

Towards Knowledge Management in RE Practices to Support Software Development

Mamoona Humayoun, Asad Masood Qazi

University Institute of Information Technology, PMAS-University Institute of Information Technology,
Rawalpindi, Pakistan

Email: mamoona@uair.edu.pk, asad.masood@dpskw.com

Received 12 June 2015; accepted 23 August 2015; published 26 August 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Requirement engineering in any software development is the most important phase to ensure the success or failure of software. Knowledge modeling and management are helping tools to learn the software organizations. The traditional Requirements engineering practices are based upon the interaction of stakeholders which causes iteratively changes in requirements and difficulties in communication and understanding problem domain etc. So, to resolve such issues we use knowledge based techniques to support the RE practices as well as software development process. Our technique is based on two prospective, theoretical and practical implementations. In this paper, we described the need of knowledge management in software engineering and then proposed a model based on knowledge management to support the software development process. To verify our results, we used controlled experiment approach. We have implemented our model, and verify results by using and without using proposed knowledge based RE process. Our resultant proposed model can save the overall cost and time of requirement engineering process as well as software development.

Keywords

Knowledge Management, Software Requirement Engineering, Software Development Methodologies, Knowledge Modeling

1. Introduction

1.1. Requirement Engineering

Software requirement engineering involves requirements elicitation, requirements specification, requirements validation and requirements management [1] [2]. Requirements elicitation involves the ways of gathering the

How to cite this paper: Humayoun, M. and Qazi, A.M. (2015) Towards Knowledge Management in RE Practices to Support Software Development. *Journal of Software Engineering and Applications*, 8, 407-418.

<http://dx.doi.org/10.4236/jsea.2015.88040>

requirements which include many traditional, cognitive, model based etc. techniques.

Whereas, the requirements specification (where analysis and negotiation of requirements are performed), and requirements of users are specified to make them understandable and meaningful for developers. Specifications can be formal as well as non-formal [3]. Formal techniques include the set of tools and techniques based on mathematical models whereas informal techniques are based on modeling the requirements in diagrams or making architecture of system. There are many and many techniques in both types of specification. Like in formal techniques of specifications, we have different formal specification languages like Z, VDM etc. and in In-formal or non-formal techniques, we have UML diagrams which include use-cases, sequence diagrams, collaboration and interaction diagrams etc.

In Requirements validation the completeness of the requirements is checked which means either gathered requirements are correct, complete or not. The main objectives are to analyze the validation and verification of RE process, to identify and resolve the problems and highly risk factors of software in early stages, and to make strengthen the development cycle [4].

Finally in requirements management phase, issues and conflicts of users are resolved. According to Andriole [5], the requirement management is a political game. It is basically applied in such cases where we have to control the expectations of stakeholders from software, and put the requirements rather than in well-meaning by customers but meaning-full by developers, so they can examine that, and actually full fill the user’s requirements.

Authors of [6] include the Requirement change management under the Requirement Engineering process. RCM is a term which is used according to the history or previous development of the similar software product(s). On the basis of historical development, we investigate the need of RCM or not.

Software Requirement Specification (SRS) is a standard and official document which describes what the developers will develop. It includes detailed requirements specifications [7]. A standard IEEE format of SRS [8] describes the recommendations for a good SRS document.

1.2. Knowledge Management in Software Requirement Engineering

Many definitions of Knowledge modeling and knowledge management can be found in literature by authors having different background. The one of them is: “A method that simplifies the process of sharing, distributing, creating, capturing and understanding of company’s knowledge” [9]. The summary of knowledge management in Software Engineering refers to this definition which says that the main purpose of knowledge modeling is to learn the software organizations. The more elaborated definition is: “A software organization that promotes improved actions through better knowledge and understanding” [10].

A knowledge model may contain multiple attributes. Many authors describe different knowledge attributes and factors. In [11], the six activities are been discussed which includes capturing, storing, sharing, learning, exploring and exploiting. In [12], the major four types of attributes are present in study which describes that there a knowledge model should describe the 4 types of knowledge which includes Domain Knowledge, Control Knowledge, System Knowledge and Explanatory Knowledge as presented in **Figure 1**.

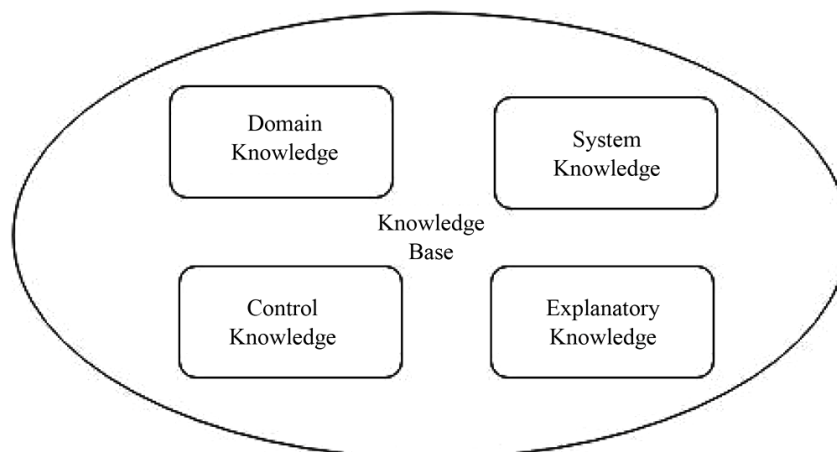


Figure 1. Attributes of knowledge modeling and management.

