# Knowledge Factors and Models of Requirements Change Management Process Based on Causality Analysis

## Yuqing Yan\*

School of Finance, Guangdong University of Foreign Studies, Guangzhou 510006, China

\* Corresponding author. Tel.: +86 13928781716; email: 56238305@qq.com Revised submitted June 19, 2018; accepted July 30, 2018. doi: 10.17706/jsw.13.7.386-394

**Abstract:** Requirements change(RC) is still a key challenge to the success rate of software, a crucial factor of software risk. For decreasing the possibility of RC, improving the quality of software, one outlet is to change management thought of RC, to adopt knowledge management(KM) method. According to the steps of KM, the knowledge sources in RC management process(RCMP) should be first discovered, and the relationship among them should be connected. This paper analyzes the key knowledge existed in RCMP, and the essence of RC, formally defines and classifies inner and external causes of RC which is integrated into an ontology model of change causes, as well as presents an ontology model of problematic requirements . By analyzing RCMP on the basis of change management steps, another ontology model of RCMP is built, and all models are finally integrated into OntoRCMPonCau which is concerned of change causes. An application of OntoRCMPonCau is introduced to collect knowledge in RCMP.

Key words: Requirements change management, knowledge, ontology.

## 1. Introduction

The volatile nature of software requirements is still considered one of the leading factors to information technology project failure[1]. Being able to cater and accommodate the volatility of requirements, the agile method has become an increasingly important development methodology. The Standish Group has discovered that agile projects are far more likely to succeed. But it can't deny that the efficiency of agile method is not unlimited. It is defined by several development principles called the agile manifesto[2], not a well organized methodology. It neglects detailed documents. Some researchers[3] compared the agile and traditional models, found that the agile project's success rate can be improved by using requirement documentation in the RE process such as UML diagram, reverse engineering and peer review. Less documentation means KM could not be well used in its development. But the initiators realized the tacit knowledge that exists in human's mind, behavior and perception is important[4], which can't be acquired directly from human's brain, but can be obtained by communicating with knowledgeable and professional people[5]. For the agile, there is no guidance about how to accumulate and manage implicit knowledge in software development process. Software project and requirements engineering(RE) management belong to KM which is a methodology for achieving breakthrough business performance through the synergy of people, processes, technology, policies, environment; a discipline of enabling individuals, teams and entire organizations to collectively and systematically create, store, share and apply knowledge, to better reach the objectives. Here in this paper, we adopt the method of KM to establish ontological models of RC and RCMP as the way of amassing knowledge in RCMP.

This paper is organized as follows: Section II reviews the related works about using ontology to enhance RE process especially RC management. Section III discusses key tacit knowledge that exists in RCMP by analyzing

literature and interviewing industrial experts. Section IV establishes an ontological presentation method from the viewpoint of software life cycle by defining concepts and relationships in managing RC, an integrated model OntoRCMPonCau is constructed. Section V discusses how to use OntoRCMPonCau to collect information in software life cycle. The last section concludes this research and lays the foundation of work to come.

## 2. Related Works about Using Ontology in RE

It is acknowledged that ontology is a useful tool for expressing abstract concepts and relationship. A main application of knowledge engineering community in requirements engineering has been to ontologically model domain knowledge to improve the acquisition of domain requirements, or to improve the management of requirements, which is the first step of KM: knowledge creation. In acquiring and analyzing requirements, Jin Zhi[6] constructed software requirement ontology meta-models (enterprise ontology and domain ontology) to conduce to domain users to describe their real systems. Sharma etc [7] analyzed the factors related to RE process, presented a hierarchical ontology framework OntoAidedRE for eliciting requirements. Based on i\* modeling method, Deb etc [8] discussed how to integrate distributed ontology from different stakeholders to make requirements more complete. In managing requirements, ontology was used to model and analyze requirements, used for detecting and processing inconsistencies, conflicts and redundancies among requirements [9]-[11]. In Global Software Development environment, there are challenges for knowledge sharing, reuse and communication for scattered stakeholders, there may give birth to serious possibilities of RC, for solving this problem, Khatoon etc [12] presented a solution by building a conceptual model for RC management driven by ontology modeling, but they did not present any details for the process of RC. In [13], the authors provided conceptual models and algorithms to develop ontology for the evolvement management, less attention to RC management. Castañeda etc[14] stated three potential uses of ontologies in organizing the knowledge in RE but not including that of the requirements management. Jayatilleke and Lai[15] presented an abstract model for depicting RC based on GQM and RDF methods, but there was no details and usage provided. Burgstaller and Egyed collected the knowledge of requirement regions to help guide developers in establishing requirements-to-code traces for better managing requirements change[16]. Ernst et al sought solutions for managing requirements evolution, which involved reusing method that is an important view of KM[17].

