

Proposed Framework to Manage Software Requirements and Reuse

Abdulahman Alshehri, Mohammed Basherri, and Rizwan Qureshi

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: arrowhead001@hotmail.com, mbasherri@kau.edu.sa, rmuhammd@kau.edu.sa

Received: 17 October 2017; Accepted: 08 November 2017; Published: 08 December 2017

Abstract—Requirement elicitation and analysis form the focal point in the initial stages of the software development process. Unfortunately, in many software development projects, developers and end-users speak different languages. On one hand, end-users prefer to use natural languages while software developers who are technically perceptive, tend to use conceptual models. This difference in technical knowledge creates a communication gap, a potential cause of poor quality software products or project conflicts. The aim of this paper is to investigate the feasibility of a novel technique that seeks to foster effective elicitation of software requirements and support the implementation of structures that match particular requirements. By combining requirement elicitation and re-usable parts, the proposed solution envisages improvements in the overall software design process leading to enhanced requirement specifications. The novel idea is to incorporate an intermediate step for mapping Unified Modeling Language (UML) to Web Ontology Language (OWL) to enable the addition of ontology languages. The proposed model is validated through a survey. The validation results show that the proposed solution allows software developers to elicit software requirements and implement structures that match certain requirements.

Index Terms—Conceptual modeling, ontology languages, process improvement, OWL, requirement elicitation, software reuse.

Software development is a multi-stage process that aims to ensure the development of a good software product. While there are various approaches to software development called Software Development Process Models, all the models tend to emphasize on the requirement gathering and analysis phase as the focal point for project managers and stakeholders [1]. Essentially, the initial phases of software development entail meetings with stakeholders to determine basic software requirements. The output of this phase is a requirement specification document, which serves as a guideline for the next stage in the software development lifecycle, the design phase. Therefore, the early stages of software development require requirement analysis and the preliminary design work [2]. Essentially, user requirements should describe both the functional and non-functional requirements so that system users, who often lack detailed technical knowledge, can understand them.

This research is aimed to investigate the existing literature to find the issues that are associated with requirement elicitation and analysis. The issues are handled by proposing a framework to help the software professionals to reduce the efforts and impediments of the analysis phase.

The paper is organized as follows. Section 2 outlines the related work. Section 3 defines the problem. Section 4 presents the details of the proposed solution. Section 5 validates the proposed solution.

I. INTRODUCTION

Software development is a process that aims to ensure the development of software that meets specific specifications and user needs. While there are multiple approaches to software development, all the models of software development processes emphasize in requirement gathering and analysis phase as the focal point in the initial development activities. Essentially, the initial activities in software development involve meetings with software developers, project managers, end-users, and other project stakeholders to define user needs. The output of such this stage is a requirement specification document that guides the design stage in the software development lifecycle. Therefore, effective communication among project stakeholders is essential in the initial software design and development activities.

II. RELATED WORK

Ibrahim and Ali [3] explore the area of software process improvement (SPI) with a particular focus on the acceptance of a customized SPI model. The customized SPI model is specially designed for small and medium enterprises of Malaysia to address the current problem that is associated with a paucity of specific international models. In order to solve this problem, Ibrahim and Ali [3] describe integrated SPA. This initiative is developed in order to streamline and integrate various software process benchmarks, assessments, and best practices. The study findings indicate that lack of resources is the most critical barrier in the existing initiatives that are taken to SPI. The main limitation of the proposed solution is that it needs to be tested using multiple case studies to generalize the results [3].

Dagnino [4] focuses on the area of SPI readiness using Game Theory Model. The problem addressed in the article is the lack of SPI readiness among organizations. According to the study, the solution is to use the readiness assessment methodology (EASE), to evaluate the readiness of software organizations to complete SPI initiatives. Another solution is to use Game Theory model to enhance interactions among SPI players [4]. Unfortunately, Game Theory suffers various limitations because it is complicated and cannot allow for the analysis of all competitive problems.

Santos et al. [5] examine the area of SPI but focusing on a reference model designed for the Brazilian context. The study aims to address the necessity of MPS reference model to align the tactical goals of the Brazilian government as well as industry needs. While the MPS.BR program provides significant benefits, it also suffers limitations because a large number of companies in Brazil lack the necessary software quality certifications and the majority of MPS and Capability Maturity Model Integrated (CMMI) certifications are still at the initial stages [5].

