Towards Developing Experience Based Factory Model For Software Requirement Engineering Process In Collaborative Environment

Mastura Hanafiah¹, and Rusli Abdullah²
¹Universiti Putra Malaysia (UPM), mastura.hanafiah@daimler.com
²Universiti Putra Malaysia (UPM), rusli@fsktm.upm.edu.my

ABSTRACT
Requirement engineering is a crucial process in software development. It is known that requirement related activities are repetitive, time-consuming and requires a lot of human involvement. Furthermore, inadequate requirement management and inappropriate requirement engineering process are among the common issues in requirement engineering. Knowledge and experiences in requirement engineering should be managed systematically and efficiently for the purpose of sharing and reuse. This paper discusses the proposal for the development of experience based factory model for requirement engineering process in collaborative environment. The proposed model, namely Requirement Engineering for Experience Based Factory Model (RE-EBF-Model), has the objective to manage requirement knowledge effectively to maximize the benefit of knowledge sharing among the community of practice. A comprehensive study of literature review and experts’ opinion from a preliminary survey serve as the guideline for the model development.

Keywords: requirement engineering, knowledge management, experience based factory, collaborative environment

1 INTRODUCTION
Requirement engineering (RE) is an important field in software engineering. It is a process is to discover the purpose and needs of stakeholders, analysis and documenting for future needs and subsequent implementation. RE consists of several activities: elicitation, analysis, specification, verification and management (Dorfman, 1997). Improper RE process always causes problems in software development which will further affect the cost and project schedule. Firesmith (2007) has identified twelve common problems in RE. Two of them that are of the interest of this research are: (i) inadequate requirement management, and (ii) inadequate requirement process (Firesmith, 2007). Inadequate requirement management means that the requirements are not stored properly in one media location, not centralized and it causes problem in maintaining and manipulating the requirement knowledge, further, making knowledge acquisition more difficult. Inadequate requirement process means that the requirements are not documented properly or missing, and RE methods used are too generic and not properly tailored to project specific, thus the results are not optimal. These problems will have negative consequences which ultimately cause budget and schedule overrun, with products that may lack of capabilities and has additional defects.

In order to overcome these problems, software experience knowledge, both tacit and explicit, should be preserved and made accessible. Explicit knowledge is the knowledge that can be written down and relatively easily transferred from one person to the next, while tacit knowledge on the other hand is more difficult to articulate because it often arises out of experiences (Nonaka and Takeuchi, 1995).

In this research, we propose a model, Requirement Engineering for Experience Based Factory Model (RE-EBF-Model), which is mainly based on the following questions: (i) what are the best RE methods and practices currently employed by the software organizations?; (ii) how can we manage the RE knowledge systematically for sharing and reuse?; (iii) how do we allow the community to work collaboratively and dynamically to maximize the RE knowledge sharing?

The following sections discuss the analysis on the literature review, the methodology, the proposed model, and the discussion from the preliminary study.

II LITERATURE REVIEW
A. The Experience Factory
Organizations have attempted to preserve and transfer knowledge and experiences in many ways. It is believed that experience needs to be packaged and evaluated for reuse potential. The needs to manage experience are derived from several reasons (Basili et.al, 2001). First, it needs to become less dependent on its employees to prevent loss of knowledge; second, it needs to unload its experts by eliciting and storing experts’ experience and make it available; third, it needs to create productive employees sooner by speeding up the learning curve; and finally, it needs to improve the business process by analyzing and
synthesizing experiences, and further make it captured, structured and available.

Experience Factory (EF) makes use of reusing products, processes and other forms of knowledge from system lifecycle in order to improve system development at a lower cost (Basili et.al, 2009). Figure 1 shows the EF basic steps which are based on Quality Improvement Paradigm (QIM) with six elementary steps: Characterize, Set Goals, Choose Process, Execute, Analyze and Package.

Figure 1. The Experience Factory (Basili et al., 2009).

Basili et al. (2009) further emphasize that EF is a logical and/or physical organization that supports project developments by analyzing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand. It packages experience by building informal, formal or schematized, and productized models and measures of various software processes, products, and other forms of knowledge via people, documents, and automated support.

B. Knowledge Management

Knowledge management (KM) is the process of capturing, developing, sharing, and effectively using organizational knowledge effectively with the aims to create value and increase or sustain competitive advantage for the organization. Researchers have argued that KM requires a broad base of activities, including conversion of data and text into knowledge, conversion of individual’s and group’s knowledge into accessible knowledge, connection of people and knowledge to other people and other knowledge, communication of information between users, collaboration between different groups, and creation of new knowledge (O’Leary, 1998b; O’Leary, 1999).

