A Process Improvement in Requirement Verification and Validation using Ontology

Sana Nazir, Yasir Hafeez Motla, Tahir Abbas, Asma Khatoon, Javaria Jabeen, Mehwish Iqra & Khush Bakhat
UIIT - PMAS Arid Agriculture University Rawalpindi
Rawalpindi, Pakistan
sana_taurus5@yahoo.com, yasir@uaar.edu.pk, tahirabbasmalik@yahoo.com, asma.khatoon18@gmaill.com & khushbakhat90@gmail.com

Abstract—**The success of software projects depends on fully conveyed requirements. Contribution of stakeholders contributing in software requirements have a place with distinctive domains, because of which ambiguities, inconsistencies might arise moreover giving rise to incorrectness. Many tools are available for software requirements verification and validation but the issue of stakeholders belonging to different domains is still under consideration. As these requirements are not understood properly, it causes schedule delay and greater cost in the last stages of software. Requirements verification and validation are the most vital activities in requirement engineering that guarantees software quality and has an extraordinary impact on software cost and its success. The requirements verification and validation are an affirmation that the product requirements record is free of unwanted requirements and totally reliable. Keeping in mind the end goal of removing conflict, recognizing failures and making the product requirements document completely useful is the key component. So these requirements should be appropriately confirmed and accepted by the clients before conveying to the development team. Thus, ontology might be picked as an answer for checking and approving the requirements among stakeholders of distinctive domains. In this study improving requirement verification and validation process through ontology. We have used expert reviews and case study to evaluate the developed ontology that indicates different concepts, properties and constraints in the process of requirement verification and validation. With the help of the developed ontology, it is shown that its use for requirement verification and validation shows improvement in consistency, applicability and other factors.**

Keywords: Verification and Validation (V&V), Ontology, Requirements and Process Improvement.

I. INTRODUCTION

Software projects are getting more incomparable and constructive with the passage of time [1]. The achievement of software projects depends on gathering the clients' requirements. Fulfilled clients' requirements from diverse domains is the base of software advancement. To observe whether the software is as indicated by clients' necessities or not, the requirements should be appropriately confirmed and approved [19]. As these stakeholders are from diverse domains therefore issues emerge when the engineers can't understand and clients can't pass on their requirements appropriately [10]. Therefore requirements representation is an issue in software requirements [9].

An attempt needs to be made to satisfy the clients/ end users' needs by giving legitimate nature of requirements in defined time. To affirm that the client is getting legitimate software item as per his need, the specifications of a framework ought to be satisfied [14]. In short, the software should be the reflection of its specification record. The fundamental reason behind it is to enhance fulfillment level of client alongside the nature of the software [9]. In this way, verification and validation of requirements are very crucial for the strongly fulfilled end client [13, 23]. It has been observed that if a client discovers an issue, it will surely enhance the expense in incredible ways, whereas, through discovering an issue at the verification stage causes less cost [2, 24].

Nowadays numerous techniques are accessible for requirements verification and validation [16]. A key purpose behind why this is still so difficult to meet client requirements is that the larger part of requirements report is casual, composed in common dialect, though the last objective (code) is formal [10]. Because of this reason all clients should have the same domain learning to handle the issues raised by having a diverse understanding of the same thing [7].

Section II of this paper contains the problem statement, section III states the related work, section IV consists of method used for research, and section V supports the proposed methodology. Lastly in section VI results and discussions are defined.

II. PROBLEM STATEMENT

Software requirements should be complete and consistent. Requirements verification and validation specifically impacts the nature of the software. Software life cycle can't be viewed as complete without it [16]. In the absence of requirement verification and validation, the project can cause delay in schedule and over cost from the estimated budget, due to finding bugs in later stages. Then again, the principle concern is that the stakeholders have a place from distinctive domains; they may participate in requirements phase usually, and then the ambiguity that arises due to the non-formal or semi-formal language can lead engineers and domain clients to have diverse suggestions about domain information and requirements, and thus have a negative effect on requirements quality. This can be solved by providing shared common understanding. For this purpose ontology can be used to provide shared knowledge.

III. RELATED WORK

In this paper, authors survey current requirements engineering techniques and distinguish future investigation recommended by rising software needs. Firstly, they reviewed the state of the improvement in RE research. The exploration regarding innovations created to address particular requirements tasks is taken into consideration, for example, elicitation, demonstrating, and segmentation. Next they assessed a few methodologies for performing and broadening RE exploration results, to help outline the extent of future examination headings. At last, they highlighted the vital flow and future exploration themes, which expect to address RE requirements for developing frameworks without boundaries [1].

Table 1 Approach for Requirement Verification and Validation

ACTIVITY	TECHNIQUES, ANALYSIS, TOOLS	NOTATIONS
REQUIREMENT VERIFICATION AND VALIDATION		Model Formalism [1]
	Simulation [1]	
	Animation [1,4]	
	Invariant generation [1]	
	Model checking [1]	
	Model Satisfiability [1]	

Different approaches are used for Requirements Verification and Validation which have been shown in different literature. Table 1 shows these techniques [1].