A study is conducted to examine the topic of SPI for small software enterprises in Indonesia [6]. The study seeks to address the existing problem associated with chaotic software development processes, low-quality products, and virtually unrepeatability successes. It is asserted that the importance of establishing a process model is required to implement SPI. The major implication of the study is that small-scale enterprises need to consider specific organizational characteristics when designing an appropriate SPI concepts and process models. In order to address this problem, this study proposes a framework for understanding the process of software development, particularly for small software organizations in Indonesia. The main limitation of this study is that it uses ethnographic qualitative approach, which means difficulties in maintaining research rigor and quality, as well as the potential subjectivity that could lead to biased interpretations.

Abdullah and Talib [7] focus on the area of applying knowledge management (KM) systems on SPI in the software industry. The major problem that software industry is facing is the informal ways that project managers are applying KM processes on SPI. Therefore, an initial study has been done in order to propose KMS model for the enhancement of knowledge expedition of SPI for SHO environment. The study involved collecting and analyzing the process in Software development companies. The data is collected using interview technique. It is analyzed using qualitative analysis method. The result shows that some issues of administration and agreement should be conceded such as security, availability, performance, scalability, and reliability. Implementation of KMS model is also possible by KMS functionality and its allied infrastructure. The main limitation is the use of individual semi-structured interviews as a data collection instrument. While this instrument has high validity, it is not very reliable because of difficulties in repeating a focused

interview (non-standardization) and small samples size. Moreover, the study only focuses on the KMS access for common devices.

Klendaer et al. [8] examine the area of Automotive SPICE implementation using the IDEAL model as a strategy towards SPI. The problem under investigation is the challenge facing suppliers in the automotive industry regarding implementation of ISO/IEC 15504 a domain-specific model called Automotive SPICE. The study demonstrates the significant contribution of IDEAL model to the success of SPICE implementation and provides a viable solution for medium-sized research and development units [8]. The main limitation of the study is that its scope does not adjust for factors such as communication and participation. Moreover, the study does not control the influence of factors such as organizational culture.

Kalinowski et al. [9] explore the SPI in the context of Brazil based on MPS-SW reference model. The paper addressed an emerging problem concerning the MPS.BR, the Brazilian Software Process Improvement program; that is challenges in complying with international ISO/IEC 12207 and 15504 standards and incompatibility with the CMMI-Dev. The study concludes that MPS.BR has been successful in providing benefits to SPI but it recommends improvement in the MPS model [9]. Therefore, the main limitations are the inconsistency between MPS.BR with international ISO/IEC standards and incompatibility with the CMMI-Dev.

Martins and de Silva [10] examine the domain of meta-modeling in SPI. The motivation for the study is to solve a key problem facing software organization; that is, improving software quality and productivity as strategies for enhanced competitiveness in the industry. The problem is even bigger because the available meta-modeling methods such as the (SMSDM) Standard Meta-model for Software Development, (SPEM) Software Process Engineering Meta-model, and (OPF) Methodologies OPEN Process Framework focus on process description and thus giving different models for similar processes [10]. This makes it difficult to compare improved processes. In order to resolve this problem, the authors compare recognized meta-models and propose a novel SPI using the metal model developed by Project IT-Process Meta-model or PIT-Process. The major limitation of this study is the lack of specificity in approaches such as OPF, SMSD, and SPEM. Furthermore, the proposed solution does not account for capability and maturity levels assessments in software processes.

Guceglioglu and Demirors [11] focus on the area of process quality within the broader topic of process analysis. In particular, the study notes a current problem where process quality is often ignored in process analysis. This occurs despite its importance along with attributes such as cost and time. The study presents a Process Quality Measurement Model (PQMM) for enabling software companies to analyze process quality effectively via enhanced process definitions. The weakness of this work is a generalization of study findings – the works extend the PQMM solution into various domains but

normalizing the measurement results remains an open problem.

Choi et al. [12] address the area of developing useful recommendations to improve software process assessments. The study notes the paucity of detailed approaches for building recommendations for strategic improvement plans in software processes. This problem makes it difficult to develop constructive recommendations and ultimately affects the improvement of the quality process. A solution is proposed to handle this problem using a systematic approach for developing recommendations through capability-based reference model called Remo [12]. The main limitations associated with Remo are the subjectivity in terms of establishing correlations and the difficulties of using the model among novice engineers.