For collaborative KM system, four main basic processes are emphasized: acquisition, storage, dissemination and application (Abdullah and Selamat, 2007). Knowledge acquisition is the process of identifying, collecting, adapting, organizing, and storing the knowledge. Knowledge storage is the organizational knowledge in repositories, either in the form of documentation or in a special format to enable future browsing and fast access. Knowledge dissemination is where knowledge is published and shared among the users of KM system. There are four techniques to disseminate knowledge: synchronous technique (same time, same place), asynchronous techniques (different time, same place), distributed synchronous collaboration (same time, different place) and distributed asynchronous collaboration (different time, different place) (Abdullah, 2008). Piktialis and Greener (2008) stated that knowledge can be made in public in two ways: the “push approach” where the knowledge is published such as on websites, intranets and newsletters, and the “pull approach” where it involves facilitated sharing between sources and receivers. Finally, knowledge application is facilitated through KM system that provides the platform to document, store and share knowledge for the purpose of learning, decision making or problem solving.

C. Knowledge Management In Requirement Engineering

RE is a complex and knowledge-intensive activities and it involves many stakeholders from different background working in different phases and activities (Maalej and Thurimella, 2013). Further, the researchers emphasized that requirement knowledge is diverse and continuously expanding and a systematic way of managing and treating the knowledge valuable assets could help organizations leverage the knowledge they possess. It is therefore essential that requirement knowledge should be managed to benefit the community of practice (CoP). Maalej and Thurimella (2013) have summarized several benefits of managing requirement knowledge: it improves understandability of requirement and reduces any mismatch; it helps to identify new requirements from knowledge of previous projects; it solves repetitive RE problems; it speeds up decision-making; it helps RE knowledge to evolve; and it improves traceability and identifies hidden interdependencies.

Managing requirement knowledge requires some techniques like data mining, information retrieval, ontology-based and agent technology to enable efficient capture, access and sharing of RE knowledge. Intelligent tool support for RE should have the ability to reason about knowledge, derive new knowledge, or deal with incomplete and scattered knowledge (Maalej and Thurimella, 2013).

ReqWiki, a novel open source web-based approach was developed by Sateli et al. (2013), is an example of a KM system for RE to support the growing needs of managing requirement. It is based on a semantic wiki that includes natural language processing (NLP) assistants, which work collaboratively with humans on the requirements specification documents. ReqWiki formalizes the generated semantic metadata by users.
in a form of ontology and can be exploited in many ways such as creating complex traceability relationships.

D. Ontology in Requirement Engineering

The term “Ontology” is derived from its usage in philosophy where it means the study of being or existence as well as the basic categories (Fensel, 2001). Castañeda et al. (2010) defines ontology as an explicit formal specification of how to represent the entities that exist in a given domain of interest and the relationships that hold among them. The use of ontologies in RE is believed to reduce the negative factors in RE process such as unambiguous, insufficient and incomplete requirements, also to manage dynamic and changing requirements. Some potential uses of ontologies in RE include representation of: (i) the requirements model, imposing and enabling a particular paradigmatic way of structuring requirements, (ii) acquisition structures for domain knowledge, and (iii) the knowledge of the application domain (Castañeda et al., 2010).

Jurisica et al. (2004) stated that ontologies may be constructed for different purpose; thus, four categories of ontologies are defined: static, dynamic, intentional and social aspects of the world. Static ontology describes things that exist, their attributes and relationships; dynamic ontology describes the world in terms of states, state transitions and processes; intentional ontology encompasses the world of agents, things agents believe in, want, prove or disprove, and argue about; and social ontology covers social settings, permanent organizational structures or shifting networks of alliances and interdependencies. Several works on RE ontologies have been carried out by previous researchers. Castañeda et al. (2010) proposes an RE ontology framework as Requirement Ontology, Requirement Specification Document Ontology, and Application Domain Technology, while Dillon et al. (2008) has proposed several ontology types related to RE: Requirements Sub-ontology, Requirements Elicitation Sub-ontology, Requirements Analysis Sub-ontology, and Requirements Specification Sub-ontology.

Many knowledge ontology-based systems have used an open source ontology editor and framework, Protégé (Protégé, 2014). Protégé’s plug-in architecture can be used to build both simple and complex ontology-based applications which may help in the rapid prototyping and development of ontology-based application.