Requirements verification is exceptionally critical for the complex frameworks to avoid expense impact. Authors investigated another strategy for software requirements verification with structural and formal semantics focused around domain ontology [11]. They additionally dealt with requirement elicitation and analysis; encouraged by semantic innovations and domain ontology. As the ontology has legitimate semantics and grants derivation to its semantic components, "intrinsic experience" and "tacit encounter" that will check breadth, constancy and precision of the requirements [26].

The author states numerous roles of ontologies in Requirements Engineering (RE). They clarified this utilization in refined way. The paramount part of this paper is the arrangement of methodologies that incorporate ontologies inside RE. The essential point of it was to explain the path in which the conventional strategies might be united with them [3].

Requirements engineering is the establishment of software engineering, and nature of requirements engineering decides the nature of projects. It is hard to depict standardized structure, however ontology is a formal portrayal of the idea of informing, focusing on the connection between elements. The ontology connected to requirements engineering makes understanding on the semantic level of data. This paper examines the ontology related ideas and assumptions, advances the general schema of utilizing an ontology needs analysis, and then using illustration to represent how to utilize ontology needs analysis, validation and change [21].

Many exploratory papers demonstrate that, although automated and manual instruments are widely utilized, still some software and software projects giveaway with vital failures that may prompt project dissatisfaction. The greater part of software models is independent of software development process and focusing on examination early shortages. This paper offered model is focused around software creating stages adjusted predefined situations. This assessment methodology evacuates some conceivable failure in each of the software improvement stage and abstains from enhancing these imperfections in each of the following stages. Learning is an essential element for a model to be insightful. In this model a learning arrangement is made and executed through which, as per established results, the assessment methodology is altered and redesigned [5].

IV. PROPOSED METHODOOGY

The flow of method used for our research is shown in Figure 1. We started from literature survey, after identifying a problem we did literature review by narrowing down from ontology perspective. After that we proposed a framework for requirement verification and validation through ontology, then moved to specification step and made a conceptual model with the help of this model. From this we moved to Formalization stage; and applied some axioms and rules. At the end ontology is developed and then we present our results after evaluating it from experts of software domain. With the assistance of writing we can infer that ontology might be utilized as an answer for minimizing ambiguities between diverse domains of individuals throughout requirements verification validation stage [17, 23].

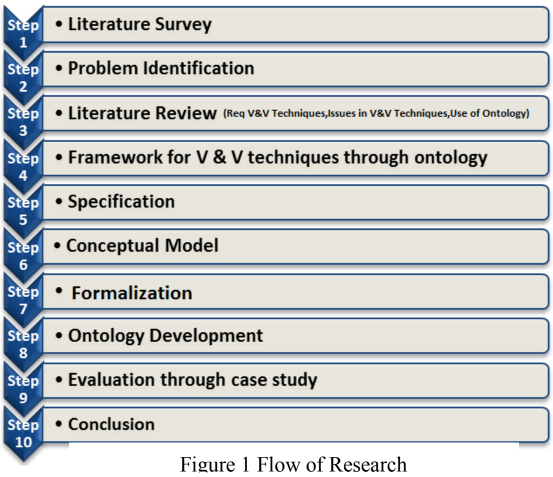


Figure 1 Flow of Research

A. Designing a Framework

To solve the problem of shared domain we have presented a methodology, the framework of which is shown in Figure 2.

