

# TOWARDS DEVELOPING LESSON LEARNT AND EXPERINCE-BASED FACTORY MODEL IN AGILE SOFTWARE DEVELOPMENT

**Rusli Abdullah, Nik Sakinah Nik Ab Aziz**

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

e-mail: [rusli@upm.edu.my](mailto:rusli@upm.edu.my)

e-mail: [sakinah112@gmail.com](mailto:sakinah112@gmail.com)

**Modi Lakulu**

Universiti Pendidikan Sultan Idris  
35900 Tg. Malim, Perak Darul Ridzuan

e-mail: [modi@fskik.upsi.edu.my](mailto:modi@fskik.upsi.edu.my)

**ABSTRACT** - *Lesson Learnt and Experience-Based Factory are two concepts that have been given in promoting knowledge as a best practice as well as to encourage software engineers to work together and shared their knowledge in developing the software especially which related to agile software development environment. By utilizing these concepts as a model of knowledge sharing process, software engineers can save time and work faster in producing the software as a product that is based on agile software methodology environment. In the context of knowledge sharing process among the software engineers as a model generation, this conceptual model is discussed based on knowledge management process which consists of knowledge acquisition, storage, dissemination and application. The model also will be discussing in terms of its functionality that to be utilized towards beneficial impact of the community of practice (CoP) in agile software development of the future. Therefore, there is a need of a model that focuses on lesson learnt and experience-based factory which is related to agile system development in becoming as a standard guidance to those who are involved in software development environment. Furthermore, by describing all those functionality and non-functionality requirements of the application, it will be covering of all aspects of services especially related to agile software development requirement, which is providing the efficient and effective of services or also called as quality of services (QoS) for community of practice (CoP) in a particular environment.*

**Keyword:** Best Practice, Knowledge Sharing, Quality of Service, Agile Software Development, Community of Practice

## 1. INTRODUCTION

The knowledge utilization or application of best practice or called as Lesson Learnt and Experience-Based which also called as a factory in software development (SD) is considering very importance element for everyone who are getting involved on the matter in ensuring they can deliver a good software or application to the respective user. By using these kinds of knowledge factory in SD, it will encourage the community of practise (CoP) such as software engineers and programmers to share knowledge as well as to promote them to work together in developing the software especially which is related to agile software development environment.

By utilizing these concepts as a model of knowledge sharing process, the CoP also can save their time and working faster in producing the software as a product for the specific user that is based on agile software methodology environment.

In the context of knowledge sharing process among the members of CoP as a model proposal, this conceptual system design model is discussed based on knowledge management (KM) process which are consisting of knowledge acquisition, storage, dissemination and application. The system design model also will be emphasized in term of its functionality that to be utilized towards beneficial or impact of their members of CoP in agile software development of the future. Therefore, the paper will focuses on lesson learnt and experience-based factory which is related to agile system development in becoming as a standard guidance to those

who are involved in software development (SD) environment. Furthermore, by describing all those functionalities and non-functionalities requirements of the application as a tool, it will be providing the efficiencies and effective of services or also called as quality of services (QoS) for their members of CoP in a particulars business application with agility method of SD environment

## 2. Literature Review

In the context of literature review analysis, the discussion will be starting with knowledge definition, knowledge management (KM) processes and how KM is used as a system in managing knowledge sharing for community of practice (CoP) in agile software development environment.

### 2.1 Knowledge

Knowledge is an actionable of information. It consists of tacit and explicit knowledge which is including an experience of best practise as lesson learnt of human being that is considering as a major part of knowledge criteria or characteristics. The most popular knowledge in a particular business of CoP for the purpose knowledge sharing process is normally involving the knowledge in a form of explicit rather than implicit or tacit. This scenario is happening because the explicit knowledge is easy to refer and browse through its well organized in a documented of presentation or called knowledge repository of the organization.

### 2.2 Knowledge Management and Its Process

Knowledge Management (KM) is a concept of managing knowledge which is can be divided into some of processes. These of KM processes are including Knowledge Construction where to become as an initiator for knowledge acquisition purposes, and it goes to Knowledge Dissemination for knowledge informed, and Knowledge Application or Use of CoP, and lastly ending by Knowledge Embodiment based on CoP purpose and specific requirement. The relationship of those processes and paradigm in related to KM process environment as what been discussed by other researchers such as AM Talib et al. [3], and R Abdullah et al. [6], as well as by McAdam and McCreedy's [7,8,9] is shown in Figure 1.