E. Agent Technology in Knowledge Management

Software agents have been widely used as part of KM solution. Agents can be viewed as a computer program with the following capabilities: communication for the purpose of co-operation and negotiation, learning to improve performance over time and autonomy to react proactively (Abdullah, 2008). Agents also act as the communicator between the system and the user by receiving instruction and producing the expected result.

Agents, such as Multi Agent Systems (MAS), are believed to solve more complex problems by employing a collection of agents that are collaborated in a given domain (Dillon et al., 2008). These agents usually have a small knowledge base with a specified intelligence that collaborates with other agents to ensure the consistent and coherent knowledge based, also it facilitates the communication and coordination between the agents. MAS is also adopted in the creation of collaborative KM system for software maintenance to enable users and software maintainers to capture and share the enterprise business domain knowledge (Mohd. Nor et al., 2009). Nitto et al. (2002) provides some example of MAS in KM: searching agents to perform search in over large repositories; user agents to represent user activities and guide users’ interaction with the system; profiling agents that manage the information about user profiles; and filtering agents that can decide which data is relevant for the user.

F. Knowledge Management in Collaborative Environment

Many researchers have integrated KM in such a collaborative environment to maximize its advantage. Whitehead (2007) points out that working collaboratively in software engineering has several goals: to establish project scope and capabilities; to drive convergence towards a final architecture and design; to manage dependencies among activities; artifacts and organizations; to reduce dependencies; to identify, record and resolve errors; and finally to record organizational memories.

Lakulu et al. (2010) has formulated a framework for KM system in collaborative open source software development with five main components: layers, components, process, knowledge and CoP. The model enables the CoP to assess the knowledge content with four conversion patterns based on Socialization, Internalization, Combination, and Externalization (SECI) model by Nonaka and Takeuchi (1995). The process component makes use of basic KM process: acquiring, organizing/storing, disseminating, and using knowledge with functionalities including a portal, collaboration tools, OS infrastructure, technologies and repositories.

For community of higher learning institution, a model has been proposed to aid the facilitation in terms of collaboration among the individuals in a faculty with the extensive collaborative functionalities such as document sharing, correspondence handling and
tracking system, group/individual email and calendar, discussion boards, and many more (Abdullah and Selamat, 2007). Lotus Notes has been used to facilitate knowledge acquiring and dissemination with the groupware technologies such as email, bulletin boards, video-conferencing, discussion room, etc. These are among the many features that can benefit the CoP and further improves the communication gaps.

III METHODOLOGY
The research is initiated with the motivation to manage requirement knowledge for sharing and reuse among the community of practice. Figure 2 shows the methodology applied in this research.

IV PROPOSED MODEL
Based on the analysis from LR and data collected from the preliminary survey, a high level model is formulated as shown in Figure 3. This model consists of four main components: Community of Practice (CoP), K-Interface, KM-Process and RE-Repository.

CoP (Community of Practice). CoP in this context are the peers, practitioners and other individuals that seek for the knowledge based to gain and understand the RE process for particular software projects. This includes all software stakeholders including project managers, software engineers, developers, architects, testers, support persons, system owners and higher management. The model will be accommodated with collaborative environment that allows the CoP to work together with shared understanding surrounding the knowledge.

K-Interface (Knowledge Interface). This is the interface that allows the CoP to access the requirement
knowledge. The representation should be lightweight (such as Wiki), but extensive that covers all necessary RE knowledge stored in the repository. Agents will be used to provide the required services such as search agents, user agents, profile agents, and filtering agents. The interface should have the capabilities for knowledge centralization, well-structured, collaborative and dynamic.

**KM-Process (Knowledge Management Processes).** This component involves four other sub-process: (i) K-Acquisition: a process to determine the sources and types of knowledge, gather, transform and categorize the knowledge according to specifications; (ii) K-Storage: a process to prepare, map, keep and index the knowledge; (iii) K-Dissemination: a process to publish and share knowledge within the CoP using these two way methods: “push approach” to publish the knowledge (e.g. in websites) and “pull approach” to share the resources; and (iv) K-Application: a process to acquire and apply knowledge for CoP in order to learn, understand, and evolve the requirement knowledge. Ontology-based coupled with agent technology will be used as the frameworks in developing these processes.

