creating knowledge, according to Nonaka [8], but it is not an explicit part of any SPI model.

If a mentoring system is to work, management should actively encourage experienced employees to give support and guidance to inexperienced employees. Management should allocate a certain amount of time to the more experienced employees for support and guidance activities. Inexperienced employees should be made feel comfortable enough to ask questions and look for help if they are unsure of certain issues.

- A certain amount of *process documentation* may be made available. This is useful, but not vital. The authors recommend that there at least be descriptions of the processes, which outline the steps involved in them.

In addition, documentation on how the processes were performed in the past can prove to be useful. Such documentation provides inexperienced employees with concrete examples of how the processes have been performed.

The authors believe however, that documentation alone is not enough for employees to learn about a process. For example, a process description of a process does not ensure that the process will be performed consistently or correctly, according to the documented description. If documents are disorganised, verbose, or badly written, the tendency is not to read them at all.

Even good documentation can be ignored. There should be interactive, online documents if possible (note: this point was further explored in the knowledge storing category which is not included in this paper).

An additional observation made by the authors as regards process learning, is that there should be a conscious effort on the part of management to get new and inexperienced employees involved in the processes, as soon as possible. So for example, the mentor of an inexperienced employee who has an adequate knowledge of programming, could gradually involve the employee in the design process. Or the mentors of more experienced designers could gradually involve them in the requirements process. In this way, more and more employees will be able to perform the processes and a company's dependency on certain key employees can be reduced.

Performing external research is another important aspect of knowledge creation. External research will not happen if employees who perform it are given no recognition or encouragement from management. It will not happen if management expect employees to perform external research only in their own time and on their own initiative. And even if encouragement is given, external research will not happen if management do not specify exactly how it should be done. To ensure that external research is a regular, routine activity performed by all employees:

- Management should give *explicit encouragement* to employees to perform external research, and should give *clear guidance* on how external research is to be performed. Employees should be made feel that they really are *part of the job* and not something that might be nice to do, if they could get time. For example, to make external research part of the job, it could be included as part of an employee's job description. This results in a very explicit statement from management which says that external research is expected to be performed and that time must be made for it. To re-enforce the need for external research, management must reward employees who make it part of how they work, for example by incorporating external research into performance appraisals and bonus schemes.
- Employees should be encouraged to *buy and read* books and magazines they feel are relevant. They should be encouraged to *look up* electronic-bulletin boards, electronic-forums, and web sites that might be useful.

3.2. Knowledge Sharing

3.2.1 Introduction

Knowledge sharing happens when employees pass on process knowledge to each other, in order to gather knowledge about the processes, to avoid repeating past mistakes, to reduce dependency on a few key employees, and to improve

decision-making. Knowledge can be shared informally – through daily communication and collaboration with other employees, and good teamwork – and formally – through project and cross-project meetings. Knowledge can also be shared by making all stored knowledge widely available to all employees.

3.2.2 Recommendations for explicit, effective Knowledge Sharing

The more process knowledge that is shared among employees, the more able those employees are to contribute to the processes and to perform them with confidence. The authors believe that process knowledge must be explicitly shared both formally and informally:

- Formal knowledge sharing is facilitated primarily through *meetings*. At meetings, knowledge is shared through the use of examples, rich narratives, and ‘war stories’ [2, 17]. The context of what is being shared is not lost, because employees relate the whole story around their experiences, and not just specific parts of interest taken out of context. At meetings, knowledge about external research, solutions to problems, tips and features found during the course of the processes, past experiences and past mistakes can be shared, and all employees can find out about what others have done. These meetings should help employees to learn from previous mistakes and ensure that the same ones are not repeated over and over again.

Meetings should take the form of *team meetings* within individual projects, and *cross-project meetings* which are company-wide. At cross-project meetings employees are able to find out what other projects have experienced with the processes, and not just what their individual projects have experienced.

Management must organise and co-ordinate meetings and instill in employees that they are important. They must educate employees in the need for the meetings and make explicit what the purpose and aims of these meetings are. In this way, meetings will be more effective and should contribute more to employee knowledge. Lack of management encouragement for meetings, especially cross-project meetings, can usually mean that they will not happen. Or even worse, they happen, but are not effective because employees do not really understand why they are being conducted.

It is important to stress that meetings provide a platform for sharing tacit knowledge. Explicit knowledge can be stored in documents and thus shared quite easily. But, it is difficult to share tacit knowledge in this way. However purposeful and organised meetings with their use of ‘war stories’ and so on, can help make explicit the tacit knowledge that is so difficult to store by ensuring that even the insights, experience, and judgements stored in employees’ heads can be shared. For example, employees can be told about issues that cannot be written down simply because these issues need to be demonstrated or put into context.

- Informal knowledge sharing is facilitated through good **teamwork**, and good **communication and collaboration** between employees on an every day basis.