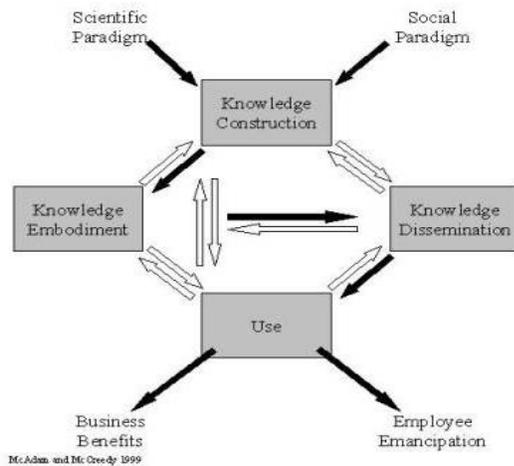


Fig. 1: Knowledge Management Process Model [9]

**2.3 Agile Software Development**

From Wikipedia [11] “Agile software development” refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The term was coined in the year 2001 when the Agile Manifesto was formulated [10]. Agile represents a group of software engineering methodologies which promise to deliver increased productivity, quality and project success rate overall in software development projects. Such methodologies are SCRUM [2,3], eXtreme Programming (XP) [4], or the lesser-known Crystal [5]. The outline of Agile Methodologies was laid down by the Agile Manifesto, published by a group of software practitioners [4,8]. In Agile Methods, it has required less formal training than traditional methods. Besides that, in terms of programming wise, the pair programming helps minimize what is needed in terms of training, because people mentor each other. Mentoring is more important than regular training that can many times be completed as self-training. Furthermore, the training material is also available for those

who are getting involved in agile development particularly for XP, Crystal, Scrum, and Feature Driven Development (FDD). Figure 2 show the steps or phase that need to be considered for agile which is consists of requirement, architectural design, development and followed by testing and feedback.

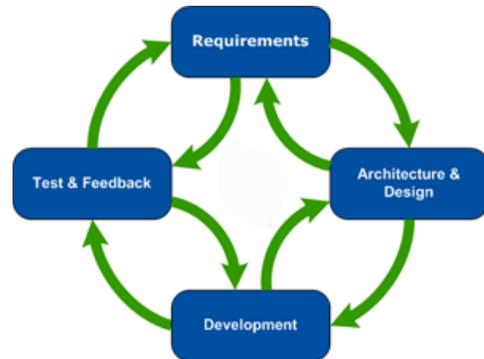


Fig. 2: Agile Development Proses [10]

**2.4 Knowledge Management and Agile Software Development**

For the purpose of managing knowledge in agility software development (SD), the knowledge can be categorized and stored in the knowledge repository as system called Knowledge Management System (KMS). The categorization of knowledge is managed based on agile development process as what been shown in Figure 1 and Figure 2. In this context, the KMS will also be functionalized and allowed the community of practice (CoP) to work together collaboratively and at the same time or different place in promoting knowledge sharing process among CoP for developing system application by using agile methodologies [1]. The next section will be explaining more on how the KMS as a tool that allowed CoP to work on agile method in producing a software as a product for the business on the respective organization.

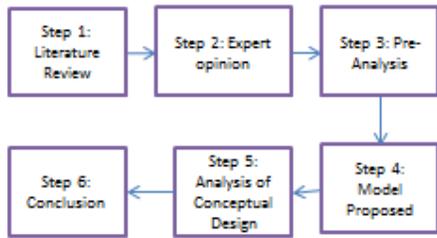
A new access control is named as Formula-Based Cloud Data Access Control (FCDAC) has been introduced to manage knowledge management in cloud computing [12]. Modi Lakulu, *et al.* [13], discuss the formulation of knowledge management system (KMS) framework for sharing knowledge in OSS using SDLC from the planning phase until the maintenance phase. The purpose of this KMS model is to allow OSS Community of Practice to share the knowledge.

The model can be evaluated based on quality and usability to ensure the use of it. A technique and some measurement are needed to evaluate a quality and the usability of the model. Four quality characteristics named Service Content, System Functionality, Information Technology and System Reliability found to be useful and effective for ensuring that high quality systems are developed [14]. According to ISO 9241-11, measures of effectiveness relate the goals or sub-goals of the user to the accuracy and completeness with

which these goals can be achieved. For usability measurement, it can break into four parts which are content, organization, and readability, navigation and links, user interface design, and performance and effectiveness [15].

**3. METHODOLOGY**

In order to formulate and propose the model of knowledge sharing of best practice as a model and its application in agile software development environment, there are few steps that has been taken and conducted based on a series of sequences as shown in Figure 3.



**Fig. 3: Steps in Knowledge Sharing Model and Its Formulation**

The methodology of the research is started by performing the analysis of literature (Step1) regarding on the agile software development and its related to resources and then followed by conducting a preliminary survey (Step2) through the expert opinion interviewed that based on those who are really involved in dealing with the agile software development environment such as software engineers, database administrator, agile programmers and agile activist users.