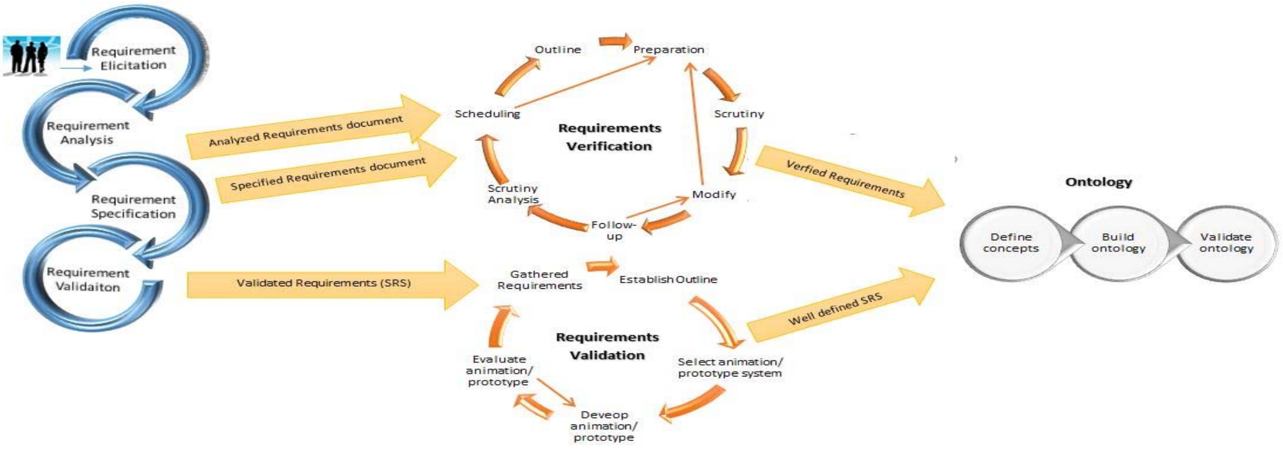


Figure 2 Model for Requirements Verification and Validation through Ontology

The basic process of verification and validation cycles is described as follows. After taking the elicited requirements from the user, they are properly analyzed. In order to make them more complete they are passed through verification cycle. Verification cycle starts from the scheduling step in which arbitrator guarantees that the item is primed for examination, chooses the scrutiny group and appoints roles, plans the meeting place and time, and guarantees the conveyance of examination materials [18]. The next phase outline is a non-compulsory phase that is planned if the scrutiny group is not acquainted with the material being analyzed and its experience. Then throughout the preparation stage, scrutiny colleagues independently get ready for their parts in the group. In the next phase i.e. scrutiny, the basic exercises of gathering are: the instigator gives appropriate scrutiny, clarification of per data as requirement, and the group recognizes failures that are characterized and recorded. The reason for next phase i.e. modification is to remedy deformities found throughout the assessment. Then comes the follow-up phase, which is a short gathering between the arbitrator and instigator to guarantee that all real issues found throughout the scrutiny have been adjusted and no optional imperfections have been presented. For verification the last phase is Scrutiny Analysis; in it the explored item is observed to access whether it is completely checked or still needs to experience an alternate stage; it is likewise investigated that the said imperfections have been altered [5].

After getting a well-defined system requirement specification, it again passes through a validation cycle. In the first phase business analysts and end clients survey the assembled requirements. In the event that the end client discovers any issues or anything absent in it, he characterizes it at this stage. After the exchange with the end user the business analyst makes an outline of the requirements to be demonstrated with animation [15]. Then the stakeholder chooses the prototype or animation framework. The main purpose for including stakeholder at this stage is that he can select the framework effectively understandable to him [6]. In the Develop Animation/ prototype stage the software engineer endeavors to fabricate prototype/ animation. At the end the end client, business analyst, software engineer and supervisor help in the evaluation stage. Stakeholder characterizes the issues or anything that is missing in the prototype, so again adjustment is carried out on it to fulfill client requirements [20].

In the end ontology concepts are characterized appropriately with a specific end goal to tackle ambiguity issue [3, 7]. It is supported with the help of expert opinions. Then relationships between distinctive classes are made with a specific end goal to fabricate ontology. The last step is to evaluate ontology with the help of a case study and experts review.

B. Specification

In the first step, after considering various requirement verification and validation techniques we divided their principle concepts (classes or entities) and properties (relationships) in plain structure [7, 8]. Differentiating the concepts and relating them will help us outline best possible outline of the conceptual model (concepts and relationships among them), for instance Arbitrator schedules activities. In this recommendation "Arbitrator" and "activities" are the concepts, although "schedules" is the property or relationship [19].

C. Conceptualization

Following the concepts clarified in table 2 a conceptual model is fabricated which indicates the concepts and relationship among those concepts [7]. A conceptual model has been intended for the differentiated concepts and properties, which will provide assistance while creating an ontology utilizing an instrument. After this, we have created the conceptual model to characterize the concepts excessively [8]. In this step (characterized by Fig 3) we will appropriately characterize the classes included in our requirement verification and validation process.