2. Related Work

Knowledge modeling is not a new area, a lot of work has been done using knowledge modeling techniques in last some decades. In software engineering, knowledge management techniques are used to provide an easy way to learn the software organizations. In software engineering there using of software development process or life cycle is used and it is called an “Experience Factory” in terms of Software engineering. In knowledge management, we collect these experiences from different resources like past papers, case studies etc. and store them into a single repository to easy reuse of software life cycle or development process [13].

In [14], we can see that the authors describe the knowledge based requirement engineering process which has to be implemented in the requirement elicitation phase. Authors define a way to use knowledge management on a strategy of Requirement engineering phase. In this paper the tacit knowledge is being converted into explicit knowledge to save additional cost as well as additional development time, which is described in [15]. Authors also claims that, there are many problems and situation in software industry, in which informal as well as formal techniques are not helpful, also there are no such algorithms defined which may help to tackle the solution. These create an ambiguity and inconsistency in the requirements, so we should focus on the knowledge management. Which can help the software engineers to obtain an acceptable solution?

By referencing [16], we can say that knowledge modeling and management can be helpful in software engineering by understanding the business needs correctly, better decision making process, accessing domain knowledge and sharing knowledge about local policies and practices.

In [17], authors have proposed a distributed and cloud computing based knowledge model as a service framework. This model will help to resolve the GSD (Global Software Development) issues. A central knowledge repository has been used as KM Server to store the knowledge in cloud environment.

In [18] a complete procedure to develop the knowledge model and run a complete knowledge based process has been described. The results of this case study have been implemented in some software organizations to observe the importance of knowledge modeling. Some activities of KM also has been described in this study, which includes the knowledge application, knowledge distribution, knowledge organization, knowledge adaptation, knowledge identification, knowledge acquisition and knowledge creation.

Authors in [19] argue that software process is essentially a knowledge process and it is structured in a knowledge management framework. In study [20] authors describes the issues faced by small and medium organizations by proper implementing knowledge management in Software Engineering area.

In [21], the authors describe the need of knowledge management in education of software engineering field. There is a big challenge still for organizations to develop a model to support effectively the knowledge management model. This paper also found some key factors which can make an impact on success of knowledge management practices in education.

A technology oriented knowledge model is been described in [22], which basically describes the complete procedure of sharing and creating knowledge in software engineering and software organizations.

An Ontology based knowledge management software development model has been proposed in the study given in [23]. Three types of packages have been described in this study, which are generic package, Ontology package and specific package. Authors basically try to analyze the instantiations in Ontology.

In [24], the authors defined a requirement engineering solution based on the natural language semantics solution. Natural language based solution can be helpful in the requirements discovery, requirements analysis and requirements specifications in form of requirements creation, requirements classification and requirements formalization.

In [25] authors identified the areas where knowledge management lies. In traditional software engineering, conceptual specifications and design phase were separate, but knowledge modeling bridge a gap between these phases. Also they talk about the generations of knowledge modelling in software engineering, which are:

In G1 the rules and frames are defined.

In G2 a symbolic representation is used to abstract the design phase. And then convert these abstractions into a model based system development.

A knowledge model can be based on the principle of task oriented, problem-solving and domain oriented. In task oriented the input and output of system is analyzed. Problem is then transformed in tree based architecture. On the other hand, in problem solving technique, a major task or goal is divided into sub tasks. Knowledge is transformed in hierarchical architecture. And in domain oriented, an Ontology is being used to explicit the do-

main elements.

KSM, a knowledge modelling tool is used to model the knowledge. In KSM the important thing is Knowledge area, which is further divided into supporting sub knowledge areas (also called supporting knowledge bodies) and their functionality. There are two languages used by the KSM to formulate the common terminologies and strategies of reasoning. Common terminologies are explained in Concel Language and the strategies are defined in Link Language.

It can also be observed from the study given in [26] that a major problem in requirement engineering is a communication, especially, when there is a communication between participants having different language and culture. So to tackle the issues, a knowledge modelling is used. A cycle/Spiral view of knowledge is proposed to solve the communication issue in RE Process. In this process the requirements are elicited and then specified. The elicited requirements are then tacit through socialization process. After elicited, the requirements are externalized or specified and after completion of circle the finalized requirements are documented, and unclear requirements again return back to specification stage.

