Developing Knowledge Factory System as a Best Practice Platform in Software Process Environment

Mazurah Abd Kudus & Rusli Abdullah

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia,
43400 UPM Serdang, Selangor

Abstract
'Knowledge' is one of the main outcomes systematic approaches of software engineering. It is considered as a crucial resource of each organization and, therefore, needs to be managed carefully especially for the software development process and the roles involved in each phase. The effectiveness application of software engineering relies on experiences, and organizations need to capitalize on their experiences by capturing and reusing them. In all approaches of experience reuse, repositories play a central role in enhancing the knowledge in experience transfer of process knowledge between projects. Like in a factory beside the production there are the people who produce. Experiences should not be stored on a dusty shelf, but engineered into best practices and processes that could be guidance to all roles in the software project especially for the developer.
This paper presents the approach to the implementation of the project knowledge based on software development life cycle.It also describe on how the knowledge factory solution can be used as a decision support system for software project managers in day-to-day development activities. The paper concludes with an evaluation of our approach to date and a description of future research directions.

Keywords
Knowledge Management (KM), Software Development Process (SDM), Knowledge Factory (KF), Best Practice

1. Introduction

Knowledge is essential in everyday work. Everyone learns by experience and this knowledge is later reused in similar kind of tasks again by adapting this knowledge to a new situation.

For an organization, past experience also plays a key role in improvement and management. How effectively past experience can be leveraged depends on how well this experience is captured and organized to enable learning and reuse. Systematically recording data from projects, deriving lessons from it, and then making the lessons available to other projects can enhance this reuse. The knowledge management literature usually deals with the mechanisms of knowledge handling, while learning approaches address the process of how to gain knowledge.

An organization which is in software development business as the main asset is the intellectual capital where knowledge management is particularly important [10]. The major problem with intellectual capital is that it has legs and walks home every day. At the same rate experience walks out the door, inexperience walks in the door. Many software organizations admit it that they face the challenge of sustaining the level of competence needed in winning contracts and fulfilling undertakings.

Knowledge can be external, i.e. which is produced by people outside the organization. This type of knowledge resides in books, journals, magazines, etc. Knowledge can also be internal, i.e. the knowledge that is created primarily within the organization, largely through experience and experimentation. Leveraging experiential knowledge is the focus in the experience factory model [15].

This paper is structured as follows. First, I provide the basic background information on knowledge factory, current problem and the objective of this research. Related work of knowledge factory is discussed in Section 2. Section 3 describes the methodology used in this research. Section 4, discusses the result of this research. Finally, Section 5 concludes this paper.

2. Literature Review

Over the past decade, the notion of knowledge sharing in Software Engineering (SE) activities such as change management, requirement and design documentation and many more has gained considerable interest from both the research and industrial communities.

Markulla [31] describes an initiative at ICL Finland to promote software engineering knowledge sharing and reuse. The focus is on supporting development tasks, such as planning, design and coding. Althoff [32] propose a generic, scalable architecture and its underlying methodology for reuse and continuous learning of all kinds of software engineering experience. Also they used a project as a scenario for demonstrating their approach. The focus is on reuse which will be more enhancement in this paper.

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

Furthermore, NovaForge [5] establish a shared database responds to the need to optimize development projects by capitalizing on best practice, tools and procedures that have already been tested within the organization and are re-usable. Looking to these works, we can find many common points. All of them, including mine, are based on the concept of Experience Factory [15]. Knowledge factory is used also in a similar way but it is depend on how it is organized.

Centering the information around a single repository through an ongoing process of capturing project experiences can prevent the duplication of efforts, avoid repeating common mistakes, and help streamline the development process. Scott [12] describes that he had coupled the process and technology to turn SDM documents into dynamic "living" resources that can be extended and improved as new project needs and application requirements emerge. As the repository accumulates through principled evolution of the SDM, it improves to enable handling a wider range of circumstances, while evolving toward answers to problems that fit the organization's technical and business context. The real question is not whether the repository is "correct" in some objective sense, but rather whether less mistake are repeated and better solutions adopted when using the repository.

3. Methodology

The research methodology that is applied in developing the knowledge factory is adopted from waterfall model in the Software Development Life Cycle (SDLC). The methodology begins with the Requirement Analysis in order to understand and gather initial requirements about the knowledge factory approach. In this phase any related literature regarding the implementation of knowledge sharing in software endgineering activities and central repository are studied in detail and analyzed. Then, based on the findings, it gives me ideas on how to structure and develop the knowledge factory.