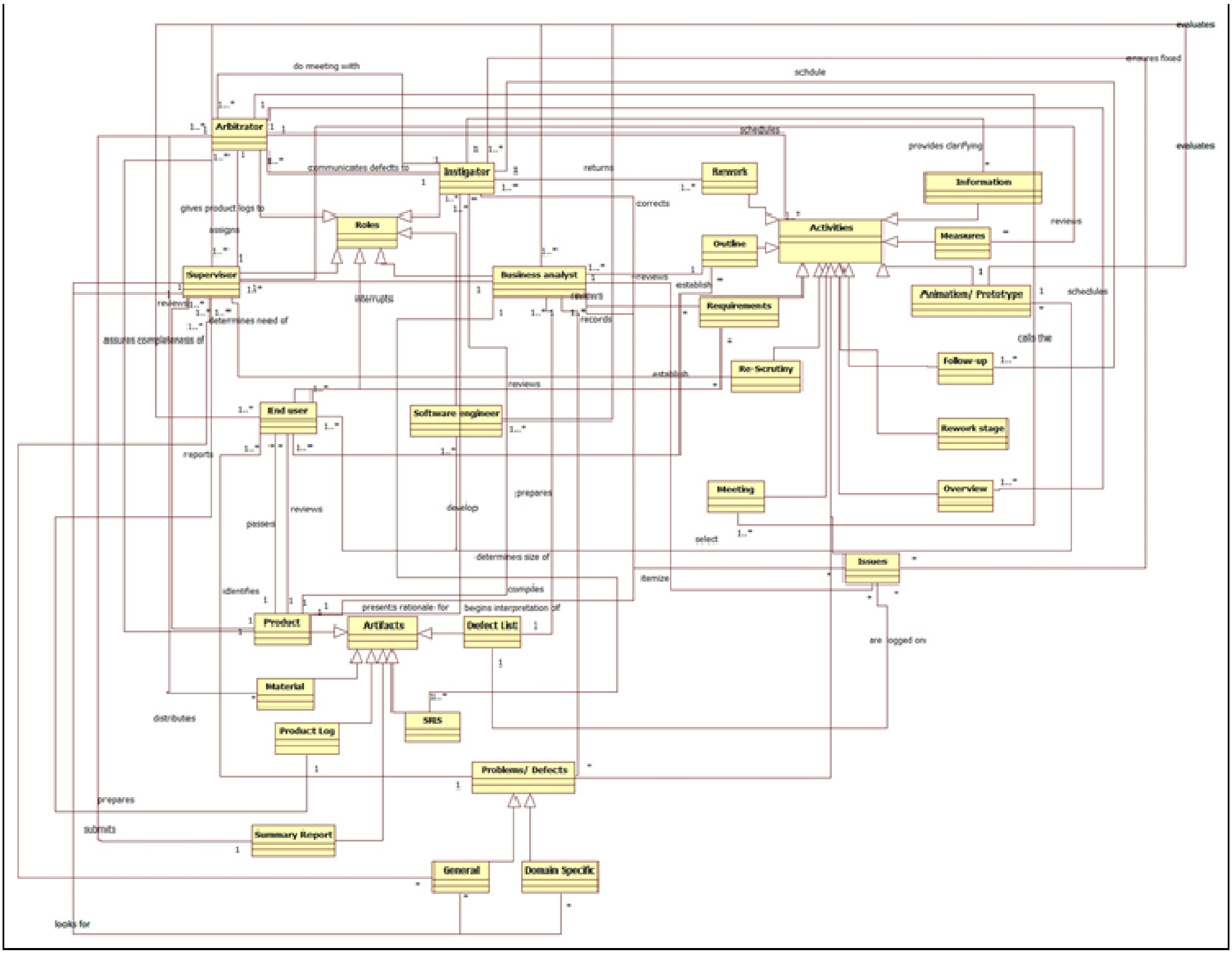


Figure 3 Higher Level Ontology Creation Concepts and Associations

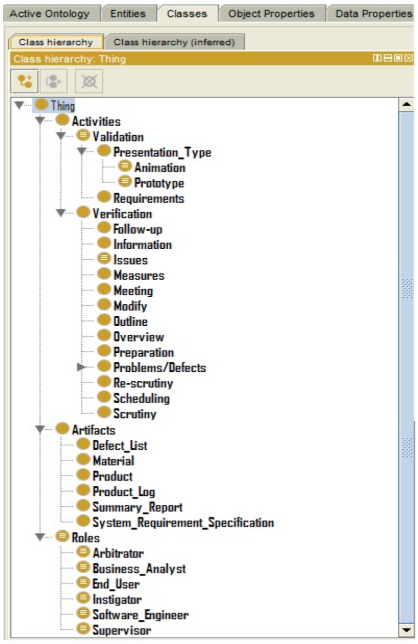
D. *Formalization*

We can exemplify our work in formalization as well; it is a more expressive way to represent knowledge. Table 2 shows the formalization. Here the conceptual model was transformed into a formal model. It described the domain described in the previous step in a more formal manner. Along with the object property; domain and range are also given in it.

Table 2 Formalization

Schedules (Arbitrator, Activities)
Assigns (Arbitrator, Roles)
Assurescompletenessof (Arbitrator , Product)
Determinessizeof (Instigator, Product)
Distributes (Arbitrator , Materials)
Schedule (Arbitrator, Overview)
Presentsrationalefor (Instigator, Product)
Reviews (Supervisor , Product)
Looksfor (Supervisor, General problems)
Looksfor (Supervisor, Specific domain problems)
Prepares (Supervisor, Product logs)
Givesproductlogsto (Supervisor , Arbitrator)
Beginsinterpretation (Business analyst , product)
Provides (Instigator, Clarifying information)
Identifies (End user , Defects)
Calls (Arbitrator, Meeting)
Interrupts (Supervisor, Business analyst)
Reports (Supervisor , Defect)
Prepares (Business analyst, Defect list)
Records (Business analyst, Unresolved/ open issues)
Itemize (Business analyst, Defect)
Areloggedon (Defects/ Open issues, Defect list)
Determinesneedof (Supervisor, Re-scrutiny)
Domeeting (Arbitrator, Instigator)
Schedule (Instigator, Follow up)
Corrects (Instigator, Defects)
Communicatesdefectsto (Arbitrator, Instigator)
Reviews (End user, Product)
Reviews (Supervisor, Measures)
Ensuresall (Instigator, Resolved issues)
Passes (End user, Product)
Submits (Arbitrator, Summary Report)
Returnsto (Instigator, Rework stage)
Reviews (Business analyst, Gathered requirements)
Reviews (End user, Gathered requirements)
Establish (Business analyst, Outline)
Establish (End user, Outline)
Selects (End user, Animation design)
Develops (Software engineer, Animation/ prototype)
Evaluates (Supervisor, Animation)
Evaluates (End user, Animation)
Evaluates (Business analyst, Animation)
Evaluates (Software engineer, Animation)
Compiles (Business Analyst, SRS)

Some rules are also applied to ontology like analyzed requirements document was passed to Requirement Verification phase, arbitrator carried out scheduling phase, domain specific and general problems were observed by the supervisor, final System requirement specification document was compiled by the business analyst, the product logs were given to the arbitrator before the start of scrutiny, the instigator was responsible for providing information about the product, the defect list contained open issues, end user selected the presentation type, summary report should be submitted by the arbitrator, on discovery of errors, instigator should return to rework stage and many more.