Dagnino and Cordes [13] propose a model called EASE that is designed for measuring organizational readiness to engage in SPI. The area of research is to evaluate SPI readiness of software development companies. The study is conducted to manage risks involved in software development projects by ensuring organizational readiness. The proposed solution is a model that incorporates stakeholder participation in the identification of drivers motivating them and assigning metrics to ensure effective evaluation of the readiness of an organization [13]. Overall, EASE considers four essential elements in SPI implementation: readiness and commitment from the sponsor, readiness, and commitment from the change agent, organizational readiness, and organizational level of SPI expertise. The main limitation associated with the proposed model is its practical feasibility limited.

Mirna et al. [14] investigate the benefits of using a multi-model environment as a reference model to SPI. The investigation focuses on the domain of SPI. The problem being addressed in this paper is the general lack of awareness among organizations on how to improve software processes. Indeed, preliminary research shows that software organization faces numerous challenges in adapting preferred SPI to their organizational contexts. The authors present a multi-model approach as a viable solution and describe the advantages of using such a reference environment in SPI. The main weakness in this area is the limited success of process improvement initiatives based on this model.

Yonghui [15] describe a framework for SPI that utilizes the Capability Maturity Model Integrated (CMMI) Continuous Model based on Quality Functional Deployment (QFD). Therefore, the research addresses issues in SPI. The problem, motivates this study, is the lack of a universally accepted framework to analyze SPI. It would be unfair to compare international standards or models for SPI because they emphasize on different criteria for quality systems. CMMI is an effective SPI model that can be combined with QFD to provide the effective framework for SPI [17]. The goals of such a framework are threefold: mapping process demands, developing a method and prioritizing SPA based on the process demands. The only limitation of the proposed SPI

Table 1. Summary of the Existing Literature

Title	Limitation
Study on acceptance of customized software process improvement (SPI) model for Malaysia's SME [3].	Proposed solution of limited value because many companies are yet to implement the standard process for software process improvement
A game theory model used to increase the readiness for software process improvement [4].	Complexity of Game Theory and inability to analyze all competitive problems in SPI
MPS.BR Program and MPS Model: main results, benefits, and beneficiaries of software process improvement in Brazil [5].	Lack of software quality certifications in Brazil and most of MPS and CMMI certifications are still at the initial levels.
Process model and software process improvement for small software organization: an ethnographic study in Indonesia [6].	Challenges with ethnographic qualitative research approach; researcher bias and difficulties in maintaining research quality
Knowledge management system model in enhancing knowledge facilitation of software process improvement for software house organization [7].	Low reliability because of non-standardization of interview process and small sample size
Using the IDEAL software process improvement model for the implementation of automotive SPICE [10].	The study does not adjust for pertinent factors communication and participation. The study does not control for the effects of organizational culture.
Results of 10 years of software process improvement in Brazil based on the MPS-SW Model [9].	Inconsistency between MPS.BR and pertinent international ISO/IEC standards and incompatibility with the CMMI-Dev reference model
PIT-ProcessM: a software process improvement meta-model [10].	Lack of specific approaches in OPF, SMSD, and SPEM. Proposed solution does not sustainance the valuation of maturity levels and capability in software processes
The application of a new process quality measurement model for software process improvement initiatives [11].	Generalizability problem of study findings and normalization of the measurement results
ReMo: a recommendation model for software process improvement [12].	The proposed solution (ReMo) is difficult to use by novice engineers. It has suffered from high level of subjectivity in establishing correlations
A model to measure organizational readiness for software process improvement [13].	Practical feasibility limited due to limited success of process improvement initiatives
Advantages of using a multi-model environment in software process improvement [14].	Constrained success of process improvement activities in view of the multi-model approach
Software process improvement framework based on CMMI Continuous Model using QFD [15].	Integrating QFD and CMMI is a challenge because they address different areas.
Towards an improvement of software development process based on software architecture, model-driven architecture, and ontologies [16].	Challenges in mapping elements in MDA, ontologies, and UML to create a combined methodology
Assessment methodology for software process improvement in small organizations [17].	Proposed methodology not yet validated

framework is how to integrate QFD and CMMI because both focus on different areas.

Espiritu et al. [16] describe a methodology for developing software using software architectures. The area of study is SPI. The study is based on the observation that software development using software architecture methodology has declined due to declining popularity of this approach. Consequently, the study proposes a solution based on the integration of ontologies, Model-Driven Architecture (MDA), and Unified Modeling Language (UML) components to create robust software architecture [16]. It is expected integrated approach to modernize and improve current SPI practices. The challenge is mapping individual elements in ontologies, MDA and UML together to create a unified methodology.