In another paper given in [27], KMOS-RE is proposed, which is a knowledge model on a strategy to requirements engineering. In this proposed solution, the author(s) claims that the complexity of requirement engineering based on an application domain. In informal structured domain, the entities, functions, events and behavior of system is explained. However there are some issues in these informal structured domains which are symmetry of ignorance, situated actions and ambiguities. To solve such issues we use knowledge management in RE. A solution is based on the tacit and explicit. Tacit involves the socialization techniques like JAD, Interviews and externalization through knowledge representation and discourse analysis etc. And in explicit we have combination through data mining etc. and internalization through the knowledge management. Proposed KMOS-RE also defines the two phenomena's machine and world. In world phenomena side we have domain experts who have domain knowledge, and in the machine phenomena side, we have requirement engineers who have knowledge of software development. When both phenomena's are being gathered, with the help of software knowledge and domain knowledge, requirements specifications can be produced.

In [28] the author(s) emphasizes on the need of knowledge modelling in requirement engineering process. They claim that the requirement engineers should be able to understand the human's action in which the developed system will be implemented. Also, a requirement engineer should be able to understand the history which will help to develop project. Similarly requirement engineer should consider cultural norms of client (client's organization) and have an access to organization directories or social network to become known about the knowledge and its source.

To be able to above mention qualities, a requirement engineer may use the knowledge modelling. In this study, authors relate the requirement engineering with cognitive theories. As RE is a team work, so a process should be well defined, use of prototypes, use of common vocabulary for communication, supportive environment and address the issues in form of notations and symbols to resolve them separately [29].

A knowledge has four dimensions; current situation of environment, experienced knowledge, group knowledge and cultural knowledge [30]. So by focusing all these pre requisites authors of [28] compare the cognitive theories and examine the improvements in the requirement engineering process.

3. Proposed Work

Our proposed methodology is based on the existing model of Faxiang Chen, Burseitn F given in [11]. However, we have integrated this model by taking its attributes to just model the knowledge. Our main focus is on complete knowledge acquisition model which has proper semantic rules for elaborating the complete problem domain. The main steps of our proposed knowledge model are:

- Knowledge Learning Stage
- Knowledge Exploring
- Knowledge Capturing
- Knowledge Storing
- Knowledge Sharing
- Knowledge Exploitation

After these steps, we will merge our proposed model into a complete knowledge management model. Here, we did not involve the users or its basic requirements about the system, here we just focus on the techniques which we will use after getting the basic requirements from users. The input of this model can be a UML Dia-

gram as well as the requirements in descriptive form. The knowledge model is illustrated in **Figure 2**.

3.1. Knowledge Learning Stage

In proposed model, the activities which are being defined in this learning stage are helpful to learn about the problems. Employees can learn knowledge of domain, system, control and explanatory from past papers, historical related software problems, previous coding and development methodologies, books and case studies etc. In next section of experiment design, we are using the case study of Police station automation system, which were developed by university students. The experiment group, go through the whole learning stage by getting knowledge of manual system from their own resources like internet, columns, manuals, relatives etc.

3.2. Knowledge Exploring

In above mentioned proposed knowledge model, we have presented the knowledge exploring phase in shape of database storage. Basically, in knowledge exploring phase, we explore the existing knowledge or related knowledge using case studies to achieve competitive advantage. In our scenario, we have been search about different approaches of implementation of Police station system, and then we come to know about an easy approach using

