Can Managing Knowledge and Experience Improve Software Process? - Insights from the literature

Neeraj Sharma[1], Kawaljeet Singh[2], D.P. Goyal[3]
[1] Associate Professor, Department of Computer Science
[2] Director, University Computer Centre
Punjabi University, Patiala - 147002
[3] Professor, Management Development Institute,
Gurgaon - 122007
sharma_neeraj@hotmail.com, director@pbi.ac.in, dpgoyal@mdi.ac.in

Abstract: *The field of knowledge management has gained wide attention across all industry sectors including software development. Managing accumulated knowledge and experiences of members of software engineering community is seen as a silver bullet to end many of the classical ills associated with the art of software development. Knowledge and experience management is also advocated in improving software development process. The present paper investigates the role of knowledge and experience management in improving the software process.*

Keywords:*Knowledge Management, Experience Management, Software Process Improvement, Experience Factory*

Introduction

There is no denying the fact that accumulating and managing software development experiences play a very significant role in improving the software process. Reusing experience in the form of processes, products, and other forms of knowledge is essential for improvement, that is, reuse of knowledge is the basis of improvement (Basili et al., 1994). The fundamental idea is to improve the software process on the basis of the accumulated knowledge and experiences of the software engineers working in the organisation. SPI efforts depend on the implicit, individual knowledge of practitioners in an organisation as an organisation's software development practices are ultimately based on the knowledge and competencies of its software developers. Changing software developers' practices in an organisation necessitates that the organisation finds ways to improve the practitioners' existing knowledge (both theoretical and practical) of its software practices. In other words, knowledge about the new processes should be made available across different organisational levels. A major challenge for organisations planning for software process improvement (SPI) initiatives is hence to create strategies and mechanisms for managing knowledge about software development. Insights from the field of knowledge and experience management are therefore potentially useful in SPI efforts so as to facilitate the creation, modification, and sharing of software processes in an organisation.

Present Paper

The present paper aims to explore the role of knowledge and accumulated experiences of the members of the software engineering (SE) community in improving software development process. Research methodology used is an exhaustive literature survey from the domains of knowledge management, experience management, software engineering and software process improvement.

Role of KM/ EM in SE

Knowledge management (KM) in software engineering is somewhat distanced from mainstream KM and the reason cited for this absence of software engineering in the wider KM literature is a tendency for discussion of such topics to take place only at conferences for the software engineering community (Edwards, 2003). Overview of works published on knowledge management in the area of software engineering has been found in the exhaustive survey of literature. Rus et al. (2001) give an overview of KM in software engineering, focusing on motivations for KM, approaches to KM and factors that are important when implementing KM strategies in software engineering organisations. Ward and Aurum (2004) posit that the management of knowledge and experience are key means by which systematic software development and process improvement occur. They further assert that although remedies such as fourth generation programming languages, structured techniques and object-oriented technology, software agents, component-based development, agile software practices, and so on have been promoted; a silver bullet has yet to be found. Lindvall et al. (2001) provide a detailed account of types of software tools relevant for knowledge management, ranging from document and content management tools to collaboration tools and tools for competence management. Further, a description of KMS in companies is found in literature, e.g., Conradi and Dingsoyr (2000). Rus and Lindvall (2002) maintain that KM compliments existing SPI methods and is not a replacement to any of these approaches and suggest that both formal and informal knowledge sharing between software developers must be supported and facilitated by organisations. A number of options for implementing and using KM systems for SE are also advanced, such as expert identification, the creation of KM champions, document management and using predictive modeling to direct decision-making. Dingsoyr and Conradi (2002) surveyed the literature for studies of KM initiatives in software engineering and found eight lessons learned reports, which are characterized after what actions companies took, what the effects of the actions were, what benefits are reported and what kind of KM strategies were used. They conclude that if we look at the work on the actual use of KM in an organisation, we find much less in the literature.