In the last two decays, more and more ontological methods used in requirements elicitation, to make requirements more complete in order to decrease the possibility of RC. But until now less literature was drawn to ontology modeling in RCMP[18], [19].

#### 3. Knowledge Related to RC Process and Management in RE

RC is inevitable in software development process. Managing RC is a critical activity in software project development, insufficient or none change management may lead to unexpected reworking and cause projects to lag behind schedule, overrun budget or even fail. Apparently, the process of managing changes in software life cycle is knowledge-intensive. It is necessary to apply a holistic, formal and knowledge oriented approach to manage RC for avoiding extra cost and software risk.

#### 3.1. The Key Elements of Knowledge Resources in RCMP

Requirements constantly change. Yet this is not a problem. The real problem is lack of understanding the nature of volatility [13], [15]. In order to understand RC well, we should first understand the management process of change, shown in Fig. 1, the solid line arrows mean execution flows, dotted line arrows mean action objects. Firstly, stakeholders(engineers, users, managers, marketing personnel, etc) raise RC requests. There must be some reasons to cause this action. Secondly, in users' support, development engineers check what the real problems happen. Thirdly, engineers will analyze change impacts to all components that exist in different software development phases. Fourthly, the members of change control team (normally they are the same engineers in the above steps) evaluate and implement the requests. Fifth, if the revisions for the specifications can't pass, then or the revisions continue or new RC requests will be reported.

#### Journal of Software

Figure 1 can show the whole RCMP forms an execution loop, involved with different stakeholders whose knowledge such as education background, experiences, innovation capability, presentation/communication skill, etc are very important and decisive for the effect of requirements elicitation and management[20,21]. It is exactly the same as the conclusion that the Standish Group made after twenty more years' investigation: "People are the primary drivers of project success or failure"[1]. Except the knowledge source that comes from stakeholders, Figure 1 also shows other knowledge such as the change strategies, the methods of examining, analyzing, accessing, implementing RC requests as well as revising and checking specifications, the legislations or laws or rules that cause requirements to change, etc.



Fig. 1. The process of RC management.

## 3.2. A Brief Introduction for RC Causes and their Classification

In 1991, John Brooke has stated[22] that in the information system there were many potential sources of change which could origin from social environment, organization decisions, users (e.g. experience, personal objectives, time and resource, constraints, mood, and other factors). 1993, Harker etc[23] specifically paid attention to the origins of RC and the classification. Since then, a lot of literature has dealt with similar topic about the causes or origins of RC and their classification[24,25].

But, none of the research placed RC factors and their relationships from the point of knowledge organization and management. The achievements from the literature have not yet played their real role in effectively managing RC in practices. And somehow, perhaps it can explain why until now RC is still a crucial challenge to the success of software development projects.

## 4. An Ontology Integration Model for RC Management Process

Software development process is knowledge-intensive, so is RCMP. Many factors are responsible for RC phenomenon. They can be the complexity or the size of the project itself or changes in business environment, technology supplies, politics, user's organizational environment or stakeholders' subjective factors or defects of software development, etc, which form implicit knowledge. And obviously, these factors influence each other through causality or dependency or other relationship which are usually tacit knowledge. Additionally, in the

literature about researching the management of RC or the identification or classification of RC causes, there often exist vague concepts, relationship and terminology, which give rise to difficulties in understanding and practical application. In order to develop a comprehensive mechanism of RC management, to improve RC management in industries, it is necessary to mine, organize and utilize these knowledge, to establish knowledge-based methods in RC process for communication, sharing and reuse.

#### 4.1. Cause Ontology Model

In the past two decades, there have been some problems in recognizing and categorizing the causes or origins of RC by reviewing literature[24], [25]. On one hand, the abstractness of factors which trigger changes is obscure, not distinct, and the real nature of RC is not very well understood. Some factors are indirect influences on requirements or system, can be recognized as external ones[22,24] such as "customers /end users gradually understand the developing system"[4]. There are others called inner factors[24] such as "requirement errors, conflict or inconsistent"[4], which are direct effects on requirements or system. But basically in literature, there are no terminologies or definitions for external and inner factors as well as the differences between them, let alone the different abstract levels among them. On the other hand, less literature was concerned with knowledge-based RCMP, paying less attention to constructing the knowledge in this process, the knowledge of change causes as well as the relationship among these concepts from the point of knowledge engineering and management. We have realized the deficiencies in the research of past literature and already established guidelines for classifying RC causes and a hierarchy model for sorting possible causes in software cycle life[24], [26].