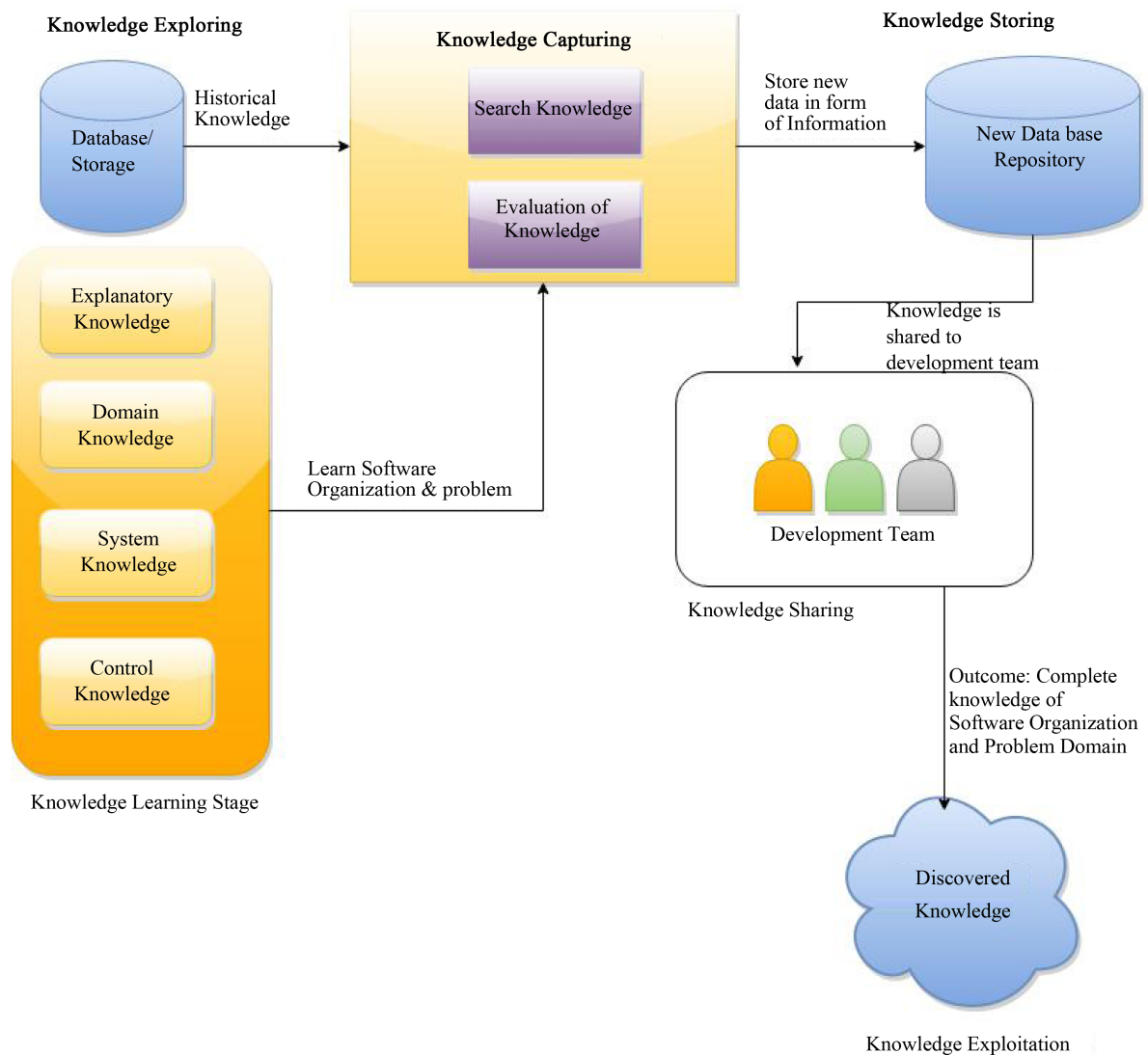


Figure 2. The proposed knowledge based model for requirement engineering.

knowledge model. Many automated systems or projects can be found on internet with named as E-Justice, Automated police station system etc. but our focus is to learn the implementation techniques to solve the situation according to our scenario and requirement.

3.3. Knowledge Capturing

The input of this phase is the knowledge which employees have learned as well as knowledge that has been explored related to software organizations as well as in our case about the software system, which has to be developed. In this phase we focus on “what”. What knowledge is required etc.? In our particular scenario of Police station automation system, we have identified the methods, roles and privileges etc. in this stage.

3.4. Knowledge Storage

The input of this phase is captured knowledge from learning and exploring stage. We store all the required knowledge in our database. Most researchers use RDBMS [31] in this stage to store the knowledge. In our case we have store knowledge related to police station automation system in database.

3.5. Knowledge Sharing

In knowledge sharing phase, tasks in form of knowledge are divided into the project team. In our implementation phase we suggest one student group member to make knowledge model of the police station system, he/she will store the knowledge in their particular database in form of textual or descriptive data as well as in form of UML diagrams. Similarly, a developer has implemented the police station automated system in Micro Soft Visual Studio 2013 using C# as programming language. And the third member will assist both members as well as make the project documentation. A sharing of knowledge can be presented in form of simple use case diagram to identify the user’s interaction with the system (see Figure 3).

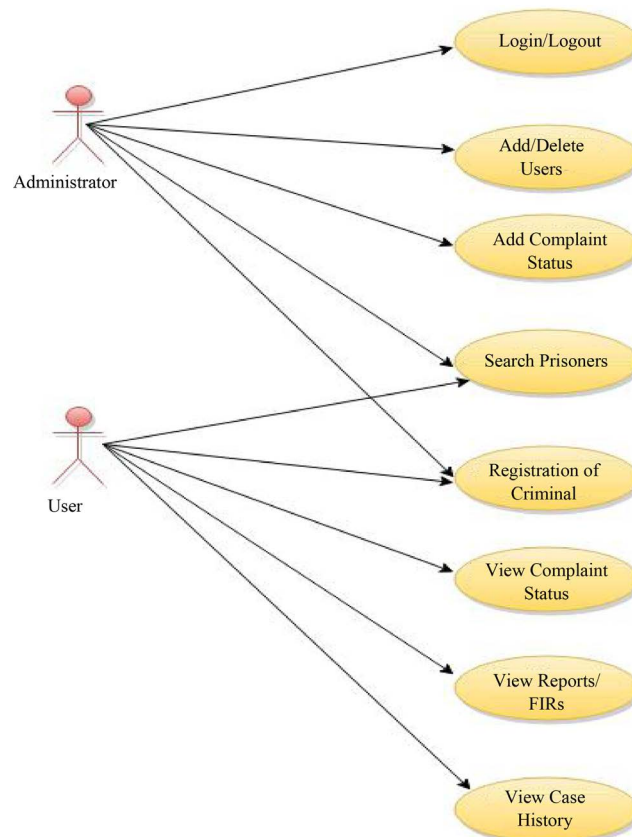


Figure 3. A simple use case diagram of police station automation system.

