Knowledge Management in Software Engineering

Wladimir Bodrow, Daniel Vila Suero, Alejandro Alonso Cabello

Abstract: *This research describes the relevance of knowledge management to software engineering. The applicability of knowledge management activities to different stages of software engineering is investigated and the value of them within the software life circles formulated.*

Key words: *Knowledge management, Software engineering.*

INTRODUCTION

The paper organized as following: in the second chapter we describe briefly the selected knowledge management approach. The third chapter is devoted to the software engineering body of knowledge. The next chapter represents the interdependencies between both and figure out the particular relevance of knowledge management approach to software engineering. Conclusions in the last chapter summarise the value of presented research for the software engineering.

KNOWLEDGE MANAGEMENT

There exist a number of approaches to knowledge management [1,2,3,4,5,6]. In this research we use the following one. Along the business processes we consider four core activities in respect to knowledge:

Knowledge generation – new knowledge in the system (application environment) will be generated in different ways: by brainstorming, scientific and applied research, social interaction, training, learning by doing, etc. Fundamental research regarding the generation of knowledge in the company was published by Nonaka and Takeuchi [7].

Knowledge storage – this activity is "responsible" for documentation of existing and newly generated knowledge in explicit and/or in implicit form. Various aspects of content management provide efficient support the storage of explicit knowledge. An important point in this context is the storage of organizational routines and existing customer relations.

Knowledge utilization – according to our focus on business processes this is the goal and output of knowledge life circle and consequently the aim of knowledge management. Because of knowledge specifics it is very difficult to recognize, measure, and document the utilization of the selected unit of knowledge in the particular business process or activity.

Knowledge sharing – within the knowledge based and knowledge oriented processes sharing occupies the exceptional position. It represents a kind of engine, which enables the generation, storage, and utilization of knowledge in the most business processes with the focus on knowledge. The well organized sharing has to provide every participant with the necessary knowledge at the right time and place in required quality. Different communications resources can be used for efficient knowledge sharing.

Above these four activities, which are directly connected to managing of knowledge we consider the two activities in the meta-level:

Knowledge identification – to be able to realize this activity one have to have certain amount of professional knowledge and capability for systematic analysis of knowledge. The investigation of customer requirement can initiate the understanding whether a single person or the company has the professional knowledge to be successful in the team or on the market.

Knowledge evaluation – this activity is focused on the investigation of the efficiency of available knowledge in knowledge utilization processes. Based on the evaluation results management can make their decisions and so improve the utilization of knowledge in the enterprise or in particular business process. European research activities in this field are currently focused on development of intellectual capital statement [8,9,10,11].

SOFTWARE ENGINEERING BODY OF KNOWLEDGE

The aim of presented study is the application of described knowledge management approach to software engineering. Software engineering is an excellent example of knowledge based and knowledge oriented process. There exist many approaches and concepts to software engineering. We follow the IEEE Computer Society definition of software engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. Software engineering contain also the study of approaches such application. [12]. The basis for the investigation presented below builds the Guide to the Software Engineering Body of Knowledge developed in the SWEBOK project of the IEEE Computer Society Professional Practices Committee [12]. The aim of this document is established in the following five objectives:

1) To promote a consistent view of software engineering worldwide.

2) To clarify the place–and set the boundary–of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics.

3) To characterize the contents of the software engineering discipline.

4) To provide a topical access to the Software Engineering Body of Knowledge

5) To provide a foundation for curriculum development and for individual certification and licensing material.

These were the decisive points to use the Guide in presented research as a basis document in respect to software engineering.

Ten knowledge areas of software engineering are defined in the Guide:

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods
- Software quality

Every of these areas have very specific characteristics and, therefore they need different approaches and techniques in order to capture, store and manage the professional knowledge.

The Guide uses a hierarchical organization to decompose each knowledge area into a set of topics with recognizable labels. Then, it treats the selected topics in a manner compatible with major schools of thought and with breakdowns generally found in industry and in software engineering literature and standards.

Additionally to listed above knowledge areas the knowledge areas of related disciplines are also considered in the Guide. They are Computer engineering, Computer science, Management, Mathematics, Project management, Quality management, Software ergonomics, and Systems engineering. Every particular knowledge area of software engineering is specified and investigated in detail in the Guide. The presentation of this investigation is not the aim of this paper – our focus is the analysis of the applicability of knowledge management approach to software engineering.