- Thing
  - Activities
    - Validation
      - Presentation_Type
        - Animation
        - Prototype
      - Requirements
    - Verification
      - Follow-up
      - Information
      - Issues
      - Measures
      - Meeting
      - Modify
      - Outline
      - Overview
      - Preparation
      - Problems/Defects
      - Re-scrutiny
      - Scheduling
      - Scrutiny
  - Artifacts
    - Defect_List
    - Material
    - Product
    - Product_Log
    - Summary_Report
    - System_Requirement_Specification
  - Roles
    - Arbitrator
    - Business_Analyst
    - End_User
    - Instigator
    - Software_Engineer
    - Supervisor

Figure 4 Classes of Requirement Verification and Validation Ontology

Classes are added in protégé which are shown in Fig 4.

E. Implementation

We have developed the ontology in Protégé, with the help of conceptual model. Ontology is shown in figure 5. It is developed by using OWL. For validation of our ontology, it will answer some competency questions, some of which are like:

i. Which role is significant in finalizing a well-documented SRS?

ii. What are the activities of the verification process?

iii. Which role helps to schedule a meeting?

iv. Do the roles involve the end user?

v. Summary Report of the whole verification process is submitted by which

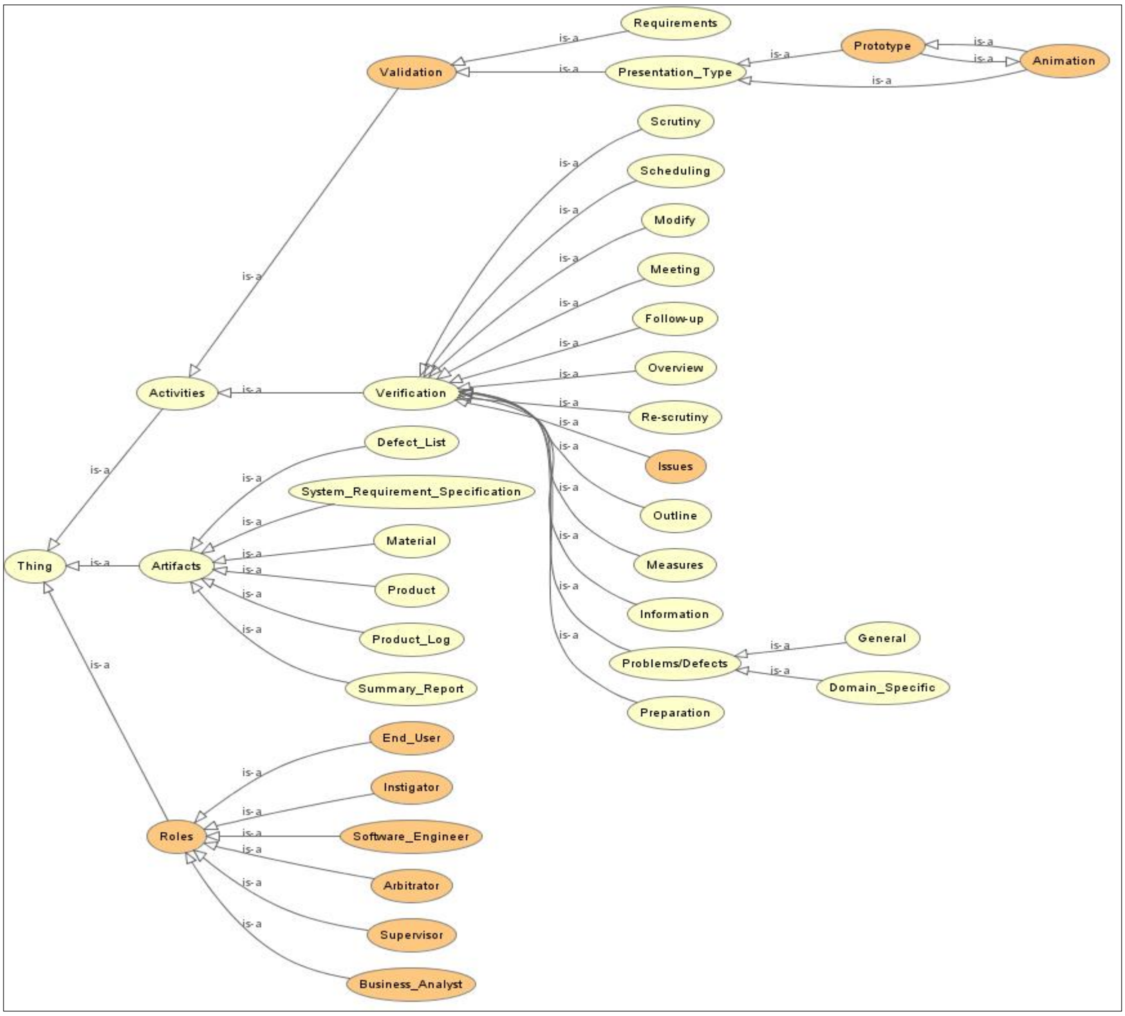
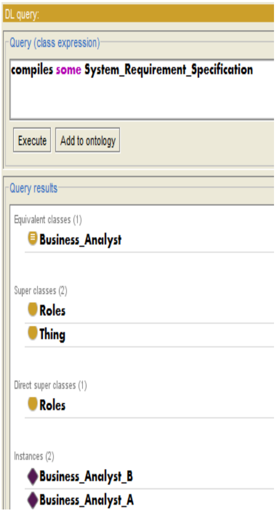