3.1 Requirement Analysis

KF does not ignore the value or need to address other software development aspects, such as process and technology, nor does it seek to replace them. Instead, it works toward software process improvement by explicitly and systematically addressing the management of knowledge. This includes its acquisition, structuring, storage and effective maintenance. There are two main strategies to manage knowledge:

1. codification or centralization: making tacit knowledge to explicit knowledge
2. personalization : supporting knowledge sharing by describing who knows what.

It is important to understand and identify what kind of knowledge that need to be captured for each phases in software development process. The system is to be used by different role in software development process from various projects in an organization. The following essential requirements is included in the system:

a. The users of the system would find various useful data and related in finding solution for their problem.
b. The users of this system are able to capture, edit and query the information available.
c. All the information can be shared and reuse.
d. The users can collaborate among them using the available tools like community forum.

The KF has also been logically divided into project knowledge (concrete) and organization knowledge (generic). Another requirement of reusability is an appropriate structure to enable tailoring and generalizing knowledge. This issue has been addressed in this paper by designing a generic template to capture and present architecture design knowledge based on software development process cycle accordingly.

3.2 System Scope

Knowledge factory portal such as best practise repositories are widely used by practitioner but are under-researched partly due to informal, unstructure nature and large scale. However, the need for understanding informal and unstructured repositories has been well-recognized.

One of the main sources of implicit knowledge is people (e.g. solution architects, domain experts and project teams), who individually and collectively carry a large amount of "*know-how*". There are two main strategies to capture such implicit knowledge to populate a knowledge repository: 1) each role need to capture implicit knowledge from individuals or teams or 2) provide appropriate tool support so that knowledge can be encoded into the system as part of the knowledge creation process. These strategies are the main scope of this paper. Furthermore the outcome of the storage assists the management in making decision that's relates to any issues raised within a project.

3.3 System Design

For this stage, all requirements is translated into specific user interface design that helps every team member in a project may involved in capturing explicit or tacit/implicit

knowledge based on their roles respectively. The following approach has been taken in implementing the factory:

Step 1: Pre-configuration. In order to be more suitable to the targeted domain, the factory can be configured with a simple information structure before being used. First, the main user for this factory is the individual with specific role in the organization or project specifically. Then, we configure the factory based on each project with the stages in software development process as shown in Figure 1. As such, it helps user to capture any knowledge to the appropriate stage in their project. This rationale is essentially the perceived evidence that why a best practice works.

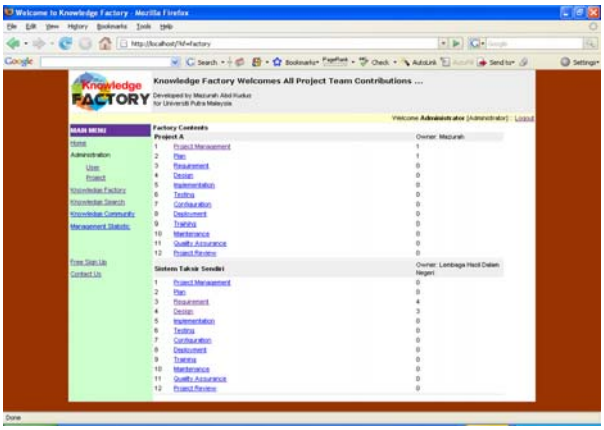

Figure 1 : Knowledge Factory Main Page

Step 2: Knowledge Capturing. In this paper , we proposed the generic template to capture the explicit knowledge such as deliverable document, minutes and so on. Figure 2 shows the generic template for a project. However, the template may vary among stage.

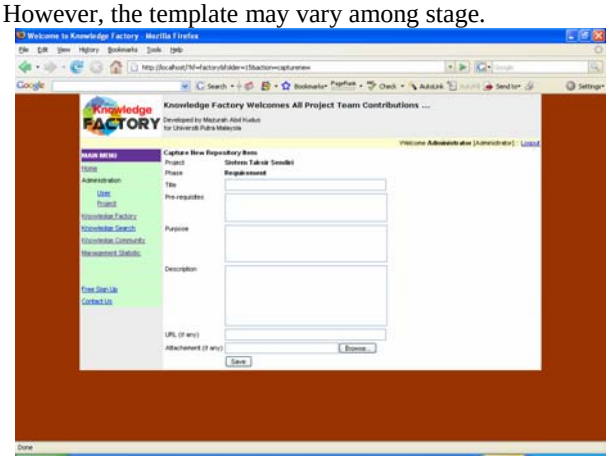

Figure 2: Knowledge Capturing screen

Step 3: Transfer experience. In order to ensure that individual experience is captured and shared among the team member discussion channel is organized based on the category of the software development stage. By having the structure, discussion are focused and easy for retrieval.