APPLICATION OF KNOWLEDGE MANAGEMENT TO SOFTWARE ENGINEERING

Due to the fact that both the knowledge management activities and the knowledge areas in software engineering are fundamental concepts to get to understand the knowledge management and the software engineering, it is important to explain the basic relations between them.

Software Requirements: As one of the most important knowledge areas, the Software Requirements area involves almost all of the knowledge management activities. The relevant activities in this knowledge area are **generation, storage, utilization, identification,** and **evaluation** of professional knowledge.

Although it is implicit that the knowledge stored in the software requirements documents will be shared, we found that the **sharing** of knowledge is not a part of the process of development and formulating of the software requirements.

Software Design: While designing the software product, we think that the most important point is being able to use the knowledge created in the *Software Requirements* phase and transform it into graphics, diagrams, etc. So it is possible to translate the requirements to a more technical “language” and store them in the project documentation.

The relevant activities in this knowledge area are **storage, sharing,** and **utilization** of knowledge.

Software Construction: This step is in action when the system itself will be built and the source code will be produced. Within this process it is important to check stored libraries and repositories from the previous projects where the code has been tested and proofed and therefore can be used to the current project. Also new lines of code will be written and new libraries could be stored in repositories of source code.

Taking all this in account, the relevant activities in this knowledge area are **generation**, **storage, utilization,** and **evaluation** of knowledge.

Software Testing: The Software Testing area involves running tests, and checking all aspects of the application. Because of that reason, all the knowledge management activities are applied at this moment.

Software developers have to identify the critic parts of the application, those that are going to be under more pressure when the system is running. Also, tests packages from previous projects can be utilized to test some parts and new ones have to be created specifically for the present project.

Apart from that, all these new tests should be stored for future applications and documents have to be created with the results of the testing. Therefore, the relevant activities in this knowledge area are **generation, storage, utilization, sharing, identification,** and **evaluation** of knowledge.

Software Maintenance: In the Software Maintenance knowledge area, may seem that all knowledge activities could be applied, but in fact the most significant in this area is the identification and evaluation of the knowledge needed to maintain the application with the less effort and more efficiently. Also it is important to share this knowledge with the rest of the team and so provide the support for everyone work. Based on this, the relevant activities in this knowledge area are **sharing, identification,** and **evaluation** of knowledge.

Software Configuration Management: This knowledge area is more or less a supporting software life cycle to help with the project management, development, etc. Consequently all the knowledge activities will be applied in this knowledge area.

The relevant activities in this knowledge area are **generation, storage, utilization, sharing, identification,** and **evaluation** of knowledge.

Software Engineering Management: As the previous knowledge area, the Software Engineering Management involves a lot of different tasks, like planning, coordinating, monitoring, etc. to ensure that the development and maintenance of software are systematic, quantified and disciplined.

Therefore, the relevant activities in this knowledge area are **storage, utilization, sharing, identification,** and **evaluation** of knowledge.

Software Engineering Process: Analogous to the Software Configuration Management Area, the Software Engineering Process area is related with the complete software life cycle. Among others it contain changes that induct some improvements of software engineering.

Taking the process perspective of the whole life cycle schema into account one can point out that the relevant activities in this knowledge area are **generation, storage, utilization, sharing, identification,** and **evaluation** of knowledge.

Software Engineering Tools and Methods: In order to recognize which tools or methods are the best to be applied to the software product, is most important to know what has to be build up and for what purpose or use.

Then, the relevant activities in this knowledge area are **utilization, identification,** and **evaluation** of knowledge.

Software Quality: In order to improve the quality of the software product, it is important to identify which parts or modules of the application should be improve, how and with what purpose. As a result, the relevant activities in this knowledge area are **storage, utilization, identification,** and **evaluation** of knowledge.

All described interdependencies are presented in the figure below.