Figure 5 Requirement Verification and Validation ontology

V. EVALUATION

We have used case study for the evaluation of our work. Name of the company is not used in the case study due to the privacy policy. Company A is a well-known IT company, developed in 1994. Now it is one of the largest company providing IT solutions in Pakistan with nearly 1300 employees and presence in all major cities. Only three of their modules were used; which are Branch enrollment, Booth enrollment and New Profile Creation. Experts of the software domain analyzed the developed ontology on following parameters: Completeness, Consistency, Accuracy, Currency, Applicability, Plausibility, Clarity, Feasibility and Accessibility [22].Collectively on these modules the employees of the organization A important for the case study are: Business Analyst: Business Analyst A, Business Analyst B; Instigator: Instigator A, Instigator B; End User: End User A, End User B; Supervisor: Supervisor A; Arbitrator: Arbitrator A, Arbitrator B and Software Engineer: Software Engineer A, Software Engineer B.

Our developed ontology was successful in answering the competency questions. The answers to the competency questions can be gained from case study by applying the DL queries on the developed ontology. As answer of competency question 1 is shown in Fig 6 for our case study.



compiles some System_Requirement_Specification

Equivalent classes (1)
Business_Analyst

Super classes (2)
Roles
Thing

Direct super classes (1)
Roles

Instances (2)
Business_Analyst_B
Business_Analyst_A

Figure 6 Answers to Competency Question

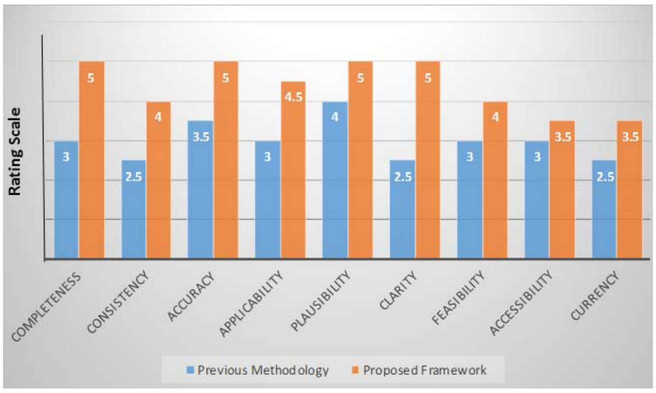


Figure 7 Comparison of Parameters for Previous Methodology and Proposed Framework

The comparison of old methodology and new methodology by the case study is shown in fig 7; on these parameters.

We have also used experts' reviews. They gave their views regarding the developed ontology, which is shown in fig 8. Limited detail is provided due to space issue. We also used a questionnaire to check experts' answers regarding these points. Thirteen different experts were chosen from different fields of IT. SE.S is a software engineer having seven year of experience in his field, QA.TL is a quality assurance team lead working since eight years in this field, QC.A shows quality control analyst having three years of experience, software developer having four years of experience is expressed by SD, business analyst team lead working since six years in this field is expressed as BA.TL, BA is used to show business analyst having four years of experience, DE shows deployment field related resource having three years of experience, software developer having ten years of experience is showed by SD.S, Lecturer having five years of teaching experience is showed by LE, QC.TL is representation of quality control team lead having six years of experience, lecturer having eight years of teaching experience is represented as LE.S, software engineer having five years of experience in his field is showed by SE and testing manager having eleven years of experience in his field is shown by TM.

Experts' opinion shows that the developed ontology gave better results for completeness, consistency, accuracy, applicability, plausibility, clarity, feasibility and accessibility [24]. It improved results.