Step 4: Retrieval. The factory information can be retrieved by using common searching function and also through search criteria of the factory with the keyword search.

Step 5: Management View. The most important to the management is to ensure the project deliverables is productive and on time. Hence they need to monitor the ongoing project. We would suggest that statistical function can help them in making decision on certain issues.

Depending on the purpose, the factory can be used in different ways. We consider that the main contribution of the factory is organizing the knowledge and the contributor involvement in sharing their experience. This is useful as best practise work and any new issues can be discussed earlier.

3.4 System Architecture

The Knowledge Factory portal is developed based on a web based clients architecture whereby the system will be located on the server side. The clients and server are connected through the internet connection and each user is able to access the system simultaneously from their respective computers. The system architecture is further illustrated in Figure 3.

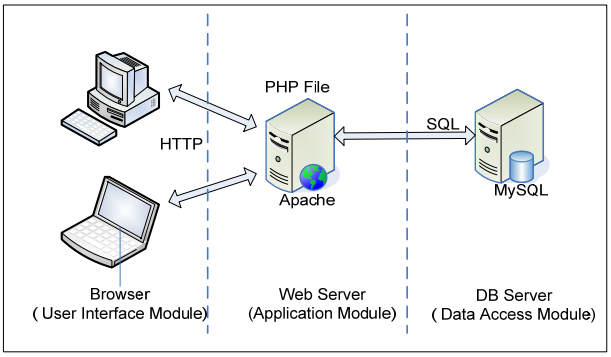

Figure 3 : System Architecture

3.5 Implementation

The Knowledge Factory portal implementation, an action has been taken whereby a series of application development were carried out. In this phase, every functional module is developed based on system design

and analysis thus program codes are written and documented.

The development environment used to construct this portal is Macromedia Homesite 5.5 which also included Macromedia Dreamweaver MX 2004. Macromedia Dreamweaver is a kind of tool act as editor to write PHP codes for web based application. XAMPP is an integrated server package of Apache, MySQL and PHP. The hardware platform used to develop the portal is a common computer with system specification as follows:

- Intel Pentium, CPU 1.8 GHz
- 512 MB Memory

4. Results and Discussion

In this section, the result of the knowledge factory portal that has been developed will be discussed. The portal also has been evaluated by the selected roles in software development project from two different project and the evaluation results as shown in graph (Figure 4 and Figure 5).

A survey has been conducted where 10 participants has been selected from two projects in selected software organization with a different roles in software development team. The roles of them are software development manager, software engineer, system engineer, test engineer and configuration engineer. The hands on of the exercise were done at participant's workstation based on their free time. Each participant was given 10 minutes to hands on the portal and another 10 minutes to answer the survey question.

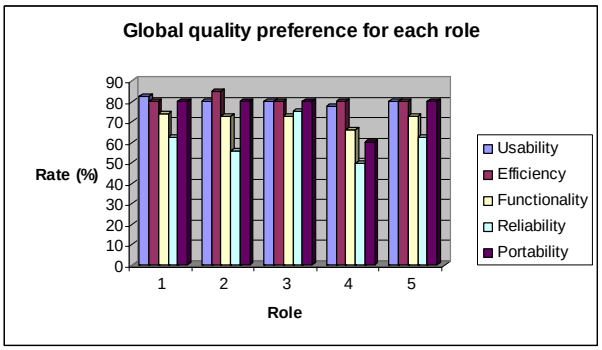

Figure 4: Global quality preference by roles

Figure 4 shows the acceptance level of quality criteria: *usability, efficiency, functionality, reliability and portability* by each roles. The result describes that more than 50% of each roles satisfy with the portal.

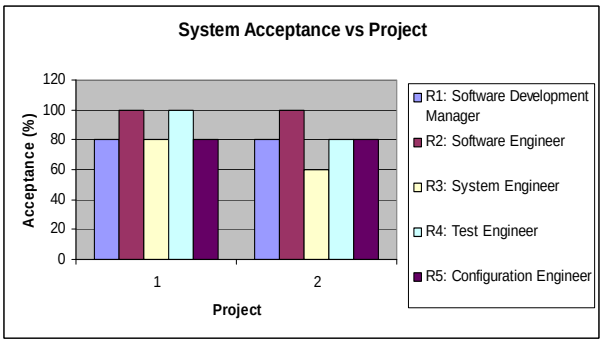

Figure 5: System Acceptance Level By Five Roles From Two Projects

Generally, the system is acccepted by most of the team member in software development team which is shown in Figure 5.