Pino et al. [17] examine the assessment of SPI, especially in small enterprises. The study focuses on the area of SPI assessment. It addresses existing problems in process assessments. It is observed that diagnosing processes in small-scale enterprises are challenging because existing methods do not offer the overall view of process capabilities. In order to solve this problem, a novel methodology METvalCOMPETISOFT is proposed for software process assessment. The proposed methodology integrates strategies used in internal or rapid assessment methods and enhanced customization to the unique organizational features in small enterprises. The major drawback in the proposed methodology is that it has not been validated. The proposed solution needs further refinement and evaluation using case studies [17].

The study is conducted to show the importance of reusability using multilayer perception (MLP) [18]. Reusability Assessment Automation Model tool is introduced to facilitate the assessment of reusability using functions instead of structures. The current software systems are developed using reusable components. The proposed tool is applied to a component that how much it is suitable to reuse by checking its attributes such as maintainability. MPL network will define three layers of reusability i.e., high, medium and low. Java programming language is used to develop the proposed tool. Correctness, categorization of bugs, precision and recall are the main attributes reported to evaluate the effectiveness of the proposed tool. The results are found reliable and efficient during the testing of the proposed tool when they are compared with the existing techniques. The results show that the proposed tool improves the process of selecting a reusable component from the existing components while developing a new project to do it automatically. The proposed automated assessment of reusable components will provide significant benefits to software development companies such as time-saving, economical, effortless and cut down resources. The proposed tool is only limited to Java-based systems to identify the reusable components. There is a need to develop an automatic assessment tool to check the properties of reusability in a component supporting all object-oriented programming languages.

Qureshi and Al-Geshari [19] propose a theoretical framework to identify a suitable design pattern to reuse according to the requirements of a customer. There are several design patterns available and it is very complicated for the junior programmers to select a suitable pattern as per the requirements of an application. The proposed framework will help the junior programmers to generate problem-related questions while checking the repository of components. The answers to relevant questions will facilitate to the junior programmers to identify the appropriate patterns of their choices. A survey is conducted to evaluate the proposed framework. The proposed framework needs to be implemented using a tool to check its authenticity.

Component-Based Development (CBD) facilitates to reuse a component [20]. There are well-known benefits of CBD such as high reliability, economical, time-saving and increase productivity. There are few limitations of CBD as well such difficulty in integration, the efficiency of application and compatibility issues. Integration of reusable components is always challenging and risk probe to fail the software projects. The research is conducted to select suitable activities to reduce risks while integrating a component using CBD. The known risks are identified those are associated with reusable components and the mitigation activities are proposed to cater the known risks. A categorization of known risks is provided based on the existing literature. A survey is conducted to validate the known risks and their mitigation strategies. The result of the survey shows that 80% of the respondents support the proposed mitigation strategies to manage the known risks. The proposed risk mitigation strategies are theoretical and it is required to test them in an industrial context to check their effectiveness.

Table 1 summarizes the limitations of related work.

III. PROBLEM STATEMENT

Software development is a process that aims to ensure the development of software that meets specific specifications and user needs. While there are multiple approaches to software development, all the models of software development processes emphasize on requirement gathering and analysis phase as the focal point in the initial development activities. Essentially, the initial activities in software development involve meetings with software developers, project managers, end-users, and other project stakeholders to define user needs. The output of such this stage is a requirement specification document that guides the design stage in the software development lifecycle. Therefore, effective communication among project stakeholders is essential in the initial software design and development activities.

A major problem associated with the early activities in software development projects is the fact that in most cases, users and software developers speak different languages [1,17]. Given that the requirements process is mostly a human endeavor, an effective requirements elicitation method must support the need for stakeholders

to communicate effectively. On one hand, users prefer natural language because it is easy to understand given that users tend to be non-technical people. However, natural language presents a number of challenges when used to write requirements. Some of the challenges include lack of clarity, confusion between system goals, complex design information, and misunderstanding of user requirements. On the other hand, developers obtain their domain knowledge from requirements or from conceptual models, especially frameworks developed in previous works. The major challenge is that users may likely fail to understand such models because they generate highly technical requirements. This paper seeks to propose a solution to this problem focusing on a combination of requirement elicitation or requirements collections from stakeholders, and reusable parts or the use of existing software components, designs, and or code.

IV. THE PROPOSED SOLUTION

This paper presents a novel framework for the semantic processing of software requirements based on the concept of ontologies as shown in fig. 1. The proposed solution aims to enable software developers to elicit software requirements and implement structures that suit particular requirements. The proposed ontology has two components: requirement elicitation and reusable parts.