Different aspects of experience management (EM) have been covered by many researchers in the field though research in the EM domain in software engineering is varied. Basili et al. (1994) and Schneider (2000) discuss the approaches to EM like the classical EF approach or a more recent Light-weight approach to EM. Methods to collect experience in software engineering environments have been debated by many researchers, e.g.

Althoff et al. (1999a) and Land et al. (2001) whereas ways to structure it have also been elaborated by many, e.g. Houdek et al. (1998) and Lindvall et al. (2001). There are studies covering various tools for managing experience in software engineering, e.g. Henninger and Schlabach (2001), Basili et al. (2002), and Mendonca et al. (2001). A generic scalable architecture along with a methodology for reusing SE experience has been suggested by Althoff et al. (1999a). They hold that an organisational memory can support continuous reuse of software engineering experience. Literature describing the results of the applications of EM in practice are also found, e.g. Mendonca et al. (2001), Brossler (1999), Dingsoyr and Royrvik (2003), Diaz and Sligo (1997).

KM/EM Approaches in SE

In software engineering, *Experience Factory* (EF) approach proposed by Basili et al. (1994) is the most popular solution for capturing and managing the SE knowledge into knowledge repositories, called the Experience Bases. The Experience Factory is a logical and/or physical organisation that supports project developments by analyzing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand (cf. Figure 1). Literature is abound with many successful stories of EF implementations in many organisations, e.g. Houdek et al. (1998), Henninger (1997) and Koennecker et al. (2000). Furthermore, experience factories covering different aspects of software engineering process are reported. Althoff et al. (1999b) explain the software development process and experimental software engineering using the EF concept. We also find the mention of domain-specific experience factories which include software development cost estimation (Finnie et al., 1997), data mining applications (Bartlmae, 1999), developing CBR applications (Althoff et al., 1999c; Bergmann et al., 1999), and ontology deployment (Kalfoglou and Robertson, 2000). Software engineering knowledge representation and technology to support the development of EB have been the main focus of research in this area. How to document and represent experiences in EB finds special emphasis in research.

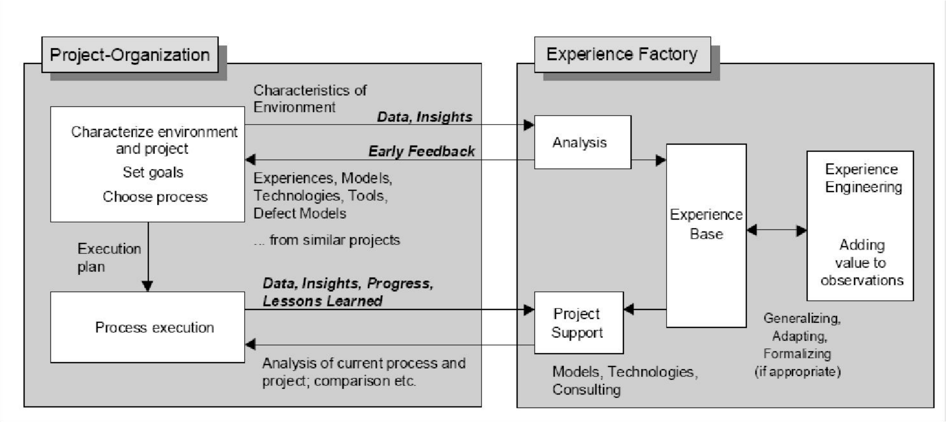
Project-Organization
Experience Factory
Characteristics of Environment
Data, Insights
Characterize environment and project
Set goals
Choose process
Early Feedback
Experiences, Models, Technologies, Tools, Defect Models
... from similar projects
Execution plan
Data, Insights, Progress, Lessons Learned
Process execution
Analysis of current process and project; comparison etc.
Analysis
Experience Base
Experience Engineering
Adding value to observations
Project Support
Generalizing, Adapting, Formalizing (if appropriate)
Models, Technologies, Consulting