The goal of ontological models used in acquisition of requirements is to acquire excellent requirements, which would be beneficial to decrease the possibilities of software risk and requirements change. Wiegers described the characteristics of excellent requirements as correct, accurate, complete, consistent, necessary, feasible, etc. If requirements are bad, that means they are incorrect, ambiguous, creeping, inconsistent, redundant, infeasible, etc[27], then the RC quests must be produced and submitted, the RC process must enter the procedure of RCMP.

For better understanding the method which is presented here, we will define some concepts as follows.

**Definition 1**. Inner Cause(IC) of RC is a map from a bad characteristic of requirements(BadCharRe) to the expression set of bad requirements(ExpBadRe) which are correspondent to this bad characteristic, it is a two-tuple set: IC = (bad characteristic, expressions of requirements with this bad characteristic), a one-to-many mapping. That means one bad characteristic can have several bad requirements with this characteristic. For example, "incorrect" is a bad characteristic of requirements, the expression set of incorrect requirements can be incorrect grammar or semantic, constrains are not required.

We can easily understand that making IC occur is owing to many external factors such as engineers are not experienced, personal objectives are not clear, schedule or resource are too much constrained, users' mood are unstable, etc. These factors don't directly trigger RC to happen, they produce the potential possibilities of RC, they are blamed for bad requirements. These external influences are abundant and complex. Here, by reviewing literature and interviewing industrial experts, we define External Cause (EC) as a hierarchical structure as follows.

**Definition 2.** EC of RC is a hierarchical model (shown in the left side of Figure 2), the factors on top level are: Stakeholder, Project Feature, Organization, Technique, Process, Environment. Each element of upper levels can be decomposed into sub-factors of lower levels. For example, Stakeholder of top level can be decomposed into Engineer, User, Marketing Seller, etc. And Engineer is composed by Analyst, Programmer, Tester, etc. Project Feature is including Size and Complexity.

It is not difficult to conceive EC structure as a multi-way tree, a leaf node might be a null or an expression of bad requirements. The ontology model of RC causes is composed by two of ontology meta-models IC and EC, shown in Fig. 2, it shows there are interrelationships between EC and IC. That means these two kinds of factors influence each other. For example, bad communication between developers and customers as a EC will cause requirements creeping or incomplete or redundant or feasible or other bad features which all belong to ICs. In

reverse, incomplete or incorrect requirements that belong to ICs will challenge engineers in their abilities, experiences, knowledge, that is said ICs will produce ECs.



Fig. 2. Ontology model of RC Causes.

# 4.2. Problematic Requirement Ontology Model

When any of RC causes occurs, it means bad requirement(s) exist(s), at this point, requirement(s) is(are) problematic. So we can define the ontology model of problematic requirement(PR) as follows.

**Definition 3**. The ontology of PR is a set which is composed of null, one or more bad requirements, shown in Fig. 3.



Fig. 3. PR Ontology model.

# 4.3. The Ontology Model of RCMP

When bad requirements emerge, change requests(CRs) will be submitted by stakeholders, then the RCMP will proceed which is already shown in Figure 1. For constructing an abstract presentation of RCMP, we define an ontology model of RCMP as a seven-array (each element is regarded as an ontology operation meta-model): (SCR, ECR, ACI, ACR, ICR, CRS, CCR). This model tells us there is a lot of knowledge (concepts, actions, rules, methods, etc) which have to be used in RCMP, they need to be continually collected, architected and renewed for further sharing, reuse, for building an automatic RCMP supported by knowledge engineering and management.



Fig. 4. The Ontology model of RCMP.

# 4.4. An Ontology Model of Integrated RCMP Based on Causality

"Developers need to know what changes have been made to an artifact they currently work on." Maciaszek said[28], "When did we inadvertently introduce this bug?", "Finding the cause of the problem gives a hope for unraveling the situation." These thoughts are also all about RC and its management, also mean while in managing RC, the causes of change should be considered. With a holistic view about RCMP, all above models should be considered in a same framework, should be integrated together into a model called OntoRCMPonCau, it is shown in Figure 5. In OntoRCMPonCau, each sub-model has its play. And no matter meta-model or integrated model, if they could play their real functions in managing RC, their attributes or attribute values should be given to each object(concept, operation, relationship, etc) that belongs to the corresponding model. And structured instances based on knowledge acquisition should be designed for all ontology objects to create different repositories according to their roles. It will be a systematical and challenging task. Our research team has finished part of attributes collecting for main objects above, case studies are proceeding in small and medium software enterprises in Guangdong Province of China. The advantage of this idea and the model OntoRCMPonCau has appeared.



