Sense Making for Software Project Management – *A Way of* Managing Knowledge in Software Development

Smita Srivastava, Shweta Kurda, Priya Dubey
Information Technology
Dronacharya College of Engineering, Gurgaon

***Abstract*- This paper is concerned about the most important part of knowledge management namely making sense of practice in a group of project managers in the Danish software company. Projects are unique and involve complexity, multi-activity tasks constrained by a predetermined time, cost and specification goals and their management is an important part of modern organizational life. They increase organizational flexibility and ability to deal with change. We apply a framework of sense making. The framework provides a rich conception of how meaning is created. It directs attention towards the creation and negotiation of construed realities and the project managers' expectations and experience. We discuss the value and relevance of this framework for software development companies and conclude that it provides substantial insight which could not have been achieved through an instrumental perspective on knowledge management.**

***Index Terms*- Knowledge management, software development, project management, sense making.**

I. INTRODUCTION

The area of concern is managing knowledge in software development, which has been of interest to IS researchers for quite a while. Examples are the experience factory (Basili & Green, 1994; Rus & Lindvall, 2002), the application of knowledge management theory to explain software process improvement (Arent & Nørbjerg, 2000; Baskerville & Pries-Heje, 1999; Kjærgaard & Kautz, Forthcoming; Mathiassen & Pourkomeylian, 2001). We describe the different ways of research on the topic and identify a lack of empirical research on how knowledge is managed in software companies and new proposals as well as lessons learned are continually suggested. Such research is valuable for developers and managers when they plan and engage in knowledge managing and relate to process descriptions.

For this purpose we empirically study about the knowledge management process among a group of project managers in the Danish software development company. We discuss the value and relevance of taking a sense making perspective and suggest further use of thereof. The framework allows us to investigate of our research questions which are: How do project managers draw on past experience when they share knowledge and collectively contribute to process descriptions in a project management handbook? And how do they collectively make sense of the project management handbook's contents and use?

The paper is structured as follows. The second section presents the research approach and methodology to improve our software development. The third section presents the theoretical framework which is used to evaluate the empirical findings, and the research design is introduced in section four. Section five holds the case description and section six is concerned about the analysis of findings, where it also makes suggestions for future research. Finally we provide our conclusions in section seven.

II. RESEARCH APPROACH AND METHODOLOGY

The research presented in this paper is based on an empirical case study and action research in a Danish software company. The challenges in software development are vast, and considerable research has addressed how these challenges can be met by passing on experience, hard earned knowledge. The research has so far led to two distinctly different and almost disjointed research paths: the technical and the social. Along the technical path we find the experience factory (Basili et al. 1994) advocating that a designated organization takes experiences from one project and transfer it to other projects. Along the social path we find applications of knowledge management theories to software process improvement. Based on social theories such as Nonaka's knowledge creation theory (Nonaka 1994; Nonaka and Takeuchi 1995) several studies acknowledge how difficult it is to move knowledge between its tacit and its explicit dimension and between the individual and the group: in software process improvement (Arent and Nørbjerg 2000; Kautz and Thaysen 2001); in software process maturity (Baskerville and Pries-Heje 1999); and in software process implementation (Mathiassen and Pourkomeylian 2003).

In summary, research along the technical path misses the important social dimension of understanding knowledge managing in software development. So far, research along the social path has been limited and with little appreciation of the role that process descriptions play in knowledge sharing.

III. THEORETICAL FRAMEWORK

As a theoretical framework for our study we use a process model. The process model is based on a theoretical model, originally developed and used to explain the establishment of a knowledge management initiative in a technology friendly organizational environment (Kjærgaard and Kautz 2008). In this study we use the model to explore the sense making processes which were in play in the process of constructing the project management handbook at SpaceSoft. The model explains knowledge management as a sense making process unfolding in the interaction between collective and individual cognition. The model operates with two sub-processes of the knowledge management process, creating and negotiating, each of which has a dominant construed reality as well as a set of cognitive processes.

IV. RESEARCH DESIGN

The research process falls into two related parts firmly linked through a collaborative practice research (CPR) approach as outlined by Mathiassen (2002): the first part follows an action research approach, and the second part is a practice study. CPR, is organized as action research complemented with experiments and practice studies in which researchers and practitioners closely collaborate.