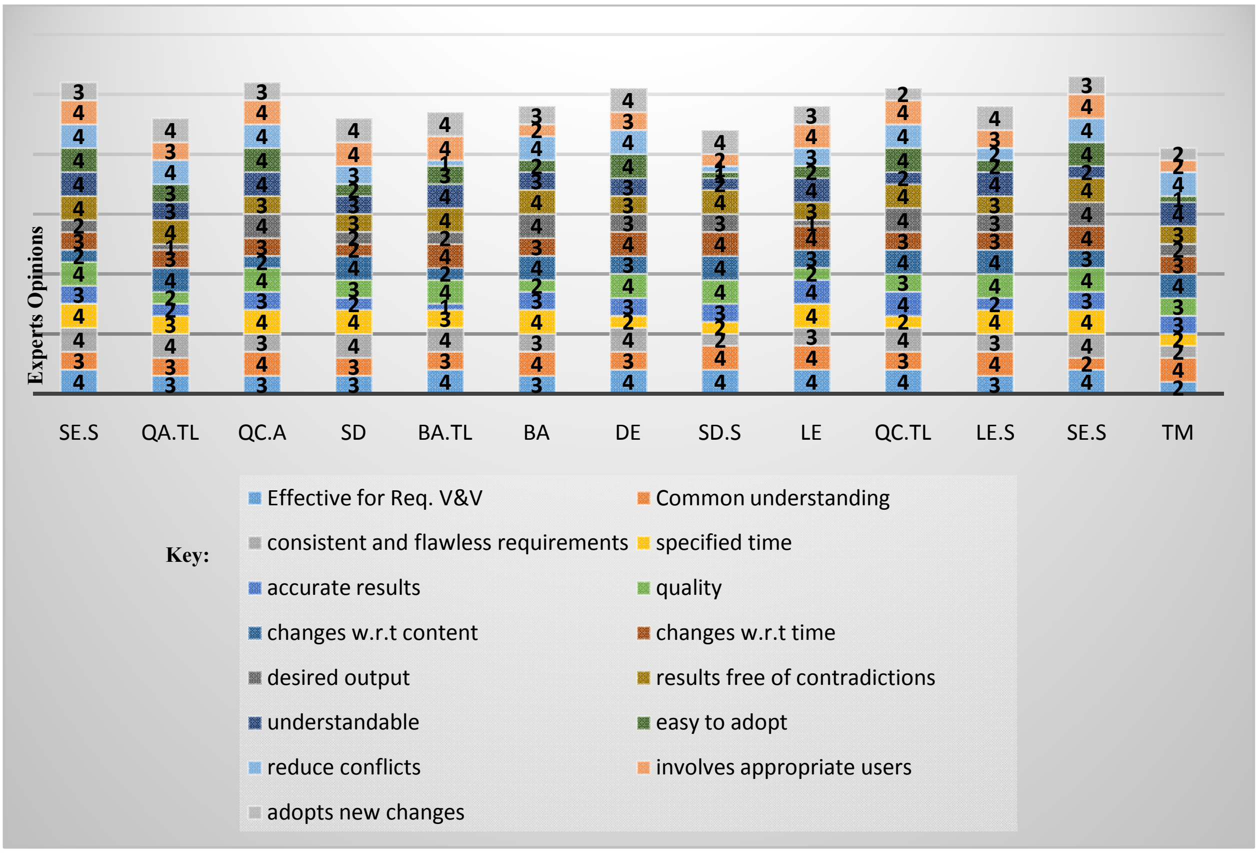


Figure 8 Participants' Opinions about the Requirement Verification and Validation Ontology

VI. CONCLUSION AND FUTURE WORK

Requirements verification and validation is a crucial activity for the project success [23]. It greatly affects the project cost, time and quality [2]. However, by implementing it through ontology it further improves the results. Very limited work is done in this field through ontology. Misunderstood requirements can cause ambiguity, incompleteness and inconsistency due to which the project cost and time increases. Ontological model is developed to solve these issues. A practice to reduce these issues due to improper requirement verification and validation is done.

On the success side, the normal is 16.2% of software projects that are finished on- time and on-plan. In the bigger organizations, the news is much more terrible: just 9% of their projects come in on-time and on-plan. And, actually when these projects are finished, a lot of people are close to a simple shadow of their unique specification requirements. Projects finished by the big American organizations have just roughly 42% of the initially proposed capacities. More Modest organizations improve more than expressed. An aggregate of 78.4% of their software projects will get sent with no less than 74.2% of their unique capabilities [2].

Although there are different approaches present for requirements verification and validation as shown in Table 1. The proposed approach, i.e. use of ontology adds a great value to enhance requirement verification and validation process. As with the help of a case study the reviews show that it improves the result in many aspects. It is clearly obvious that developing ontology for requirement verification and validation can provide better results.

In the near future we will further enhance our developed ontology. Hence, by using ontology an effort is made to improve the results.

ACKNOWLEDGEMENT

This research is partially supported by i-LaB Australia under The Technology for Humanity Project and Asia-Pacific World Congress on Computer Science and Engineering 2014.

REFERENCES

[1] Betty, H. and Atlee, M. "Research Directions in Requirements Engineering." 2007. FOSE '07 2007 Future of SOFTWARE Engineering, pp. 285-303.

[2] Bloch, M., Blumberg, S. and Laartz, J. "Delivering large-scale IT projects on time, on budget, and on value."2012.

[3] Castaneda, V., Ballejos,L., Laura, M.and Galli,M. "The Use of Ontologies in Requirements Engineering."Global Journal of Researches in Engineering. Vol. 10 Issue 6 (Ver 1.0) November 2010.

[4] Dargham, J. and Semaan, R. "A Navigational Web Requirements Validation through Animation".Internet and Web Applications and Services, ICIW '08. Third International Conference, Athens, 8-13 June 2008, pp. 211 – 216.