Fig. 5. OntoRCMP PonCau model.

# 5. The Method to Use OntoRCM PonCau

A requirements baseline is a set of requirements that has been reviewed and agreed upon and served as the basis for further development[27]. The formal change process will begin from the baseline. Before setting up the baseline, requirements are in the situation of acquirement and analysis, can be changed at any time owing to users or other factors. After the decision makers define a baseline, the business analysts should place the requirements under change control. In change process(CP) from the beginning of requests, information that records CP and connects individual requirements to other system elements should be collected. We have developed a document template which consists of three forms[29], shown in Table 1 to Table 3.

ange Request
ange Request

PN ID RT RC AP D CC CT P M										
	PN	ID	RT	RC	AP	D	CC	CT	Р	Μ

**Notes.** PR: project name, ID: project ID, RT: requirements type, RC: request content, AP: applicant, D: date of request, CC: change causes, CT: change type, P: development phrase, M:memo.

#### Journal of Software

Table 2. Information	of Reviewing Request
----------------------	----------------------

PN	ID	ReC	AC	CCo	CS	CD	CP
RI	Ev	ED	DMR	DR	DM	DD	М

**Notes.** ReC: related components, AC: affected components, CCo: change cost, CS: change schedule, CD: change difficulty, CP: change priority, RI: risk index, Ev: evaluator, ED: evaluation date, DMR: decision-making reasons, DR: decision results, DM: decision maker, DD: decision-making date.

Table 3. Information of Change Request

PN	ID	FR	Im	ST	FD	RR	R	Μ	
----	----	----	----	----	----	----	---	---	--

**Notes.** FR: final requirement, Im: implementor, ST: spending time for change, FD: finishing date, RR: reviewing results, R: reviewer.

The idea of these three tables comes from OntoRCMPonCau, and some fields such as "CC" in Table 1 need to be analyzed and recorded in details according to Figure 2 and Figure 3.

We are doing case studies in Guangzhou Zhongzhi software development company limited, China. We will further report the results.

#### 6. Conclusion and Future Work

The view of KM is put into use in the field of requirements change management. Some knowledge-related concepts and relationship in RCMP are established by ontological modeling. An integrated model OntoRCMPonCau with the concern of causality of RC is presented. A potential application of OntoRCMPonCau is proposed for guiding the collection of information in RCMP.

Next, we will do case studies to collect knowledge in practical RCMP according to Table 1 to 3, and set up interviews in industries about how to create repository for/by OntoRCMPonCau model.

#### Acknowledgment

This work is supported in part by grants from the Municipal Philosophy and Social Science of Thirteenth Five-year Plan Project of Guangzhou (2017GZYB45), the Provincial Philosophy and Social Science of Thirteenth Five-year Plan Project of Guangdong (GD17CGL18).

#### References

- [1] Alan. Z. What we really know about successful projects. Retrieved from: https://www.scrumalliance.org/community/articles/2016/october/what we really know about successful projects#sthash.B1lUy96n.dpuf
- [2] Beck, K., Beedle, M., Bennekum, A V. (2001). The agile manifesto. *Journal of Applied Behavioral Science*, 2001, *14(6)*, 55–58.
- [3] Batool, A., Motla, Y. H., & Hamid, B. *et al.* (2013). Comparative study of traditional requirement engineering and Agile requirement engineering. *Proceedings of the International Conference on Advanced Communication Technology*.
- [4] Duffy, J. (2000). Knowledge management: To be or not to be? *Information Management Journal, 39*, 33-43.
- [5] Lucia, A., & Qusef, A. (2010). Requirements engineering in agile software development. *Journal of Emerging Technologies in Web Intelligence*, *2*(*3*), 5-8.
- [6] Jin, Z. (2000). Ontology-based requirements elicitation. *Chinese Journal of Computers*, 2000, 23(5), 486-492.

