An Ontological Model of Experience-Based Knowledge Management in Agile Software Development Environment

Nik Sakinah Nik Ab Aziz and Rusli Abdullah

Abstract—Agile software development (ASD) involves more in discussion and communication between team members compared to documentation; it is difficult for team members to share knowledge among them. In this matter, knowledge management (KM) plays critical role in managing and capturing the knowledge especially experience-based knowledge. This research main purpose is to create ontology to represent the knowledge that has been stored in knowledge repository after capturing the experience knowledge from the development team in ASD. This ontology model will help by describing a relationship that will later on help in sharing the experience knowledge easily among Community of Practice (CoP). A literature review of the common things in the ASD been conducted, we then creating ontology model by using a tools called Protégé. This ontology called as Experience-Based Knowledge Management (EBKM) Ontology Model. The model is deployed to apply to the reality as a prototype to demonstrate its real value.

Index Terms—Agile software development, knowledge management, ontology, experience-based factory, community of practice.

I. INTRODUCTION

Agile have a lot of methodologies that helps in various part of ASD process. Some examples of the methodologies are Extreme Programming (XP), Scrum, Dynamic Systems Development Method (DSDM), and Feature Driven Development (FDD). Each methodology has its own goal to achieve in ASD process. Since ASD involves more in discussion and communication between team members compared to documentation, it is difficult for team members to share knowledge among them. Hence, the knowledge in the development team should be captured and manage so that they can share the knowledge or information easily. In this matter, knowledge management (KM) plays big role in managing and capturing the knowledge especially experience-based knowledge. KM in software development (SD) is not a new topic, it been discuss many years back on experience-based knowledge. Knowledge Management (KM) plays big role in managing and capturing the knowledge especially experience-based knowledge by creating knowledge factory. By using this kind of factory in SD, it will encourage developers to share knowledge and work together in developing software especially which related to agile software development (ASD) environment [1]. Ontology helps to share common understanding of the knowledge, ease changes in domain knowledge (add, delete class / instance) [2].

This research main purpose is to create ontology to represent the knowledge that been stored in knowledge repository after capturing the experience knowledge from the development team in ASD. This ontology model will help by describe a relationship that will later on help in sharing the experience knowledge easily among Community of Practice (CoP). One of the objectives of this research is to propose an ontology model experience-based of knowledge management in ASD. A literature review of the common things in the ASD been conducted, we then creating ontology model by using a tools called protégé. In this ontology, all classes and properties are defined based on the findings. This ontology called as Experience-Based Knowledge Management (EBKM) Ontology Model. The other objective is to develop a prototype based on the ontology model proposed. The ontology was evaluated based on the usefulness in practice which are the ontology helped in saving development time and reduce development cost (such as resources, expenses, and tools) as stated by R. Abdullah et al. [1]. The model is deployed to apply to the reality as a prototype to demonstrate its real value. This research only focused on agile software development based on XP and Scrum methodology. The ontology is created to describe the connection between each practice in ASD.

II. LITERATURE REVIEW

A. Agile Software Development

Agile SDLC is a combination of iterative and incremental process models which focus on process adaptability and customer satisfaction by rapid delivery of working software product [3]. The author conclude that in each of the iterations, consists of planning, requirement analysis, designing, building and testing. Sheetal Sharma et al. [4], the agile process follows the software development life cycle which includes requirements gathering, analysis, design, coding, testing and deliver partially software, and feedback. In the whole process, customer satisfaction is the higher priority. There are several of agile methodologies that can be used in agile software development. Cockburn and Highsmith explain what is new about agile methods is not the practices they use, but their recognition of people as the main driving force which can lead to project success [5]. Many researchers
claimed XP and Scrum methodologies are widely used in the industry and are the most popular (e.g. [6]-[8]). The details of XP and Scrum practices are given by K. Beck [9] and K. Schwaber, and J. Sutherland [10].

B. Knowledge Management

The categorization of knowledge is managed based on agile development process as what been discussed by Graeme Smith et al. [11]. 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]. A new access control is named as Formula-Based Cloud Data Access Control (FCDAC) has been introduced to manage knowledge in cloud computing [12]. Modi Lakulu et al. [13], discuss the formulation of 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.