3.6. Knowledge Exploitation

In this last phase of knowledge model, the output in our case was a UML diagram of whole project. The group of students who develop this project previously, without any restriction of using RE or software development model was facing difficulties in this particular area, because they did not get any domain knowledge properly, and they were not clear about what to do.

4. Design of Experiment

For experiment design, the guidelines given by Kitchenham *et al.* [32] and Wohlin *et al.* [33] were used. In study of Kitchenham, we can see that the guidelines for experiment in Software engineering are being represented to analysis, data collection in a brief way, not in a detailed. However, Wohlin *et al.* uses some statistical techniques in experiment to analyze the results in more detailed form.

4.1. Goal and Variables

The goal of this experiment is to empirically investigate the proposed knowledge model. By implementing it on a particular case study, which we can analyze its maximum features and attributes in the first stage. Our experiment based on small project named as Police Station Automation system, of graduate students.

During experiment, we have identified following variables (see [Table 1](#)).

4.2. Participants

As we have two groups (Controlled and Experiment), so in each group we have 3 students, each group has 1 supervisor or expert and 1 reviewer. Participants are responsible to develop a program, documentation as well as testing. Because the basis of all steps, we can analyze the proposed model (its complexity, benefits and obstacles).

4.3. Objects

The software system which is being developed during this experiment is named as “Police Station Automation System”. Each group is restricted to use same development platform Microsoft Visual Studio 2012 and Programming language C# with MS SQL Server as database storage.

4.4. Division of Participants

The division of participants, according to the above mentioned criteria for both groups (controlled as well as experiment) is given below.

4.5. Questionnaire from Participants

To check the correctness and verification, we have designed a questionnaire for all participants of both groups as well as their supervisors. The questionnaire contains following questions (see [Table 2](#)).

5. Execution of Experiment

To execute the experiment we have assigned the tasks to two groups of students. One group is performing the

Table 1. Identifying variables of experiment.

Sr. #	Description	Unit	Possible Values/Range
1	Time Taken by Group/Individual	Hours	20 Hours.
2	Programming Experience	Months	12 months (Max)
3	Group	-	Controlled/Experiment
4	Affiliation	-	Academia, Industry
5	Architecture/Design Experience	Months	12 months (Max)

Table 2. Questionnaire to be filled by participants of both groups (control and experiment).

Sr. #	Question
Q1	Does the proposed model complete the correctness?
Q2	Is the knowledge management model full fill ease of use property for all stakeholders?
Q3	Is this Flexible?
Q4	Does it hold enhancement of Usability property
Q5	Is the proposed model is also good for the understanding of User's? (User's Satisfaction)
Q6	Is Knowledge Management complex?

role of control group and the other is experiment.

5.1. Sample and Preparation

As said earlier, there are two groups of students of University located in Rawalpindi, Pakistan. To analyze the results, there are 3 members and 1 supervisor (Manager/Professor). In figure **Chart 1** and **Chart 2** we have mentioned the experience and knowledge of participants. Further, we allow control group to develop a software by using their existing knowledge and experience and collect the requirements from either station or by else. But the experiment group is restricted to follow the complete proposed knowledge management model to elicit the requirements. Knowledge management helps the participants in software reuse, software maintenance, evolution and quality control of software system.

5.2. Data Collection Performed

Data collection procedure was performed as it is defined in the section Design of Experiment. Questionnaire is filled by all the participants after the completeness of software development. No participant was left to fill the questionnaire and to give feedback about the proposed knowledge management practices.

5.3. Validation of Results

To ensure the validity of results, we conduct the experiment in a controlled environment. No participant was allowed to leave the room/Lab before completion of Questionnaire. Also, as mentioned earlier, both the developments were made one by one not at the same time or at the front of each other. Group 1 was unknown about the Group 2 activities and vice versa. Also, groups were designed in such a way, so they were not aware about the experiment. Basically it was a surprised experiment.

The results of questionnaire of both groups are mentioned in **Chart 3**.

6. Result Analysis

We analyze the results of experiment by using following steps:

6.1. Descriptive Statistical Analysis

We conduct two developments of same project from different group. One is control group and other is experiment group. After complete the requirement gathering and completion of software development we can analyze the results in a statistical way: (see **Table 3**).

Now we will analyze the results according to the experience of the participants. Our hypothesis of this step is: The results will shows that the experience of participants is a main factor in software development, but by using proper methodology we can overcome the minor differences in experiment in development (**Table 4**).