From the survey, it shows that the Knowledge Factory is very helpful for several purposes:

1. Solving a technical problem which is something to look up and follow.
2. Improving individual work station which used the information found in the factory by adjusting their technical tools.
3. Avoiding redundancy in the sense of having to explain same solution to several people.
4. Getting an overview of problem areas which is team member always check what was new and they will know what is in there and do not have to search.
5. Finding competence means that through the factory we can find those who has specific competence in the company.

It is also like newsgroup in one sense but contains mostly knowledge to the organization.Knowledge Factory can be use of 'credit' that people get feedback on what they have contributed and are thus motivated to contribute more. If a company emphasis on knowledge sharing, the employee who has written many experiences may get small reward as the knowledge sharer of the month.

5. Conclusion and Future Work

The prototype system offers a wide range of advantages in software development environment. Some of the advantages are:

- The use of internet / intranet: By using intranet / internet, this system can be reached anywhere and at anytime. This will shorten the time taken, as well as effort and cost effective.

- Computerized system: The whole system is using the computer technology, so the cost of using paper by the old manual system can be overcome. The information will be much safer to store, accountable and easy to refer to later on.
- Communication and discussion: This system provides facility where every user can communicate and discuss among user regardless they are from other project. So every user can exchange information and idea among them.
- Knowledge sharing: In this system, experience and document can be shared among team member in software development project. New team member may gain the project knowledge from the factory without seing the former team member.

The prototype system only provides an initial demonstration of the feasibility of the KF approach and the effectiveness used in software development process. Therefore, future development needs to be more focus and specialize in every cycle of software development process. First, more collaborative tools to be included in order to ensure the team member are easily connected whenever any past experience need to be discussed in detail. Furthermore, detail attributes need to be extracted and more automation function included. This can facilitate the KF to be more effective, fully utilize and give more contribution to the successful of future project in an organization.

The project implementation has initiated some ideas on setting up the knowledge factory as the solution for the research problem. The implementation of the knowledge factory may vary from organization to organization. It can be further continuously refined and evaluated with the real user's participation from each cycle in software development process. The real user evaluation is expected to provide more valuable input to the refinery of knowledge factory. No doubt to say that, Knowledge Factory is the best practice in inspiring organizations to improve their software development process.

6. References

[1] Lars Mathiassen and Pouya Pourkomeylian, "Managing Knowledge In A Software Organization", Journal of Knowledge Management, Vol. 7 No. 2 2003, pp 63-80

[2] V. Sappanen, S. Komi-Sirvio, A. Mantyniemi, T. Rahikkala and M. Pikkarrainen, "Context of KM Based SPI: The Case of Sofware Design Experience Reuse", PROFES 2001, LNCS 2188, pp 57-67,2001.

[3] Scott Henninger and Jason Schlabach, "A Tool For Managing Software Development Knowledge", PROFES 2001, LNCS 2188, pp 182-195, 2001.

[4] Gerardo Canfora, Aniella Cimitile and Corrado Aaron Visaggio, "From Knowledge Management Concepts Toward Software Engineering Practices, PROFES 2001, LNCS 2188, pp 407-422, 2001.

[5] NovaForge, "Software Development Factory", January 2007, www.bull.com

[6] Gunther Ruhe, "Learning Software Organization", Fraunhofer Institute for Experimental Software Engineering *(IESE)*

[7] Peng Xu, "Knowledge Support in Software Process Tailoring", Proceedings of the 38th Hawaii International Conference on System Sciences – 2005

[8] Muhammad Ali Babar, Ian Gorton, and Ross Jeffery, "Capturing and Using Software Architecture Knowledge for Architecture-Based Software Development", Proceedings of the Fifth International Conference on Quality Software (QSIC 2005)

[9] Victor R. Basili, "The Experience Factory Organization", IEEE Software May/June 2002, pg 30-31.

[10] Kurt Schneider and Jan-Peter von Hunnius, "Effective Experience Repositories for Software Engineering", IEEE 2003, pg 534-539.

[11] Warren Harrison, "Best Practices: Who Says?", IEEE Software 2004, pg 8-11.