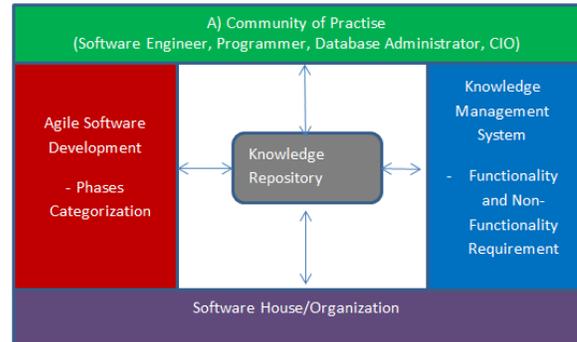
At this stage, a preliminary analysis has been done in formulating the knowledge sharing model (Step3) as a propose model which closed related to agile system development environment (Step4). After that, the simple measurement of the knowledge sharing model as a conceptual system and it application system design model is also analyzed (Step5) in determining the best criteria of service level in agile system environment (Step6) as an ending stage of steps which is also including the conclusion stage.

**4. Proposed Knowledge Sharing Model of Lesson Learnt and Experience-Based Factory**

The proposed model of knowledge sharing lesson learnt and experience of in agile software development (KS) environment is called as KS-LEF system. The conceptual system is developed which is based on what was been suggested by Rising [10] of agile SD, and their relationship with KM process as well as their system functionalities is shown as in Figure 4. While, Table 1 is showing on the how the KM as a system (KS-LEF) can be functionalised and non-functionalised based on the CoP requirements.

In this context, the KS-LEF is consisting of knowledge repository as a factory that knowledge is generated and shared by the CoP and it will be functioning based on KM processes for agile selection method of SD environment. Besides that, the KS-LEF also will be serving to the CoP based on the following main features which are including the knowledge template as knowledge ontology, knowledge

indexes for knowledge retrieval, warning and alerting functionality is used for informing of new knowledge generation and given by some others CoP regarding on agile knowledge for the benefits of CoP.



**Fig. 4: Conceptual Knowledge Sharing Model of Agile Software Development (KS-LEF)**

**Table 1: Knowledge Sharing Implementation in Agile Software Development**

Agile Software Development	Knowledge Management Process	Lesson Learnt and Experience-Based Factory
-Pre-project (Requirement Engineering) -Project Life Cycle based on Agile Methodology (eXtreme Programming, Crystal, Scrum, Feature Driven Development (FDD) -Post project (Development and Implementation)	Knowledge Acquisition	Knowledge Template
	Knowledge Dissemination	Knowledge Ontology and Codification
	Knowledge Application or Use	Knowledge of Early warning System
	Knowledge Acquisition	Knowledge Alert and Notification
	Knowledge Dissemination	Used the knowledge
	Knowledge Application or Use	
	Knowledge Embodiment	Knowledge Retrieval Knowledge Indexing

**5. RESULT AND DISCUSSIONS**

As result of our study and as what been discussed in our methodology section, the conceptual system design proposal has been evaluated based quality factor of services (QoS) of CoP agreements such as save time and faster to develop the application as a product to the specific user. Figure 5 has shown the agreement by the practitioners and most of them have given a good signed of implementing the system in their environment.

Meanwhile, Figure 6, Figure 7 and Figure 8 have also shown that the most regular types of knowledge application, urgency level of KM functionality requirement and it benefits by the practitioners or CoP based on their urgency of KMS system functionality in supporting knowledge sharing of agile system development environment.