The action research was conducted in a Danish software company, Space Soft, which develops software for the European Space Agency. In SpaceSoft the problem solving purpose was to improve software processes i.e. (1) diagnose the problems with the current software processes and practice; (2) design how to improve software processes and change software development; (3) take action to change accordingly; and finally (4) assess how well the improvements have been and then re-start a cyclic improvement process.

V. CASE DESCRIPTION – SPACESOFT

The focus of this article is on the improvement of project management and how a company named Spacesoft addressed this improvement of project management. SpaceSoft produces software as a subcontractor for the European Space Agency. The company was founded in 1992 and is rather old in the fast moving software development business, and it has lived through many changes. It develops a wide range of dedicated software for verification and validation, ground station control, and checkout systems and declares strategy to deliver fixed-priced software on time. In 2002 SpaceSoft decided to focus on improving its software processes and entered into a collaborative research project called Software Processes and Knowledge.

VI. ANALYSIS

This article shows three examples of content from the software project management handbook and then presents an overall analysis of the creation and negotiation of the handbook content. The three examples are: a technique for estimation, a template for project planning, and a procedure for shipment.

6.1 A Technique for Estimation:

For the creation of the handbook, the usual practice of estimation at SpaceSoft was an educated guess, qualified by the project managers' own experience. The estimate was subsequently presented to the CEO who, based on his experience and the interests of the company, discussed it with the respective project manager and made adjustments before approving it.

6.2 A Template for Project Plan:

SpaceSoft the project planning is a difficult discipline—primarily because plans were continuously changing or ought to be changed, but were not. Reasons for not making required changes to project plans were typically shortage of time when new requirements were accepted or when project managers did not realize that (or how) new requirements changed the course of an assignment's implementation. By the end of the last review workshop there was agreement among the project managers that the template was highly relevant and that they would adapt their work to it accordingly.

6.3 A Procedure for Shipment:

There had been several incidents in the past where projects had not been able to deliver products to the customer on time or with items missing due to problems with packaging and shipment. Based on this experience, the CEO suggested including a procedure in the handbook, which would ensure, in detail that all possible problems in the final shipping process were taken care of in due time and he made a first draft of this procedure himself. It ensures that mistakes and omissions were avoided while the critics argued that not every step should be executed if it was not relevant in a specific shipping context.

VII. DISCUSSION

The framework as a theoretical lens to explore knowledge managing in software development and can describe the analysis in the following three findings:

1. The project managers' dominant construed realities of the issues in question are essential for understanding how they make sense of these issues. Their different construed realities lead to a collision between their expectations and experiences.
2. The collisions are important and should be accepted because they trigger negotiations of a new construed reality.
3. The negotiation process is necessary for the project managers to work on their different construed realities. The negotiations should not be terminated prematurely for the short-term sake of time and efficiency, but at the risk of becoming ineffective on the long term. Resolving collisions take time and resources.

VIII. CONCLUSION

The case and the analysis illustrate a set of complex knowledge processes that the framework has helped us untangle. This understanding can be described as follows:

1. The project managers' construed realities are essential for the creation of a process description.
2. Collisions of their expectations and experiences are important and should be accepted.
3. Negotiation of the process description is necessary.
4. As a result the process description becomes a shared product, which the project managers can adapt

REFERENCES

[1]. Arent, J., and Nørbjerg, J., (2000).Software Process Improvement as Organizational Knowledge Creation—A multiple case analysis. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Wailea, Hawaii, p. 11.

[2]. Basili, V. and Green, S. (1994) Software process evolution at the sel. IEEE Software 11 (4), 58-66.

[3]. Kjærgaard, A. L., and Kautz, K., (2008). A Process Model of Establishing Knowledge Management: Insights from a Longitudinal Field Study. OMEGA, (36:2): 282-297.

[4]. Kautz, K., and Thaysen, K., (2001). Knowledge, Learning and IT Support in a Small Software

Company. *The Journal of Knowledge Management*, (5:4): 349-357..

[5]. Nonaka, I., (1994).A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, (5:1): 14-37.

[6]. Nonaka, I., and Takeuchi, H., (1995). *The Knowledge-Creating Company*, Oxford University Press, Oxford

[7]. Rus, I., and Lindvall, M., (2002).Knowledge Management in Software Engineering. *IEEE Software*, (19:3): 26-38.