There are various ways of eliciting requirements including using interviews, brainstorming, questionnaires, and conceptual prototyping. The result of the requirements elicitation is a list of needs or requests described graphically or textually. It is envisaged that using an appropriate tool during the elicitation proves in natural language can allow system developers to check and test the requirement specification. This provides a possible feedback. Secondly, a major reason for developing ontology is to enable separation of domain knowledge from the operational knowledge and reuse of domain knowledge. Therefore, the idea is to establish a relationship between requirements elicitation and reusable parts as a strategy for streamlining the requirement specification and overall software design process.

The solution aims to incorporate an intermediate step involving conceptual design work. The additional step is actually a new method for mapping UML to OWL. This approach provides an opportunity for formal representation of domain knowledge. The challenge is that conventional engineering approaches do not rely on ontology representations using Semantic Web Languages. Therefore, the proposed solution will seek to adapt Protégé to support the addition of ontology languages. The solution involves two major steps. The first step is analyzing XSLT processors, that is, the .NET Framework Class Library and XML Spy to generate OWL files by using XSLT file as the input to enhance the mapping rules generated with Poseidon using OUP.

The second step entails understanding existing OUP examples in order to explore its actual potential. The process of developing an .own file entails developing a model using OUP in Poseidon and exporting it to XMI, then running the application and importing the generated .owl file using Protégé

The proposed solution aims to achieve three goals.

A. Goal 1–Process Natural Language Texts and Mine Them as Domain-Specific Concepts

The first goal is to process natural language texts and mine the texts as domain-specific concepts. This is possible using natural language processing (NLP) engine and application-specific knowledge. NLP and text mining focus on exploiting resources characterized as rich knowledge in order to facilitate the understanding and extraction of unstructured text. The NLP methods and techniques can help in using entities and relationships that serve as indicators mined from texts. Similarly, text-mining techniques can help to find dependencies between the entities. The idea is to use the utility of domain ontologies, which also alludes to a number of challenges including the problem of identifying and defining the concept definitions. The NLP methodologies would provide the means for automating the improvement of ontology from free text documents.

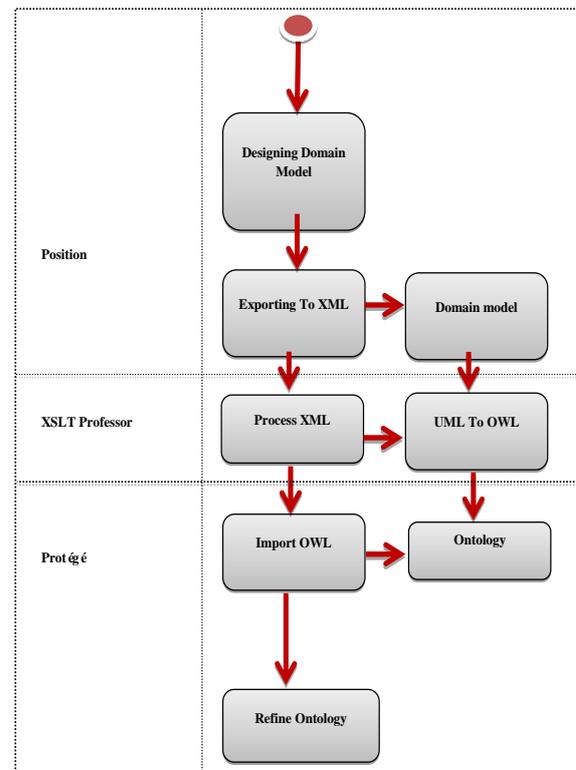


Fig.1. The Proposed Approach.

B. Goal 2–Enable the Ontology System to Guide the Requirements Elicitation

System developers need to keep their requirements well defined and updated throughout the system development lifecycle. Semantic Web technologies have

important applications in requirements engineering because they focus on requirements analysis. The idea is to use semantic guidance system based on concepts and relations of domain ontology to facilitate the definition of (well-structured) requirements. The guidance process relies on concepts, axioms, and relations of domain knowledge to recommend appropriate requirements. The objective is to explore the feasibility of semantic guidance system in supporting requirement elicitation.

C. Goal 3–Develop Techniques for Selecting and Adapting Appropriate Re-Usable Parts

Reusability in software engineering refers to the practice of existing software components, designs, test suites, or documentation to satisfy specific system requirements. The idea of developing a technique for selecting appropriate reusable parts aims to reduce the software development time. Therefore, the focus of the study is to explore possible re-use techniques and approaches in software engineering for reuse in future projects. The goal envisages that reuse will rely on the ability to develop large software components from smaller units, and thus the need to identify commonalities among various components.