The experience of participants have small variation because some students usually starts internship during the studies. That's why some participants have affiliation with industry and others.

6.2. Analysis on Opinion of Participants

As mentioned above, after the development the software system, participants were asked to fill the questionnaire

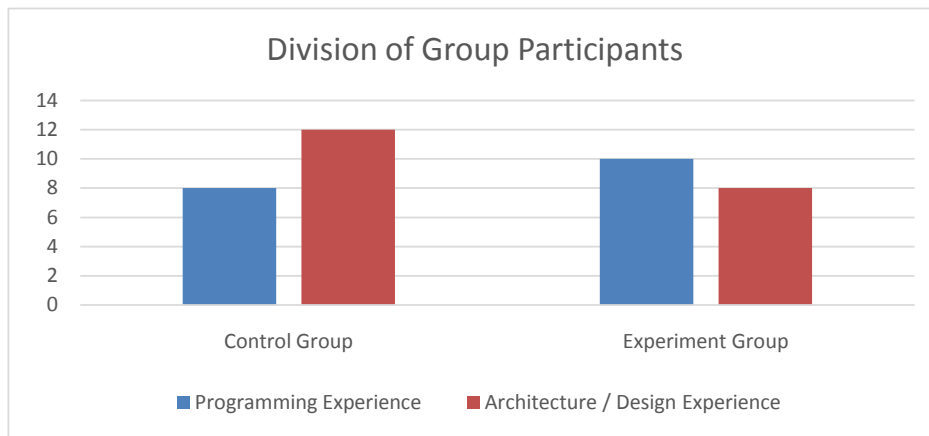


Chart 1. Distribution of participants according to their experience (In months).

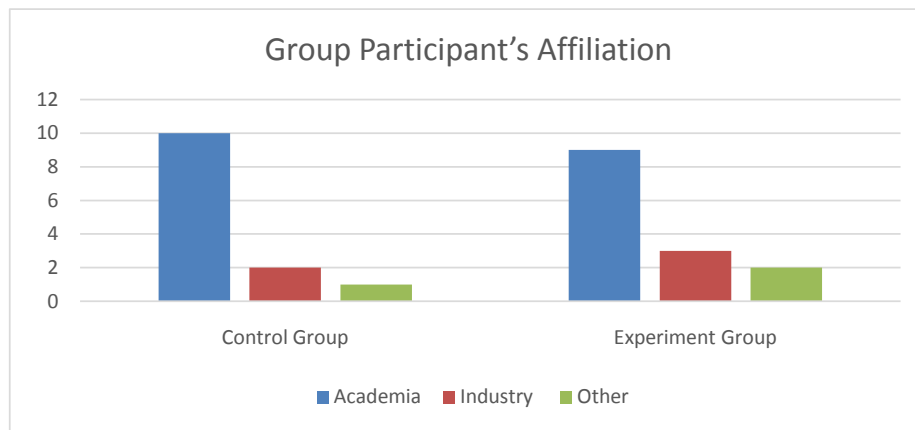


Chart 2. Distribution of participants according to their experience in different affiliation (In months).

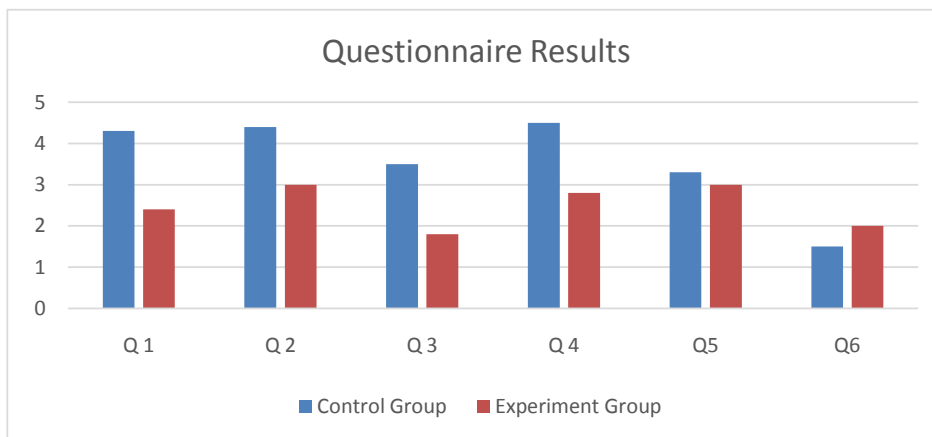


Chart 3. Average questionnaire results comparison by both groups.

to give the feedback about the proposed knowledge management based methodology of requirements gathering. The results show that the experiment group gives better results. Experiment group is go through with all the steps proposed in a Proposed solution section. Later on, when the control group compares their results with experiment group, they also give their positive remarks about that knowledge management based solution.

Table 3. Statistical analysis of time taken in development.