C. Knowledge Management and ASD

KM and Agile Software Development (ASD) are two organizational processes that face common barriers when introduced and applied in software development. The main barrier in initiating the product development in agile software development and implementing KM into software organizations is the need to deal with the conceptual change, mainly the organizational cultural change that ASD and KM brings when introduced [14]. Many studies have revealed that the introduction of KM and ASD processes have increased productivity, shortened time-to-market and resulted in higher product quality (e.g. [15], [16]). Boehm and Turner [17] note that agile methods rely on tacit knowledge and it depends on the ability to cultivate and share it. The pairing of KM and ASD is not new; a connection between the two concepts has been recognized by various researchers [18]-[22].

D. Ontology

There are several ontology languages available such as Resource Description Framework (RDF) [23], Web Ontology Language (OWL) [24], DARPA Agent Markup Language (DAML) [25], Ontology Interchange Language (OIL) [26], DAML+OIL [27], Simple HTML Ontology Extensions (SHOE) [28] etc. for capturing knowledge of interest. Different ontology languages have different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C) (http://www.w3.org/). It has the most complete set of expressions for capturing the different concepts and relationships that occur within ontologies; therefore, the software engineering knowledge is captured in OWL [29].

E. Ontology Model in Software Development

Relationship between ontologies and agile methods always appears in literature. Knublauch [30] believes that by using the correct tools, ontology given development can be applied more in agile development. However, there is no generic ontology have been proposed. Reflect to the previous paper, David [31] has proposed an ontology model. The researcher use the ontology approach to analyzing the core component of an agile methodology based on an existing literature that related to agile methodology. David believes this will assist our understanding of the kernel of Software Engineering (SE) theory that underlies within agile methodology. Pornpit Wongthongtham et al. [29] has created an ontology based on UML diagram to describe each phases in software engineering cycle generally. They use class diagram, entity relationship diagram, activity diagram and some use case diagram to describe the ontology. From the model, the researchers develop a prototype to visualize the ontology. As a result, it shows the ontology shows it is useful in practice.

III. METHODOLOGY

In order to create a model of knowledge sharing, there are few steps that have been taken, as shown in Fig. 1 below.

Our research is start by done some literature review to find out the common things in ASD and KM for Agile practices. By analyzing the collected data, we came out with the result. The results from the literature review then reviewed by some expert such as system engineer and developer in agile development environment who has more than 2 years of experience in ASD. We finally created the model that proposed named Experience-Based Knowledge Management (EBKM) ontology model. This model develops by using Protége 4.

The prototype was developed in order to evaluate the model and also to help user to understand how this ontology helps in sharing the knowledge between team members. This prototype was developed by using PHP language and MySQL database with Apache service. We then run a survey to a CoP (scrum master, developer, project manager, and tester) that involved directly in agile software development by distributing the questionnaires.

There were about 25 respondents (14 man and 11 women) that involved in the survey. The respondents mostly have agile development experience around two years and above. The data analysis was done based on the collected data from this survey. The result was analyze based on the usefulness of the model either it helps in saving time and reduce cost of the software development especially in agile.

A. EBKM Ontology Model

Manjula Shenoy et al. [32] presented the scope of ontology
mapping, how the ontology mapping processed and steps involved. It compares the features of various ontology mapping systems and the language, strategy used in each approach and their advantages and disadvantages. By refer to the articles written by A. Rabiyathul Basariya and O.S Jannahath Nisha [33], we create the ontology model. This paper was creating ontology model for education domain. The model then will be used as a reference to link information in the university website to get the relevant information. The steps creating the Experience-Based Knowledge Management (EBKM) ontology model are as Fig. 2 below:

B. Define Class

Based on literature review and expert opinion, we have five classes which are planning, analysis, design, implementation and feedback. The subclass are user stories, onsite customer, sprint plan meeting that consist of product backlog and sprint backlog, pair programming, collaborative workspace, daily scrum meeting and collective ownership. All the classes are coming under <owl:Thing> class [33].

C. Define Object and Datatype Property

Properties are included both object property and datatype property. Object property relates object to other object. For example the property usePairProgramming belongs to the class Pair Programming. Data type properties, which relate objects to datatype values. Each datatype properties have the range that specifies the type of data. Property has certain domain. Domain indicates in which class the property belongs to [33]. For examples, in user stories domain it have few characteristic which are having project id which project this story belongs to, use story id, user story title so developer and customer can refer to it easily, the description what should be in the system for that user story and etc.