Overall, the proposed solution envisages addressing the current problem in requirement engineering by mapping UML to OWL. This will allow system developers and users to communicate effectively from the early phases of software development process. Through the identified goals, it will be possible to investigate the problem further and contribute towards finding a solution. Therefore, further research should address the feasibility of using natural language as a domain-specific concept, using an ontology to direct requirement elicitation, and determine the techniques for effective selection and adaptation of reusable parts. These efforts will contribute to improving software improvement.

V. VALIDATION

The validation of the proposed solution relies on a public survey that is targeting experts in software engineering and professionals in system design and development. The selection of participants has followed a purposive sampling approach in which the researcher relies on personal acquaintances and professional contacts. Overall, 31 professionals participated in the survey. Each of the three goals is evaluated using five questions.

The evaluation of the survey questionnaire responses relies on a Likert scale that is ranging from 1 to 5 i.e., (1) Very High, (2) High, (3) Nominal, (4) Low, and (5) Very Low.

A. Cumulative Analysis of Goal 1.

Table 2 shows the cumulative statistical analysis of the survey responses.

Table 2. Cumulative frequency analysis of goal 1

Q. No.	Very Low	Low	Nominal	High	Very High
Q1	0	0	19.3	29.0	51.1
Q2	0	0	19.3	67.7	12.9
Q3	0	0	25.8	58.0	16.1
Q4	0	12.9	29.0	45.1	12.9
Q5	0	9.6	37.7	32.2	19.3
Total	0	22.5	131.1	232	112.3
Avg.	0%	4.5%	26.4%	46.4%	22.5%

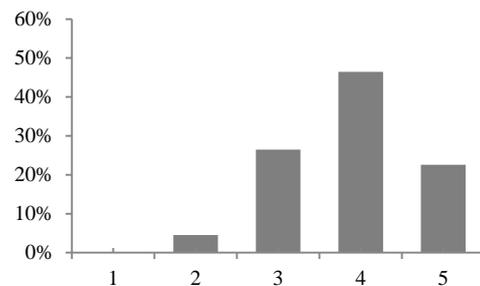


Fig.2. The Cumulative Analysis of Goal 1.

According to Table 2, 68.9% of the responses were in favor of Goal 1, of which 46.4% agreed while 22.5% strongly agreed to the effects of the proposed framework on processing natural language texts and mining them as domain-specific concepts. 26.4% of the cumulative responses were neutral while 4.5% of the participants were not in favor of it as shown in fig. 2.

B. Cumulative Analysis of Goal 2.

Table 3 illustrated the findings of Goal 2. Goal 2 was about enabling the ontology system to guide requirements elicitation. According to Table 3, 52% of the responses were in favor of Goal 2 of which 5% strongly agreed and 47% agreed to it. 15% of the respondents were not in favor of Goal 2. 32% of the respondents remained neutral as shown in fig. 3.

Table 3. Cumulative analysis of goal 2

Q. No.	Very Low	Low	Nominal	High	Very High
Q6	0	15.7	42.1	42.1	0
Q7	0	14.8	25.9	59.2	6.5
Q8	0	25.9	22.2	14.8	6.2
Q9	0	7.4	29.6	62.9	6.1
Q10	0	8.6	39.1	52.1	5.9
Total	0	72.4	158.9	231.1	24.7
Avg.	0%	15.0%	32.0%	47.0%	5.0%

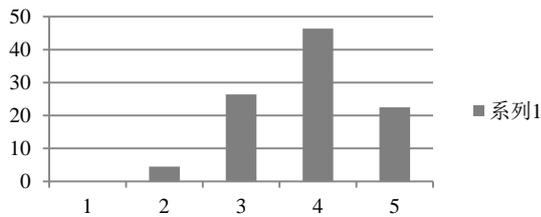


Fig.3. The Cumulative Analysis of Goal 2.

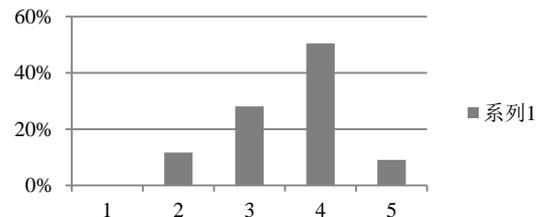


Fig.5. The Cumulative Analysis of 3 Goals.