Sr. #	Time Taken in Requirements Engineering		Time Taken in Development	
	Control Group	Experiment Group	Control Group	Experiment Group
1.	40 Hours	50 Hours	60 Hours	20 Hours
	Mean of Control Group	50	Mean of Experiment Group	35
	Median of Control Group	50	Median of Experiment Group	35
	S.D of Control Group	14.14	S.D of Experiment Group	21.21

Table 4. Influence of participants on basis of experience.

Group Affiliation	Arch./Design Experience	Programming Experience	Participants	Participants having both knowledge
Control Group	3	2	4	1
Experiment Group	4	2	4	2

6.3. Limitations of the Study

The study is implemented in an academic environment and has a small industry contribution. However we can further extend our work by implementing the proposed solution to an industry level. As there are small number of participants which is also a limitation of the study. Because if we have large number of participants, we can analyze the results in more better way.

7. Conclusion and Future Work

This paper is about presenting a knowledge based model to support the software requirement engineering phase which will definitely be a supportive material or supportive product for a complete software development methodology. Our proposed model is based on the study of Faxiang Chen, Burseitn F. [11], but more explanatory and detailed form. We present a knowledge model, which will help the software engineers to learn deeply about the software organization as well as the problem domain. Also, in our proposed model, the feedback of customer is also included continuously to get better and efficient results.

To implement proposed model, we use controlled experiment approach by using guidelines given in [32] [33]. We give a small task of simple police station automation system to graduate students of university located in Rawalpindi. Later on, we give a same project assignment to graduate students of another university located in Islamabad, and they were restricted to learn problem domain by using steps of proposed knowledge based model. Later on, we get a feedback from both groups as well as their supervisors about proposed methodology. Similarly, to refine the results more, we use experts review session to get feedback and comments about our proposed methodology. The resultant of our research is a well-defined and well structure problem domain as well as a proper knowledge model which will help the software engineers to learn about organization.

In the future, we can analyze this model by changing different experiment objects which are basically other software systems of real enterprises and more complex. By changing objects, our variables will also change, like in this case we apply it on graduate students, further we can apply it on software engineers having rich experience. Another future work of this study is use of ontologies, experiencing different data sets and including some other emerging things.

Acknowledgements

I am very Thank full to my Professor Dr. Mamoona Humayoun for her support in this study. I also thank to all the participants as well as their professors who take part in this experiment.

References

- [1] Chikh, A. (2011) A Knowledge Management Framework in Software Requirements Engineering Based on SECI Model. *Journal of Software Engineering and Applications*, 4, 718-728. <http://www.SciRP.org/journal/jsea>