Figure 1: Experience Factory Organisation

However, we also find studies which forewarn about the potential barriers in implementing EF approach in SE organisations (Basili et al., 1994). They conclude that EF implementations require a significant investment of time and efforts. Need to leverage alternate approaches to distribute knowledge have been stressed. A short term solution to this problem has been suggested in the form of ‘Answer Garden’ approach. An approach called the ‘Knowledge Dust to Pearls’ has been explained by Basili et al. (2001). This approach addresses some of the issues with KM in software development and allows the experience-base to become more useful more quickly than traditional approaches. There are also descriptions of EM techniques for managing SE knowledge as an alternative to the EF approach, e.g. Johansson et al. (1999) apply an ‘Experience Engine’ approach. Project review technique has been suggested by Kess and Haapasalo (2002) to improve software quality. Another popular technique, called the Case Based Reasoning (CBR), is recommended by Tautz and Althoff (1997) for a large EB.

Knowledge Acquisition in SE

Birk et al. (1999, 2002) give an overview of the various knowledge acquisition techniques in a software engineering setting wherein they advise a methodology for developing customised knowledge acquisition methods. These techniques are also referred to as 'knowledge harvesting' techniques in literature. Dingsoyr et al. (2001) present two common techniques for harvesting knowledge or externalising tacit knowledge from projects: Experience reports and Postmortem reviews. Experience report is a single-author document, usually written by the project manager to sum up the experience(s) gained from the finished project. These reports have fixed templates so as to make comparisons across the projects easier. Apart from containing usual project data viz. project start and finish dates, labour used, documents produced etc., it also includes problems encountered during the project and the proposals for future improvements. Postmortem reviews or Postmortem analysis (PMA) or project retrospectives have been suggested as another method of acquiring experience in previous software projects and improving it in future projects.

Dyba et al. (2004) render the project-based experiential learning model which suggests different kinds of reviews, called workshops, to be held before, during and after the project. The workshop before the project initiation is held with a goal to explain the purpose of the project to the project members. Workshops during the project execution serve the purpose of on-project feedback and provide an opportunity for short-term improvements. The after-project workshop makes the experiences of the project explicit and reusable in future projects. A description of another lightweight approach to PMA is also found in literature. This technique tries to seek project experience using personal interviews of the project team members (Schneider, 2000). Table 2.5 highlights the summary of literature on the role of KM/ EM in the field of software engineering.

Table 2.5: Summary of Literature on Role of KM/ EM in SE and Gap Analysis

S.No	Topics	Sub-topics Covered	References	Gap
1	KM in SE	Overview of KM in SE	Rus et al. (2001); Dingsoyr and Conradi (2002); Kess and Haapasalo (2002); Seleim and Kalil (2007)	None is integrated to SE environment.
		Types of software tools relevant for KM	Lindvall et al. (2001)	
		KMS in organisations	Conradi and Dingsoyr (2000)	
		KM in SPI	Rus and Lindvall (2002)	
		KM architecture	Borges and Falbo (2002); Markkula (1999); Althoff et al. (1999)	
2	EM in SE	Experience Factory approach	Basili et al. (1994)	Little research exists on the detailed use and structure of successful experience repositories.
		Light-weight approach	Schneider (2000)	
		How to collect experience	Althoff (1999a); Land (2001)	Lack of experimentation in using KM insights in SPI.
		How to structure experience	Houdek et al. (1998); Lindvall (2001)	
		Tools for managing experience	Henninger (2001); Basili (2002); Mendonca et al. (2001)	
		Application of experience management in practice	Brossler (1999); Dingsoyr and Royrvik (2003); Diaz and Sligo (1997)	
3	Experience Factory Approach	Principles and the working of the EF approach	Basili et al. (1994); Althoff et al. (2000); Houdek et al. (1998)	Limitations of time and investment. Experimental nature. It requires Organisational restructuring.
		Domain-specific experience factories	Cost estimation (Finnie et al., 1997); Data mining applications (Bartlmae, 1999); Developing CBR applications (Althoff et al., 1999c; Bergmann et al., 1999); Ontology deployment (Kalfoglou and Robertson, 2000)	Reliance upon a vast amount of written documentation.
4	Alternative Approaches	Experience Engine Approach	Johansson et al. (1999)	Technical details and design issues have not been discussed.
		Project review technique	Kess and Haapasalo (2002)	
		Case Based Reasoning	Tautz and Althoff (1997)	
5	Knowledge Acquisition in SE	Overview of the knowledge acquisition	Birk et al. (1999) and Birk et al. (2002)	Too informal.
		Experience reports and Postmortem reviews	Dingsoyr et al. (2001)	Time consuming.
		Project-based experiential learning model (Workshops)	Dyba et al. (2004)	
		Lightweight approach to PMA	Schneider, 2000	Lack of standarisation.