C. Cumulative Analysis of Goal 3.

Table 4. Cumulative frequency analysis of goal 3

Q. No.	Very Low	Low	Nominal	High	Very High
Q11	0	25.93	14.81	59.26	0
Q12	0	36.00	16.00	48.00	0
Q13	0	6.90	27.59	65.52	0
Q14	0	9.52	19.03	71.43	0
Q15	0	0	52.00	48.00	0
Total	0	52.42	129.43	292.21	0
Avg.	0.00%	15.75%	25.98%	58.27%	0%

Cumulative responses of the Goal 3 were shown in Table 4. Table 4 showed that 58.27% of the responses agreed with goal 3. 25.98% of the responses remained neutral for Goal 3. 15.75% of the responses were not in favor of the Goal 3 as shown in fig. 4.

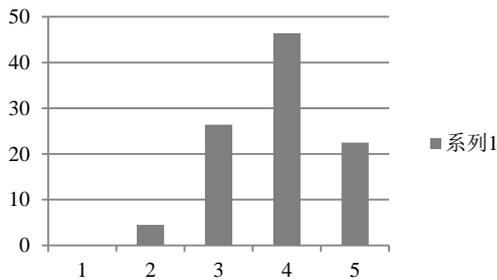


Fig.4. The Cumulative Analysis of Goal 3.

D. Final Cumulative Analysis of Three Goals.

Cumulative responses to the three goals were shown in Table 5. Table 5 showed that 59.6% of the responses agreed with 3 goals in which 9.1% of the software engineers strongly agreed and 50.5% of the professionals agreed to it. 28.1% of the responses remained neutral for 3 goals. A total of 11.7% responses disagreed with the three Goals as shown in fig. 5.

Table 5. Cumulative frequency analysis of 3 Goals

Goal No.	Very Low	Low	Nominal	High	Very High
Goal 1	0	4.5	26.4	46.4	22.5
Goal 2	0	15.0	32.0	47.0	5.0
Goal 3	0	15.7	25.9	58.2	0.0
Total	0	35.2	84.3	151.6	27.5
Avg.	0%	11.7%	28.1%	50.5%	9.1%

VI. CONCLUSION

A major challenge in software development projects is establishing effective communication between users who typically prefer to use natural language and software developers who often use conceptual models. Essentially, this means that users and system developers use different language that could lead to misunderstandings of technical requirements. This research proposes a technique for semantic processing of software requirements using ontologies. In particular, the proposed framework comprises of two components i.e., requirement elicitation and re-usable parts. This study results show that the proposed framework will enable software developers to elicit of software requirements and foster the implementation of structures that match specific requirements. Indeed, the findings show that using ontologies facilitates formal representation of domain knowledge. Future work is to validate the proposed solution using multiple case studies to generalize the result.

REFERENCES

- [1] Z. Subhan, and A. T. Bhatti, "Requirements analysis and design in the context of various software development approaches," *International Journal for Research in Applied Science and Engineering and Technology*, vol. 3, pp. 812-820, 2015.
- [2] A. Souag, C. Salinesi, R. Mazo and I. Comyn-Wattiau, "A security ontology for security requirements elicitation," *Proc. the International Symposium on Engineering Secure Software and Systems*, Milan, Italy, pp. 158-176, 2015.
- [3] S. Ibrahim, and R. Z. R. M. Ali, "Study on acceptance of customized software process improvement (SPI) model for Malaysia's SME," *Proc. 5th Malaysian Conference in Software Engineering (MySEC)*, Johor Bahru, Malaysia, pp. 25-30, 2011.
- [4] A. Dagnino, "A game theory model used to increase the readiness for software process improvement," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, USA, pp. 2928-2933, 2014.
- [5] G. Santos, M. Kalinowski, A. R. Rocha, H. T. Guilherme, K. C. Weber, and A. A. Jose, "MPS.BR Program and MPS Model: main results, benefits, and beneficiaries of software process improvement in Brazil," *Proc. 8th International Conference on the Quality of Information and Communications Technology*, Lisbon, Portugal, pp. 137-142, 2012.
- [6] I. Hidayah, Wahyuni, and Nugroho L. E., "Process model and software process improvement for small software organization: an ethnographic study in Indonesia," *Proc.*