**RE-Repository (Requirement Repository).** The experiences and knowledge captured during RE activities serves as the basis source of this model. For each of the RE phase, a repository is maintained for identified tacit and explicit knowledge: (i) RE-Elicitation: the process of discovering, reviewing, documenting, and understanding the user’s needs and constraints; (ii) RE-Analysis: the process of refining the user’s needs and constraints; (iii) RE-Specification: the process of documenting the user’s needs and constraints clearly and precisely; and (iv) RE-Verification: the process to ensure the system requirements are complete, correct, consistent, and clear. The repositories should keep the best practices of RE, derived from the LR and detailed study performed with relevant software companies.

V DISCUSSION

As mentioned earlier, a preliminary survey has been conducted at the beginning of the research. The findings from LR and experts’ opinion are further analyzed. The study managed to collect the responses from 12 software practitioners. Among others are project managers (2), business analyst (1), software engineers (5), developers/programmers (2), software tester (1) and support persons (1). Over 80% practitioners involve in web-based projects and more than 50% practitioners involve in medium size projects.

Various techniques and approaches have been performed by the practitioners for requirement elicitation; informal modelling (25%), use cases (66%), user stories (41%), focus groups (25%) and scenarios (33%); while for requirement analysis, most of the practitioners has adopted object oriented analysis and design (66%). Table 1 shows the number of respondents’ on their satisfaction level on the current RE practice in their organizations with rating 1-very dissatisfied, and 5-very satisfied.

The average rating shows little satisfaction result (3.25). Additionally, about 50% of the respondents do not have ready platform for requirement knowledge management. Table 2 and Figure 4 show the number of respondents on their satisfaction level on several aspects particularly when taking over an existing project at the initial stage from previous teams (Q8).

The average rating falls between 3.00 and 3.25 for all aspects. This shows that the respondents’ satisfaction levels are somehow neutral and little satisfied.

![Table 1. Satisfaction on Current RE Practice.](image)

<table>
<thead>
<tr>
<th>Satisfaction on current RE practice</th>
<th>Very dissatisfied</th>
<th>Disappointed</th>
<th>Neutral</th>
<th>Satisfied</th>
<th>Very Satisfied</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>3.25</td>
</tr>
</tbody>
</table>

![Table 2. Result Survey Q8.](image)

<table>
<thead>
<tr>
<th>High level understanding</th>
<th>Very dissatisfied</th>
<th>Disappointed</th>
<th>Neutral</th>
<th>Satisfied</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements capturing activities</th>
<th>Very dissatisfied</th>
<th>Disappointed</th>
<th>Neutral</th>
<th>Satisfied</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Finding requirement documents</th>
<th>Very dissatisfied</th>
<th>Disappointed</th>
<th>Neutral</th>
<th>Satisfied</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RE Communication</th>
<th>Very dissatisfied</th>
<th>Disappointed</th>
<th>Neutral</th>
<th>Satisfied</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4. Result Survey Q8.

Figure 5 summarizes the responses on the organization’s need to have a centralized, structured, collaborative and dynamic environment for KM system for RE (Q9). For RE-EBF-Model’s proposed features (centralized, structured, collaborative and dynamic), more than 50% of the respondents agree or strongly agree that those attributes are important and essential for KM system for RE.

Finally, Figure 6 shows that most of the respondents agree (58%) or strongly agree (16%) that by having a centralized, structured, dynamic and collaborative knowledge management for RE, the software...
development process in the organization can be improved (Q10).

In summary, the results from this study illustrate that, in software organization, adequate requirement management and appropriate RE process are essential and should be formulated or refined in order to improve the software development process in general.

VI CONCLUSION AND FUTURE WORKS

The objective of requirement management is to capture, store, disseminate and manage the requirements’ explicit and tacit knowledge. Inappropriate requirement management will result in miscommunication and misinterpretation which will further increase the effort and cost in terms of transition and knowledge transfer. KM in RE is still an innovative research area. There are a number of KM tools for software engineering, but there is still lack of research on KM for RE. The contribution of this study is to propose a model on how KM can be used in RE in order to promote knowledge sharing for the benefit of CoP. The model, RE-EBF-Model is introduced which consists of several main components: CoP, K-Interface, KM-Process, and RE-Repository. Experience based factory elementary processes, coupled with agent and ontology-based frameworks will be employed to facilitate the development of the model. In future works, a more detailed study will be conducted to understand current RE processes and relevant measurement metrics such as efficiency, complexity, completeness, and usability will be evaluated.

REFERENCES