[12] Scott Henninger, "Capturing and Formalizing Best Practices in a Software Development Organization", Int'l Conf. Software Engineering and Knowledge *Engineering (SEKE '97)*, Madrid, Spain, 1997, pp. 24-31.

[13] Kenyer Dominguez, Maria Perez, Luis E.Mendoza and Anna Griman, "Quality in Development Process for Software Factories According to ISO 15504", Universidad Simon Bolivar,Venezuela.

[14] Torgeir Dingsoyr, "An Evaluation of Research on Experience Factory", Norwegian University of Science and Technology, Norway.

[15] Victor R. Vasili, Gianluigi,Caldiera, H. Dieter Rombach, "The Experience Factory", Encyclopedia of Software Engineering, 1994: John Wiley and Sons.

[16] Ioana Rus and Mikael Lindvall, "Knowldege Management in Software Engineering", IEEE Software, May/June 2002, pp26-30.

[17] Tushar K.Hazra, "Building Enterprise Portals: Principles to Practice", ICSE 2002, pg 623 – 633.

[18] "Ultimate Aim – Software Factory", 2002, www.processwave.net/Essays/Software Factory

[19] Kieron O'Hara and Nigel Shadbolt, "Managing Knowledge Capture: Economic, Technological and Methodological Considerations", University of Southamton, UK.

[20] B.Faust, "Implementation of Tacit Knowledge Preservation and Transfer Methods", IAEA-CN-153/2/P/24, 2006.
[21] Fred Nikols, "The Knowledge in Knowledge Management", 2003.
[22] Larry Wilson and Pam Holloway, "The 10-Page Guide to Eliciting Implicit Knowledge (Version 1.5)", 2000, www.knowledgeharvesting.com
[23] Victor Basili, Patricia Costa, Mikael Lindvall, Manoel Mendonca, Carolyn Seaman, Roseanne Tesoriero, and Marvin Zelkowitz, "An Experience Management System for a Software Engineering Research Organization", 2001.
[24] Torgeir Dingsøyr and Reidar Conradi, "A Survey of Case Studies of the Use of Knowledge Management in Software Engineering", Norwegian University of Science and Technology, Norway.
[25] Kumiyo Nakakoji, "Supporting Software Development as Collective Creative Knowledge Work", Proceedings of the Second International Workshop on Supporting Knowledge Collaboration in Software Development (KCSE2006), Ye, Y., Ohira, M., (Eds.), NII, Tokyo, pp.1-8, September, 2006.
[26] Cat Kutay and Aybuke Aurum, "Knowledge Transformation for Education in Software Engineering", Int. J. Mobile Learning and Organisation, Vol. 1, No. 1, 2007, pp 58-80.
[27] Ligia da Motta Silveira Borges and Ricardo de Almeida Falbo, "Managing Software Process Knowledge", Vitória - ES – Brazil.
[28] Ralf Kneuper, "Supporting Software Processes Using Knowledge Management", Handbook of Software Engineering and Knowledge Engineering Vol. 2 @ World Scienti_c Publishing Company.
[29] Wangenheim, C.G., D. Lichtnow, A. Wangeheim, and E. Comunello, "Supporting Knowledge Management in University Software R&D Groups, in Advances in Learning Software Organization", K.-D. Althoff, R.L. Feldmann, and W. Muller, Editors. 2001, Springer.
[30] Philip Friedlander and Doug Collins, "Component-Based Software Development and The Software Factory", AICPA InfoTech Update, March/April 2002.
[31] M. Markkula, "Knowledge Management in Software Engineering Projects", in Proc. of the 11th Int. Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslautern, Germany, 1999.
[32] K-D. Althoff, A. Birk, S. Hartkopf, W. Muller, M. Nick, D. Surmann, and C. Tautz, "Managing Software Engineering Experience for Comprehensive Reuse", in Proc. of the 11th Int. Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslautern, Germany, 1999

Mazurah Abd Kudus is an ex-student of Faculty of Computer Science and Information Technology of University Putra Malaysia. She was just graduated in Master of Computer Science (2008) and currently working in IT department of semi-government sector.

Rusli Abdullah is a senior lecturer in Information System Department, Faculty of Computer Science and Information Technology of University Putra Malaysia. He holds a B.Sc in Computer Science from University Putra Malaysia (1988), M.Sc in Computer Science from University Putra Malaysia (1996), and PhD in Knowledge Management field at Faculty of Computer Science and Information System at University Tecknologi Malaysia (2005). He has more 12 years of teaching experience and with about 8 years of system development experience as a system analyst at higher learning institution.