Conclusions

Exhaustive literature search and analysis reveal that little research exists on the detailed use and structure of successful experience repositories, especially in software organisations. It is concluded that these studies contend that the issues related to knowledge creation, modification and sharing have important place in SPI initiatives, but to understand where and how to use KM insights to improve SPI practice, we need to experiment with and further develop different KM insights in SPI practice. This study has been an attempt in this direction.

References

1. Althoff, K.-D., Birk, A., Hartkopf, S., Muller, W., Nick, M., Surmann, D., Tautz. C.: Managing Software Engineering Experience for Comprehensive Reuse. In: Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering. Kaiserslautern, Germany. (1999a)
2. Althoff, K.-D., Bomarius, F., Mller, W., Nick, M.: Using case based reasoning for supporting continuous improvement processes. In: Perner, P. (ed.) Proceedings of the 12th German Workshop on Machine Learning (1999b)
3. Althoff, K.-D., Nick, M., Tautz, C.: CBR-PER: A tool for implementing reuse concepts of the experience factory for CBR systems. In: Proceedings of the 7th German Conference on Knowledge Based Systems (XPS99). Workshop on Case-Based Reasoning (1999c)
4. Bartlmae, K.: An experience factory approach for data mining. In: Proceedings of the 2nd Workshop in Data Mining and Data Warehousing as Basis of Modern Decision Support Systems (1999)
5. Basili, V.R., Caldiera, G., Rombach, H.: The Experience Factory. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering, vol. 1, Chapter X, pp. 468–476, John Wiley & Sons, NJ, USA (1994)
6. Basili, V.R., Lindvall, M., Costa, P.: Implementing the Experience Factory Concepts as a Set of Experience Bases. In: Proceedings of the 13th International Conference on Software Engineering & Knowledge Engineering. pp. 102-109. Knowledge Systems Institute (2001)
7. Basili, V.R., Schneider, K., Hunnius, J.-P.V.: Experience in Implementing a Learning Software Organization. IEEE Software. pp. 46-49 (May/June 2002)

8. Bassi, L., Van Buren, M.: Valuing investments in intellectual capital. International Journal of Technology Management. vol. 18 (5), pp. 414 - 432 (1999)

9. Bergmann, R., Breen, S., Goker, M.; Manago, M., Wess, S.: Developing Industrial Case Based Reasoning Applications - The INRECA Methodology. LNAI. vol. 1612. Springer-Verlag (1999)

10. Brossler, P.: Knowledge Management at a Software Engineering Company - An Experience Report. In: Proceedings of the Workshop on Learning Software Organizations, pp. 163-170. Kaiserslautern, Germany (1999)

11. Conradi, R., Dingsoyr, T.: Software experience bases: a consolidated evaluation and status report. In: Proceedings of the 2nd International Conference on Product Focused Software Process Improvement (PROFES, 2000). pp. 391-406 (2000)