Management must encourage employees to work, not as single units, but as unified teams, where the overall performance of a project team is judged and not just the performance of individual employees. In other words, if a project team has performed badly, then everybody on it has performed badly. It is not enough for one member to perform their particular area well. If other areas have done badly, that is a reflection on even those employees who have personally performed well.

To re-enforce the **knowledge sharing environment**, management should be explicit in creating an open, honest environment where knowledge sharing – both formal and informal – is power. They should allow employees to make time for sharing. Management should recognise and reward employees who share, by using the extent of their sharing as one of the criteria considered for bonuses and promotions [12].

The more process knowledge that is shared, both formally and informally, with employees, the more everyone will know about the process. Therefore, important insights to and information about the processes are not retained by particular employees, but are openly available to all employees. In this way, explicit knowledge sharing can help reduce a company's **dependency on a few key employees**. Additionally, **decision-making** is not left to just a few key individuals, who can become overloaded and suffer from additional pressure and stress as a result. Instead, decision-making can be shared among the larger number of employees who have knowledge of the processes.

4. Conclusion

The research presented here took a first step in investigating how SPI initiatives might benefit from effective and explicit knowledge management. This first step was taken by investigating what, if any, knowledge management activities were present in software processes and by investigating how these activities were being performed.

The findings of the investigation established that four knowledge management activities – knowledge creation, knowledge storing, knowledge sharing, and knowledge leverage – existed in the software processes studied, to varying degrees. The investigation also established how these activities were being performed. From the findings, the authors were able to make recommendations as to how these knowledge management activities could be performed more explicitly and more effectively for the purposes of software process improvement.

For the authors, the next step is to incorporate the recommendations made into a knowledge management framework. The framework will give software

companies practical advice and guidelines on organising current knowledge creation, storing, sharing, and leverage activities better, and how they can make them more effective and more explicit. The authors believe that the use of the knowledge management framework will bring about improvements in software processes. The framework is not proposed as an alternative to, or a replacement of, any SPI model. If anything, the framework when implemented, should complement and enhance any SPI model that might already be in place in software companies.

Once developed, the authors will pilot the knowledge management framework in selected software companies, monitoring its performance to determine its impact on software processes. What is certain at this point, is that knowledge management does indeed figure in software processes, and the authors believe it has a valid role to play in their improvement. But, it will only be by piloting the framework in software companies that the authors will determine for certain how significant that role is, and if knowledge management can indeed help improve software processes.

References

- [1] Ackerman, A.S., *Augmenting the Organisational Memory: A Field Study of Answer Garden*. URL <http://www.ics.uci.edu/CORPS/ackerman.html>, 1994.
- [2] Bannon, L., Kuutti, K., *Shifting Perspectives on Organisational Memory: From Storage to Active Remembering*. IEEE HICSS'96, Hawaii, 1996.
- [3] European Software Institute. URL <http://www/esi/es>, 1997.
- [4] Department of Enterprise and Employment, *Small Business and Services Structural Funds*. Dublin: Stationary Office, 1998.
- [5] Macintosh, A., *Position Paper on Knowledge Asset Management*. URL <http://www.ntgi.net/ntgi/yk2/info/kmfr.htm>, 1997.
- [6] Malhotra, Y., *How to Facilitate Knowledge Sharing*. URL <http://www.brint.com/papers/compint.htm>, 1993.
- [7] Maykut, P., Morehouse, R., *Beginning Qualitative Research: A Philosophic and Practical Guide*. London: Falmer Press, 1995.
- [8] Nonaka, I., *The Knowledge Creating Company*. Harvard Business Review, November-December, 1991.
- [9] Ould, M.A., *The CMM and ISO 9001. Software Process - Improvement and Practice, Volume 2*, 1996.
- [10] Paulk, M.C., *List of Maturity Levels 4 and 5 Organisations*. URL <http://www.sei.cmu.edu/activities/cmm/high-maturity/HighMatOrgs.html>, 2000.
- [11] Prusak, L. (ed.), *Knowledge in Organisations*. Butterworth-Heinemann, 1997.
- [12] Quinn, B.J., Anderson P., Finkelstein S., *Managing Professional Intellect: Making the Most of the Best*. Harvard Business Review, March-April, 1996.

- [13] Skyrme, D., Management Insight No.2 Knowledge Management: Making Sense of an Oxymoron. URL <http://www.skyrme.com/insights.htm>, 1996a.
- [14] Skyrme, D., Management Insight No.1 The Global Knowledge Economy: And Its Implications for Business. URL <http://www.skyrme.com/insights.htm>, 1996b.
- [15] Wiig, K.M., Position Statement on the Management of Knowledge. URL <http://www.ntgi.net/ntg/yk2/info/kmfr.htm>, 1996.
- [16] Young, R., Accelerated Knowledge Transfer. URL <http://www.knowledgeassociates.com/website/KMJournal.nsf>, 1996.
- [17] Zack, M.H., Reisman, J.G., Serino, M., Knowledge Management and Collaboration Technologies. URL <http://www.lotus.com>, 1996.