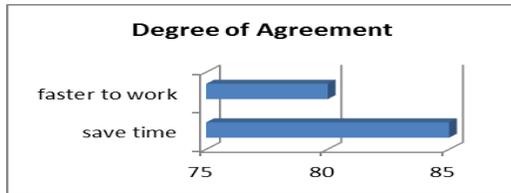


Fig. 5: The Agreement of Quality of Services by CoP

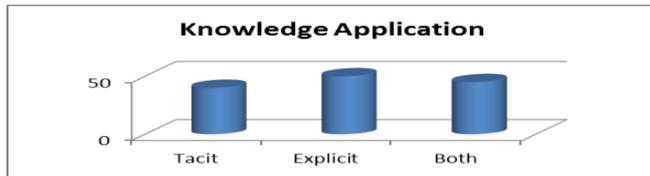


Fig. 6: The Most Regular Type of Knowledge Shared by CoP

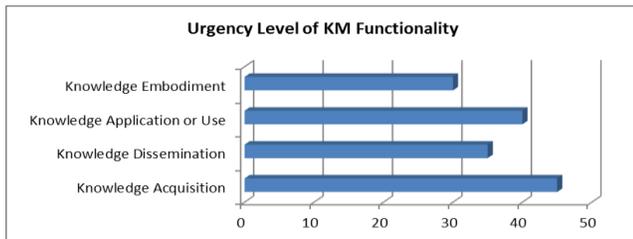


Fig. 7: Urgency Level of KMS functionality for KS-LEF

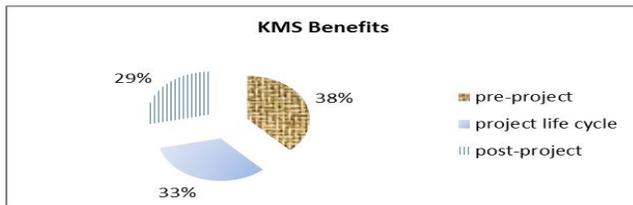


Fig. 8: KMS benefits as KS-LEF of Software Development Environment

## 6. CONCLUSION AND FUTURE WORK

As a conclusion, in the context of knowledge sharing process among the software engineers as a model toward generating conceptual system model, KM process which consists of knowledge acquisition, storage, dissemination and application is a good way or technique that is needs to be considered in agile software development (SD) environment. The conceptual design model called KS-LEF also discussed in term of its functionality that to be utilized towards beneficial or impact of the community of practice (CoP) in agile SD of the future. Therefore, there is a need of a model that focuses on lesson learnt and experience-based factory which is related to agile system development (SD) in becoming as a standard guidance to those who are involved in software development environment. Furthermore, by describing all those functionalities and non-functionalities requirement of the application such as security and system performance, so that it would be covered of all aspects of services especially related to agile SD requirement, which is providing the efficient and effective of services or also called as quality of services (QoS) for community of practice (CoP) in a particular environment.

## REFERENCES

- [1] Ambler, S., Lessons in Agility from Internet-Based Development, *IEEE Software*, vol. **19**, no. 2, 66-73, 2002.
- [2] Ambler, S. W., 2002, *Introduction to Agile Modelling (AM)*. Available: <http://www.ronin-intl.com/publications/agileModeling.pdf>.
- [3] AM Talib, R Atan, R Abdullah, and M Azrifah , 2011, CloudZone: Towards an integrity layer of cloud data storage based on multi agent system architecture, Open Systems (ICOS), *IEEE Conference on*, 127-132, 2011..
- [4] Sharp, H., Robinson, H., and Petre, M. The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers*. 108-106. (2009).
- [5] Beck, K., Cockburn, A., Jeffries, R., and Highsmith, J. 2002. Agile Manifesto, <http://www.agilemanifesto.org>
- [6] R Abdullah, S Sahibudin, RA Alias, and MH Selamat, Applying knowledge management system with agent technology to support decision making in collaborative learning environment, *Journal of American Academy of Business*, **7** (1), 181-188, 2005,
- [7] Schwaber, K. and Beedle, M., *Agile Software Development with SCRUM*. Prentice-Hall, 2002,
- [8] Turk, Dan, France, Robert, and Rumpe, Bernhard, 2002. Limitations of Agile Software Processes, in Proc. 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering-XP2002, Available: <http://www4.informatik.tumuenchen.de/~rumpe/ps/XP02.Limitations.pdf>.
- [9] McAdam, R. and McCreedy, S., A Critical Review of Knowledge Management Methods, *The Learning Organization*, **6.3**; 91-100, 1999.
- [10] J. Rising, Agile software development process, managed mayhem, 2009. <http://www.managedmayhem.com/2009/05/06/agilesoftware-development-process/>
- [11] Wikipedia 2013. – *Agile Software Development*, Accessed by Jun 2013.
- [12] AM Talib, R Atan, R Abdullah, MAA Murad, , Towards New Data Access Control Technique based on multi agent system architecture for cloud computing, 2011.
- [13] Modi Lakulu, R Abdullah, MH Selamat, H Ibrahim, MZMN Mohd Nor, A Framework of Collaborative Knowledge Management System in Open Source Software Development Environment, *Computer and Information Science*, **3** (1), P81, 2010,
- [14] M Abdellatif, AB Sultan, M Jabar. R Abdullah, A Technique for Quality Evaluation E-Learning from Developers Perspective, *American Journal of Economics and Business Administration*, **3**, (1), 157-164
- [15] R Abdullah, KT Wei, Usability Measurement of Malaysia Online News Websites, *International Journal of Computer Science and Network Security* **8**(5), 159-165, 2008.