12. Diaz, M., Sligo, J.: How Software Process Improvement Helped Motorola. IEEE Software. vol. 14, pp. 75-81 (1997)

13. Dingsoyr, T., Brede Moe, N., Nytro, O.: Augmenting Experience reports with Lightweight Postmortem Reviews. In: Bomarius, F., Komi-Sirvio, S. (eds.). Proceedings of the Third International Conference on Product Focused Software Process Improvement (PROFES 2001). LNCS, vol. 2188, pp. 167-181. Springer-Verlag, Berlin Heidelberg (2001)

14. Dingsoyr, T., Conradi, R.: A Survey of Case Studies of the use of Knowledge Management in Software Engineering. International Journal of Software Engineering and Knowledge Engineering. vol. 12(4), pp. 391-414 (2002)

15. Dingsoyr, T., Royrvik, E.: An Empirical Study of an Informal Knowledge Repository in a Medium-Sized Software Consulting Company. Proceedings of the 25th International Conference on Software Engineering (2003)

16. Dyba, T., Dingsoyr, T., Moe, N.B.: Process Improvement in Practice: a Handbook for IT Companies. Kluwer Academic Publisher, Boston (2004)

17. Edwards, J.S.: Managing Software Engineers and Their Knowledge. In: Aurum, A., Jeffrey, R., Wohlin, C., Handzic, M. (Eds). Managing Software Engineering Knowledge. pp. 5-27. Springer-Verlag (2003)

18. Finnie, G., Wittig, G., Desharnais, J.-M.: Estimating software development effort with case based reasoning. In: Proceedings of the 2nd International Conference on Case Base Reasoning. pp. 13–22. Springer-Verlag (1997)

19. Henniger, S.: Capturing and formalizing best practices in a software development organization. In: Proceedings of the 9th International Conference on Software Engineering and Knowledge Engineering (1997)

20. Henninger, S., Schlabach, J.: A Tool for Managing Software Development Knowledge. In: Proceedings of the PROFES 2001. LNCS, vol. 2188, pp. 182-195. Springer, Heidelberg (2001)

21. Houdek, F., Schneider, K., Wieser, E.: Establishing Experience Factories at Daimler-Benz: An experience report. In: Proceedings of the 20th International Conference on Software Engineering, pp. 443-447 (1998)

22. Johansson, C., Hall, P., Coquard, M.: Talk to Paula and Peter – They are Experienced. In: Proceedings of the Workshop of Learning Software Organizations, pp 69-76. Kaiserslautern, Germany (1999)

23. Kalfoglou, Y., Robertson, D.: Applying experienceware to support ontology deployment. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering (2000)

24. Kess, P., Haapasalo, H.: Knowledge creation through a project review process in software production. International Journal of Production Economics. vol. 80(1), pp. 49-55 (2002)

25. Koennecker, A., Jeffery, R., Low, G.: Implementing an experience factory based on existing organizational knowledge. In: Proceedings of the Australian Software Engineering Conference (2000)

26. Land, L., Aurum, A., Handzic, M.: Capturing Implicit Software Engineering Knowledge. In: 2001 Australian Software Engineering Conference, pp. 108–114 (2001)

27. Lindvall, M., Rus, I., Jammalamadaka, R., Thakker, R.: Software Tools for Knowledge Management, Technical report. DoD Data Analysis Center for Software. Rome, N.Y. (2001)

28. Mendonca, M.G., Seaman, C.B., Basili, V.R., Kim, Y.-M.: A Prototype Experience Management System for a Software Consulting Organization. In: Proceedings of the 13th International Conference on Software Engineering and Knowledge Engineering. pp. 29–36. Buenos Aires, Argentina (2001)

29. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering. IEEE Software. vol. 19(3), pp. 26–38 (2002)

30. Rus, I., Lindvall, M., Sinha, S.S.: Knowledge Management in Software Engineering - A DACS State-of-the-Art Report. Fraunhofer Center for Experimental Software Engineering and The University of Maryland, Maryland (2001)