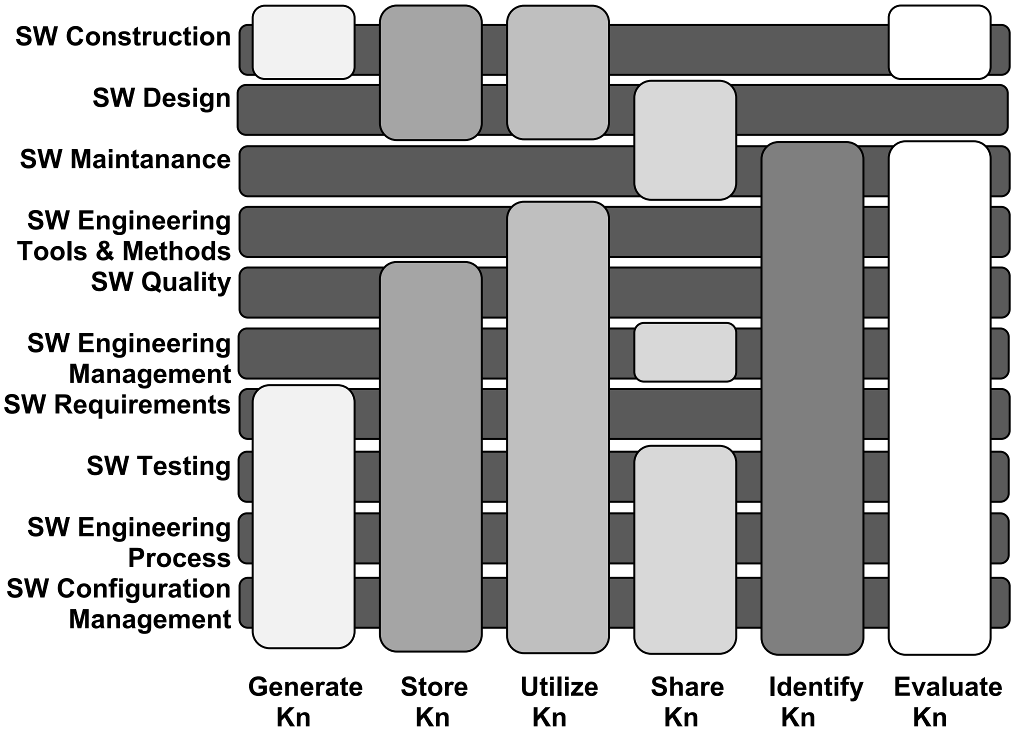

Fig.1. Knowledge areas and knowledge activities
(Abbreviations: SW – software, Kn – knowledge)

In the graphic, the knowledge areas (software engineering) are represented with horizontal bars, and the six knowledge management activities are represented by vertical bars. The interrelations are represented with the "crossing" of the activities over the knowledge areas. That is, if the selected activity is relevant in a particular knowledge area, the vertical bar representing it will be above the horizontal bar representing the knowledge area.

CONCLUSIONS AND FUTURE WORK

In this paper we analysed the applicability of knowledge management approach to software engineering. The relevance of selected knowledge management activities to all knowledge areas of software engineering was figured out and presented in the overview. This presentation provide a) the better understanding of knowledge value of each knowledge area in software engineering. And b) it allows the practitioners (software developer) to improve their activities within software engineering with respect to managing their professional knowledge. The next step in this research is the development and implementation of the skills manager tool to manage the professional software engineering knowledge in the practice.

REFERENCES

[1] Holsapple, C.W. (ed.), Handbook on Knowledge management I, Springer, Berlin, 2003

[2] Mertins K. / Heisig P. / Vorbeck P. (eds.), Knowledge Management,. Springer, Berlin, 2001

[3] Bodrow, W. / Bergmann, Ph., *Wissensbewertung in Unternehmen*, Erich Schmidt Verlag, Berlin, 2003

[4] North, K., *Wissensorientierte Unternehmensführung*, 3. Ed., Gabler, Wiesbaden, 2002

[5] Probst, G. / Raub, S. / Romhardt, K., *Wissen managen*, 3. Ed., Gabler, Wiesbaden, 1999

[6] European Guide to good practice in KM, Proceedings of CEN/ISSS workshop in Madrid, 2003

[7] Nonaka, I. / Takeuchi, H., *The Knowledge Creating Company*, New York, Oxford, 1995

[8] Koch, G., *ICR Made in Austria*, in: Contribution at Intellectual Capital for Comunities, Paris, 2005

[9] Mertins, K./ Alwert, K. / Heisig P. (eds.) *Wissensbilanzen*, Springer, Berlin, 2005

[10] http://www.incas-europe.org/index-en.htm/, last visit 23.7.2009

[11] http://www.seneca.de/, last visit 23.7.2009

[12] http://www2.computer.org/portal/web/swebok, last visit 23.7.2009

ABOUT THE AUTHORS

Prof. Dr. Wladimir Bodrow, PhD, Business Informatics, University of Applied Sciences HTW-Berlin,

Phone: +49 (030) 5019-2478, E-mail: wladimir.bodrow@htw-berlin.de

Daniel Vila Suero, Universidad Politecnica de Madrid.

Alejandro Alonso Cabello, Universidad Politecnica de Madrid.