- <http://dx.doi.org/10.4236/jsea.2011.412084>
- [2] Flores, F. and Fransico, M.M. (2010) Towards a Systematic Service Oriented Requirement Engineering Process (S-SoRE). In: Varajão, J.E.Q., Cruz-Cunha, M.M., Putnik, G.D. and Trigo, A., Eds., *ENTERprise Information System*, Springer, Berlin, 111-120. http://dx.doi.org/10.1007/978-3-642-16402-6_12
 - [3] Batra, M., Malik, A. and Dave, M. (2013) Formal Methods: Benefits, Challenges and Future Direction. *Journal of Global Research in Computer Science*, **4**.
 - [4] Boehm, B.W. (1984) Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, **1**, 75-88.
 - [5] Andriole, S. and Safeguard Sci. Inc. (1998) The Politics of Requirements Management. *IEEE Software*, **15**, 82-84. <http://dx.doi.org/10.1109/52.730850>
 - [6] Flores, F., Mora, M., Álvarez, F., O'Connor, R. and Macias, J. (2008) Handbook of Research on Modern Systems Analysis and Design Technologies and Applications. In: Global, I.G.I., Ed., *Chap. VI: Requirements Engineering: A Review of Processes and Techniques*, Minnesota State University, Mankato, 96-111. <http://www.igi-global.com/book/handbook-research-modern-systems-analysis/490>
 - [7] Sommerville, I. (2011) *Software Engineering*. Pearson Education, Upper Saddle River.
 - [8] IEEE Std. 830-1998 (1998) IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society Software Engineering Standards Committee, No. SH94654.
 - [9] Davenport, T.H. and Prusak, L. (1998) *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston.
 - [10] Dybå, T. (2011) *Enabling Software Process Improvement: An Investigation on the Importance of Organizational Issues*. PhD Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim.
 - [11] Chen, F.X. and Burseitn, F. (2006) A Dynamic Model of Knowledge Management for Higher Education Development. *The Proceedings of the 7th International Conference on Information Technology Based Higher Education and Training*, Ultimo, 10-13 July 2006, 173-180. <http://dx.doi.org/10.1109/ithet.2006.339762>
 - [12] Devedzic, V. (2011) Knowledge Modeling—State of the Art. *Journal of Integrated Computer-Aided Engineering*, **8**, 1-27.
 - [13] Basili, V.R., Caldiera, G. and Rombach, H.D. (1994) The Experience Factory. In: Marciniak, J.J., Ed., *Encyclopedia of Software Engineering*, John Wiley, Hoboken, 469-476.
 - [14] Olmos, K. and Ordas, J. (2013) A Strategy to Requirements Engineering Based on Knowledge Management. *Requirements Engineering*, **19**, 421-440.
 - [15] Gacitua, R., Ma, L., Nuseibeh, B., Piwek, P., de Roeck, A., Rouncefield, M., Sawyer, P., Willis, A. and Yang, H. (2009) Making Tacit Requirement Explicit. *Proceedings of the 2nd International Workshop on Managing Requirements Knowledge (MARK)*, Atlanta, 1-1 September 2009, 40-44.
 - [16] Rus, I. and Lindvall, M. (2002) Knowledge Management in Software Engineering. *Journal of Software*, **19**, 26-38.
 - [17] Khalid, S., Shehryar, T. and Arshad, S. (2015) The Role of Knowledge Management in Global Software Engineering. *Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management*, Dubai, 3-5 March 2015, 1-5.
 - [18] Ward, J. and Aurum, A. (2004) Knowledge Management in Software Engineering—Describing the Process. *Proceedings of the 2004 Australian Software Engineering Conference*, Washington DC, 13-16 April 2004, 137-146. <http://dx.doi.org/10.1109/aswec.2004.1290466>
 - [19] Kess, P. and Haapasalo, H. (2002) Knowledge Creation through a Project Review Process in Software Production. *International Journal of Production Economics*, **80**, 49-55. [http://dx.doi.org/10.1016/S0925-5273\(02\)00242-6](http://dx.doi.org/10.1016/S0925-5273(02)00242-6)
 - [20] Dingsøy, T., Brede Moe, N. and Nytro, O. (2001) Augmenting Experience Reports with Lightweight Postmortem Reviews. *Proceedings of the 3rd International Conference on Product Focused Software Process Improvement*, Kaiserslautern, 10-13 September 2001, 167-181.
 - [21] Aurum, A., Parkin, P. and Cox, K. (2004) Knowledge Management in Software Engineering Education. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, Joensuu, 30 August-1 September 2004, 370-374. <http://dx.doi.org/10.1109/icalt.2004.1357439>
 - [22] Li, J.J. (2007) Sharing Knowledge and Creating Knowledge in Organizations: The Modeling, Implementation, Discussion and Recommendations of Web-Log Based Knowledge Management. *Proceedings of the 2007 International Conference on Service Systems and Service Management*, Chengdu, 9-11 June 2007, 1-6.
 - [23] Wongthongtham, P. and Chang, E. (2007) Ontology Instantiations for Software Engineering Knowledge Management. *Proceedings of the 2007 IEEE International Symposium on Industrial Electronics*, Vigo, 4-7 June 2007, 1859-1863.

- [24] Georgiades, M.G., Andreou, A.S. and Pattichis, C.S. (2005) A Requirement Engineering Methodology Based on Natural Language Syntax and Semantics. *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, Paris, 29 August-2 September 2005, 473-474.
- [25] Cuena, J. and Molina's, M. (2000) The Role of Knowledge Modeling Techniques in Software Development: A General Approach Based on Knowledge Management Tools. *International Journal of Human-Computer Studies*, **52**, 385-421. <http://www.idealibrary.com>
- [26] Pilat, L. and Kaindl, H. (2011) A Knowledge Management Perspective of Requirement Engineering. *The Proceedings of the 5th International Conference on Research Challenges in Information Science (RCIS)*, Gosier, 19-21 May 2011, 1-12.
- [27] Olmos, K. and Rodas, J. (2014) KMoS-RE: Knowledge Management on a Strategy to Requirements Engineering. *Journal of Requirements Engineering*, **19**, 421-440.
- [28] White, S.M. (2010) Application of Cognitive Theories and Knowledge Management to Requirements Engineering. *Proceedings of the 4th Annual IEEE of Systems Conference*, San Diego, 5-8 April 2010, 137-142.
- [29] Rouse, W.B., Cannon-Bowers, J.A. and Salas, E. (1992) The Role of Mental Models in Team Performance in Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, **22**, 1296-1308.
- [30] Lambe, P. (2007) *Organizing Knowledge and Organizational Effectiveness*. Chandos Publishing Ltd., Oxford.
- [31] Ahsan, M., Hafeez, Y., Anwar, A. and Azeem, M.W. (2014) Knowledge Management Model for Support of Requirement Engineering. *Proceedings of the 2014 International Conference on Emerging Technologies (ICET)*, Islamabad, 8-9 December 2014, 7-12.
- [32] Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K. and Rosenberg, J. (2002) Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions on Software Engineering*, **28**, 721-734. <http://dx.doi.org/10.1109/TSE.2002.1027796>
- [33] Jedlitschka, A., Hamann, D., Gohlert, T. and Schroder, A. (2005) Adapting PROFES for Use in an Agile Process: An Industry Experience Report. *Product Focused Software Process Improvement*, **3547**, 502-516.