- [7] Sharma, S., & Ingle, M. (2011). An ontology aided requirement engineering framework. *International Journal of Advanced Research in Computer Science*, *2*(1), 279-283.
- [8] Deb, N., Chaki, N., & Ghose, A. (2015). Using model towards ontology integration and completeness checking in enterprise systems requirement hierarchy. *Model-Driven Requirements Engineering Workshop*.
- [9] Zelenkov, Y. (2016). Impact of knowledge management and change management on the effectiveness of the firm: Evidence from the russian companies. *Proceedings of the International Knowledge Management in Organizations Conference on the Changing Face of Knowledge Management Impacting Society.*
- [10] Nguyen, T H., Bao, Q. V., & Lumpe, M., *et al.* KBRE: A framework for knowledge-based requirements engineering. *Software Quality Journal, 22(1)*, 87-119.
- [11] Knauss, E., Liebel, & G., Schneider, K., *et al.* (2017). Quality requirements in agile as a knowledge management problem: More than just-in-time. *Proceedings of the International Requirements Engineering Conference Workshops*.
- [12] Khatoon, A., Motla, Y. H., & Azeem, M, *et al.* (2014). Requirement change management for global software development using ontology. IEEE, *Proceedings of the International Conference on Emerging Technologies*.
- [13] He, H., Wang, Z., & Zhang, Y., *et al.* (2012). An ontology-based framework of requirements evolvement management. *Journal of Software*, 2012, *7*(9), 2018-2025.
- [14] Castañeda, V., Ballejos, L., & Ma, L. C., et al. (2010). The use of ontologies in requirements engineering. Global Journal of Researches in Engineering, 2010, 10(6), 2-8.
- [15] Jayatilleke, S., & Lai R. (2013). A method of specifying and classifying requirements change. *Proceedings* of the Australian Software Engineering Conference.
- [16] Burgstaller, B., & Egyed, A. (2010). Understanding where requirements are implemented. *Proceedings* of the IEEE International Conference on Software Maintenance.
- [17] Ernst, N. A., Borgida, A., & Jureta, I. (2011). Finding incremental solutions for evolving requirements. *Proceedings of the International Requirements Engineering Conference*.
- [18] Yan, Y. Q., & Zhang, Z. H. (2016). A method to connect the sources of software risk and requirements change. Proceedings of the 2016 International Conference on Management, Economics and Social Development.
- [19] Yan, Y. Q., Sun, W. J., Zhong, M., Zhang, Z. (2017). A study to the sources and information collection and metric model of software requirements change. *Proceedings of the 2016 International Conference on Modern Management, Education Technology and Social Science*.
- [20] Janes, A., Remencius, T., Sillitti, A., *et al.* (2013). Managing changes in requirements: An empirical investigation. *Journal of Software: Evolution and Process.*
- [21] Ahmad, Z., Hussain, M., & Rehman, A., *et al.* (2015). Impact minimization of requirements change in software project through requirements classification. *Proceedings of the International Conference on Ubiquitous Information Management and Communication*.
- [22] Brooke, J., & Usability, C. (1991). Adaptable systems and community computing. Proceedings of the Fourth International Conference on Human Computer Interaction.
- [23] Harker, S. D. P., & Eason, K. D., & Dobson, J. E. (1993). The change and evolution of requirements as a challenge to the practice of software engineering. *Proceedings of the IEEE International Symposium on Requirements Engineering*.
- [24] Yan, Y. Q., Sun, W. J., & Zhang, Z. H. (2016). A risk-oriented cause identification framework of software requirements changes and case studies. *Software Engineering*.

- [25] Yan, Y. Q., & Zhang, Z. H. (2016). A method to connect the sources of software risk and requirements change. *Proceedings of the 2016 International Conference on Management, Economics and Social Development.*
- [26] Yan, Y. Q., Sun, W. J., Zhong, M., & Zhang, Z. H. (2016). A study to the sources and information collection and metric model of software requirements change. *Proceedings of the 2016 International Conference on Modern Management, Education Technology and Social Science*.
- [27] Karl, W., & Joy, B. (2013). Software Requirements. Microsoft Press, 2013.
- [28] Maciaszek, L. A., & Liong, B. L. (2006). *Practical Software Engineering: A Case Study Approach*. China Machine Press.
- [29] Yan, Y. Q., Sun, W. J., Zhong, M., & Zhang, Z. H. (20160. A study to the sources and information collection and metric model of software requirements change. *Proceedings of the 2016 International Conference on Modern Management, Education Technology and Social Science*.



**Yuqing Yan** was born in Zhanjiang city, Guangdong Province, China in 1963. She received her bachelor degree in mathematics from South China Normal University in 1985, Master degree and PhD in computer software and theory from Sun Yatsen University, Guangzhou, China in 1999 and 2011.

She is an associate professor of Guangdong University of Foreign Studies, Guangzhou, China, has published more than 20 scientific articles in the field of software engineering management, three of them are:(1)A method to connect the sources of software risk and

requirements change[C](ICMESD 2016: 1179-1185); (2)A Risk-Oriented Cause Identification Framework of Software Requirements Changes and Case Studies[J](Software Engineering. 2016, 19(12):5-9); (3)Dependency based Technique for Identifying the Ripple Effect of Requirements Evolution[J](Journal of Software, 2012,7(3):544-550). Her previous research interest was machine learning, artificial intelligence. Current research interest is knowledge management of software requirements engineering.