D. EBKM Ontology Model

Various formalisms have been developed for modelling ontologies, notably the Knowledge Interchange Format (KIF) [34] and knowledge representation languages descended from KL-ONE [35]. However, these representations have had little success outside Artificial Intelligence (AI) research laboratories [36], [37] and require a steep learning curve. In this study, graphical notations of modelling agile software development ontology as an alternative formalism are presented. The main aim is not only to create a graphical representation to make it easier to understand, but importantly, this model should be able to capture the semantic richness of the defined agile development ontology [36]. The ontology model of EBKM is shown in Fig. 3 below.

E. EBKM System Prototype

This prototype only covers some part of the ontology. We choose the most important part to show to the CoP which is user stories. This prototype shows how ontology helps in sharing the development knowledge. Fig. 4-Fig. 6 below are some of the web page that shown how the knowledge is shared. The evaluation of EBKM ontology model is through the deployment ontology on the platform.
ontology was evaluated based on all related practices are inserted and the relationship between them are what they used in the development environment.

In Fig. 7 shows 80% of the respondent agreed that EBKM ontology consists of all the practices used in ASD.

![Agree that the model is right](image)

Fig. 7. Element in EBKM ontology model are related to ASD.

Next evaluation was either this model can be used to them as a reference while developing a system. Fig. 8 shows that 80% agreed that this model can be used as reference for them to start development using agile. This been evaluated based on prototype that was developed which is based on saving time (84% agree) and reduced development cost (84% agree). This result is illustrated in the Fig. 9 and Fig. 10.

![Model can be use as reference](image)

Fig. 8. EBKM use as reference.

![save the development time](image)

Fig. 9. Ontology help in save development time.

![Can reduce development cost](image)

Fig. 10. Ontology help in reduce development cost.

Nevertheless, the prototype proved that the model helps CoP to share knowledge among them more easily. Most of the respondent agreed all the required data that need to be shared are displayed in the prototype which was about 82%.

V. CONCLUSION AND FUTURE WORK

ASD is more focus on communication and discussion rather than documentation. The issue that always been discuss by CoP is how they want to capture the experience knowledge that gain by the developer. Since then, lots of proposed solutions have been introduced to capture the data. The other related issue concern by CoP is how to easily share the collected data that been stored in repository easily among software developers to save time for development and also to reduce cost of the development.

We proposed EBKM ontology model which later on can be used to develop a web. We develop a prototype as overviews to a user how the web works based to the ontology model proposed. As the result, 84% of the respondent agreed that this proposed model help them in saving time for software development and also reduce for cost in developing the system.

In conclusion, this proposed model is useful for the CoP to refer to for them to share knowledge that will save time and reduce cost of development. There are many improvements that can be made through future work which could consider agile software development ontology evolution. For future work, we hope that a framework on how to develop a full EBKM system can be done. So that everyone can refer to the framework to create a web to share the knowledge among CoP.

REFERENCES


Nik Sahinah Nik Ab Aziz was born in Terengganu, Malaysia on 1986. She received the B.S degree in computer science from the Universiti Tekniskal Malaysia Melaka (UTeM), Malaysia in 2009.

She has four years experiences of system development as a System Analyst from one of the software house in Selangor. Her profession focus covers the area of software development cycle in developing a system. Currently, she is working towards the MS Degree in Software Engineering from Universiti Putra Malaysia (UPM), Malaysia. Her project and research interest include studying the knowledge management and agile software development.

Rusli Abdullah is an associate professor in Information System Department, Faculty of Computer Science and Information Technology, of Universiti Putra Malaysia. He holds a B.Sc in computer science from Universiti Putra Malaysia in 1988, M.Sc in computer science from Universiti Putra Malaysia in 1996, and PhD in knowledge management at Faculty of Computer Science and Information System at Universiti Teknologi Malaysia in 2005.

He has more than 12 years of teaching experience and about 8 years of system development experience as a system analyst at higher learning institutions. He currently teaches System Analysis and Design, and Software Quality at both undergraduate and graduate levels. His research interests include knowledge management and software engineering, computer supported collaborative of work, and workflow management.

He has also published and written books, articles, and technical papers in numerous journals and conference proceedings with regards to his research interests. Some of his published books in Malay version are Introduction to Workgroup Computing, Introduction to Analysis and Design Systems, and Information Technology and Its Applications.