[5] Hashemi, N., Siew, T. and Ow, H. "A Scenario-Based Model to Improve the Quality of Software Inspection Process."Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference, Kuantan, 25-27 Sept. 2012, pp. 194 – 198.

[6] Jwo, J. and Cheng, Y. "Pseudo Software: a New Concept for Iterative Requirement Development and Validation."Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific, Aichi, 4-7 Dec. 2007, pp.105 – 111.

[7] Khatoon, A., Hafeez, Y., Asghar, S. and Ali, T. "An Ontological Framework for Requirement Change Management in Distributed Environment."2014.

[8] Khatoon, A., Hafeez, Y., Azeem, M., Naz, H. and Nazir, S."Requirement Change Management for Global Software Development using Ontology." Emerging Technologies (ICET), 2013 IEEE 9th International Conference Islamabad, 9-10 Dec. 2013, pp. 1–6.

[9] Kof, L., Gacitua, R., Rouncefield, M. and Sawyer, P. “Ontology and Model Alignment as a Means for Requirements Validation.”Semantic Computing (ICSC), 2010 IEEE Fourth International Conference, Pittsburgh, PA, 22-24 Sept. 2010,pp. 46 – 51.

[10] Kroha, P., Janetzko, R. and Labra, J. “Ontologies in Checking for Inconsistency of Requirements Specification.”Advances in Semantic Processing, 2009. SEMAPRO '09. Third International Conference Sliema, 11-16 Oct. 2009, pp. 32-37.

[11] Magnini, B. and Speranza, M. “Merging Global and Specialized Linguistic Ontologies.”2006.

[12] Mokos, K., Meditskos, G., Katsaros, P., Bassiliades, N. and Vasiliades, V. “Ontology-based Model Driven Engineering for Safety Verification.”Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference, Lille, 1-3 Sept. 2010, pp. 47 – 54.

[13] Merz, F., Hendrik, C., Gorges, T. and Bosch, R. “Abstract Testing: Connecting Source Code Verification with Requirements.”Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference Porto, Sept. 29 2010-Oct. 2 2010, pp.89 – 96.

[14] Ping, M., Feiyan, M. and Ming, Y. “A Knowledge-based Validation Method of Complex Simulation Models.”Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference (Volume: 2) Sanya, 23-24 Oct. 2010, pp. 3 - 6

[15] Raja, U. “Empirical Studies of Requirements Validation Techniques.”Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference, Karachi, 17-18 Feb. 2009, pp. 1 – 9.

[16] Sharma, R. and Biswas, K. “Using Norm Analysis Patterns for Automated Requirements Validation.”Requirements Patterns (RePa), 2012 IEEE Second International Workshop, Chicago, IL, 24-24 Sept. 2012, pp. 23 – 28.

[17] Sharma, A. and Kushwaha, D. “Complexity Measure based on Requirement Engineering Document and its Validation.”Computer and Communication Technology (ICCCT), 2010 International Conference Allahabad, Uttar Pradesh, 17-19 Sept. 2010, pp. 608 – 615.

[18] Schaefer, I. and Poetzsch, A. “Compositional Reasoning in Model Based Verification of Adaptive Embedded Systems.”Software Engineering and Formal Methods, 2008. SEFM '08. Sixth IEEE International Conference, Cape Town, 10-14 Nov. 2008, pp. 95 – 104.

[19] Serpeloni, F., Moraes, R. and Bonacin, R. “A Semi-Automated Approach to Validate Ontology Mappings.”New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference, Tozeur, May 31 2010-June 2 2010, pp. 309 – 313.

[20] Sharma, A. and Singh, D. “A Metric Suite for Early Estimation of Software Testing Effort using Requirement Engineering Document and its Validation.Computer and Communication Technology (ICCCT), 2011 2nd International Conference, Allahabad, 15-17 Sept. 2011, pp. 373 - 378.

[21] Shunxin, L. and Leijun, S. “Requirements Engineering Based on Domain Ontology.”Information Science and Management Engineering (ISME), 2010 International Conference (Volume: 1) Xi'an, 7-8 Aug. 2010.

[22] Thunkijjanukij, A. 2009. “Ontology Development for Agricultural Research Knowledge Management.”

[23] Tomar, P. and Singh, N. 2010. “Verification & Validation of Components with New X Component-Based Model.”Software Technology and Engineering (ICSTE), 2010 2nd International Conference (Volume: 2), San Juan, PR, 3-5 Oct. 2010, V2-365 - V2-371.

[24] Vallespir, D. and Herbert, J. “Effectiveness and Cost of Verification Techniques Preliminary Conclusions on Five Techniques.” 2009.

[25] Wenzel, S., Rabe, M. and Spieckermann, S. “Verification and Validation Activities within a New Procedure Model for V&V in Production and Logistics Simulation.”Simulation Conference (WSC), Proceedings of the 2009 Winter, Austin, TX, 13-16 Dec. 2009, pp. 2509 – 2519.

[26] Zhang, L. and Ye, C. “Semantic-based Requirements Analysis and Verification.”Electronics and Information Engineering (ICEIE), 2010 International Conference (Volume: 1), Kyoto, 1-3 Aug. 2010 V1-241 - V1-246.