- International Conference on Computer & Information Science*, Kuala Lumpur, Malaysia, pp. 852-856, 2012.
- [7] R. Abdullah, and A. M. Talib, "Knowledge management system model in enhancing knowledge facilitation of software process improvement for software house organization," *Proc. International Conference on Information Retrieval & Knowledge Management (CAMP)*, Kuala Lumpur, Malaysia, pp. 60-63, 2012.
- [8] R. Klendauer, A. Hoffmann, J. M. Leimeister, M. Berkovich, and H. Krcmar, "Using the IDEAL software process improvement model for the implementation of automotive SPICE," *Proc. 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, Zurich, Switzerland, pp. 66-72, 2012.
- [9] M. Kalinowski, K. Weber, N. Franco, E. Barroso, V. Duarte, D. Zanetti and G. Santos, "Results of 10 years of software process improvement in Brazil based on the MPS-SW Model," *Proc. 9th International Conference on the Quality of Information and Communications Technology*, Braga, Portugal, pp. 28-37, 2014.
- [10] P. V. Martins, and A. R. da Silva, "PIT-ProcessM: a software process improvement meta-model," *Proc. 7th International Conference on the Quality of Information and Communications Technology*, Oporto, Portugal, pp. 453-458, 2010.
- [11] A. S. Guceglioglu, and O. Demirors, "The application of a new process quality measurement model for software process improvement initiatives," *Proc. 11th International Conference on Quality Software*, Madrid, Spain, pp. 112-120, 2011.
- [12] S. Choi, Dae-Kyoo Kim, and S. Park, "ReMo: a recommendation model for software process improvement," *Proc. International Conference on Software and System Process (ICSSP)*, Beijing, China, pp. 135-139, 2012.
- [13] A. Dagnino, and A. Cordes, "A model to measure organizational readiness for software process improvement," *Proc. 5th IEEE International Conference on Software Engineering and Applications*, Beijing, China, pp. 1-8, 2014.
- [14] M. Mirna, M. Jezreel, A. Giner, A. Calvo-Manzano Jose, and S. Tomas, "Advantages of using a multi-model environment in software process improvement," *Proc. IEEE Conference on Electronics, Robotics, and Automotive Mechanism Conference*, Morelos, Mexico, pp. 397-402, 2011.
- [15] C. A. O. Yonghui, "Software process improvement framework based on CMMI Continuous Model using QFD," *International Journal of Computer Science Issues*, vol. 10, no. 3, pp. 281-287, 2013.
- [16] F. B. Espiritu, A. S. Lopez, and L. J. C. Rosales, "Towards an improvement of software development process based on software architecture, model driven architecture and ontologies," *Proc. 24th International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Cholula Puebla, México, pp. 118-126, 2014.
- [17] F. J. Pino, P. Cesar, F. Gracia, and M. Piattini, "Assessment methodology for software process improvement in small organizations," *Information and Software Technology*, vol. 52, no. 10, pp 1044-1061, 2010.
- [18] Surbhi Maggo, Chetna Gupta, "MLP based Reusability Assessment Automation Model for Java based Software Systems," *International Journal of Modern Education and Computer Science (IJMECS)*, vol.6, no.8, pp.45-58, 2014.
- [19] M. Rizwan Jameel Qureshi, Waleed Al-Geshari, "Proposed Automated Framework to Select Suitable Design Pattern," *International Journal of Modern Education and Computer Science (IJMECS)*, Vol.9, No.5, pp. 43-49, 2017.
- [20] Fariha Fatima, Saqib Ali, Muhammad Usman Ashraf, "Risk Reduction Activities Identification in Software Component Integration for Component Based Software Development (CBSD)," *International Journal of Modern Education and Computer Science (IJMECS)*, Vol.9, No.4, pp.19-31, 2017.

Authors' Profiles



Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honored as the best researcher in Computer Science discipline from seven campuses of COMSATS Institute of Information Technology, Pakistan in 2008.



Dr. Mohammed Basher is an Assistant Professor and the Chairman of Information Technology Department at the Faculty of Computing and Information Technology in King Abdulaziz University, Saudi Arabia. Dr. Basher received PhD in Computer Science from the School of Engineering and Computer Science at Durham University.



Abdulrahman M. Alshehri is a Master degree student in the IT Department at King Abdulaziz University, Saudi Arabia.

How to cite this paper: Abdulrahman Alshehri, Mohammed Basher, Rizwan Qureshi, "Proposed Framework to Manage Software Requirements and Reuse", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.9, No.12, pp. 49-56, 2017.DOI: 10.5815/ijmeecs.2017.12.06