ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

Cent CORE: Centralized Cloud Oriented Requirement Engineering Strategy for Tracking and Elicitation of Dynamic Requirements

Gerard Deepak[1], Navya Prakash[2], Reevan Chris Pinto[3], Sheeba Priyadarshini J[4]

M.E. Student, Dept. of Computer Science and Engineering, UVCE, Bangalore University, Bangalore, India [1]

M.Tech Student, Department of Studies in Computer Science, University of Mysore, Mysore, India [2]

B.E. Student, Dept. of Computer Science and Engineering, UVCE, Bangalore University, Bangalore, India[3]

M.Sc Student, Department of Computer Science, St. Josephs College, Bangalore, India [4]

ABSTRACT: Requirement Engineering is one of the most important stages of Software Engineering. Eliciting requirements is highly critical and a complex process as the software end product totally depends on the quality of requirements that were collected. The property of the requirements is dynamic that keeps changing and constantly evolving. The Traditional Strategies for Requirement Engineering lacked organization and change management was entirely manual which consumed a lot of time and skilled labor. A centralized strategy for Elicitation of Dynamic Requirements using the concept of Requirement Cloud is proposed with high level of organization and structuring. A novel idea of using Cloud Storage Service for Requirement Engineering is implemented using a heuristics approach. Change management is incorporated and a few activities like requirements document generation is automated in this approach. Finally a survey between the Traditional Requirement Engineering and Proposed Cloud Methodology is conducted to prove the proposed methodology is better than the traditional strategies of Requirement Engineering.

KEYWORDS: Cloud Based RE ,Dynamic Requirements,Requirement Cloud, Requirement Elicitation, Requirement Engineering, Requirements Validation, Stakeholders

I. INTRODUCTION

Requirement Engineering is one of the most primary and yet the most important stage of Software Engineering. Requirement Engineering is never a single step process but is rather recursive and time consuming.Though there are several ways and techniques for requirement engineering but there always exists a conflict between the stakeholders involved in requirement collection and analysis. The stakeholders involved in requirement engineering could be as simple as the client and the developer or it could be as complex as involving a number of clients and a development team or an organization. In reality the most conventional cases involve the client and well as the developer up to be an organization, thus mostly Business-Business interaction takes place for engineering requirements. Since the interaction is between businesses, it must be professional and also must be kept short and simple.
A proper consensus must be obtained between the stakeholders for requirement engineering. The most prominent criteria for requirement engineering are the correct interpretation of the requirements such that the end product which is the engineered software is in compliance with the requirements that the client has specified.

Motivation: The process of requirement engineering is highly dynamic as the requirements keeps changing and evolving. A constant monitoring of the changes which the requirements undergo is required. Proper requirement management for effective software design is a pure mandate. The design of software must be a quicker stage in the SDLC life cycle and to achieve this in the software development life cycle model, the clarity of requirements must be very high. A model of requirement elicitation or requirements inception [1] is a requisite. Security awareness is also a major concern for requirement collection which otherwise can lead to several contradictions and problems.

Contribution: The model that is implemented here overcomes the problem of dynamism of software requirements. A proper channel for change tracking and management of requirements is achieved. A model for requirement validation is included in the final system. The incorporation of Cloud Computing in Requirement Engineering is achieved to build a centralized system for requirement gathering. A practical experiment of real life scenario of requirement engineering was conducted in order to evaluate the need for an organized system for effective requirement elicitation.

Organization: The organization of this paper is as follows. The section 2 provides a brief overview of related research work. Sections 3 and 4 present the Problem Definitions and Proposed Architecture respectively. The Implementation is presented the section 5. The Survey Analysis and Results is discussedin section 6.Section 7 presents the Heuristics approach. The paper is concluded in section 8 where the future work is also discussed.

II. RELATED WORK

Daniel et al., [2] proposed an Artifact based requirement engineering where the Requirement engineering stakeholders concentrate on the artifacts themselves rather than the methodologies followed in gathering those artifacts. The artifact based requirement engineering deviates from the activity based requirement engineering and it depicts an approach for Requirement Engineering.Michael et al., [3] have analyzed the process of requirements and have studied the gap between the lack of proper requirements engineering and have analyzed the effect of improper methodologies of requirement engineering in producing a defective software with a mismatch between the needs and the outcome of the software product engineered. The results are depicted with five case studies in order to understand the scenario of requirements engineering.

Axel et al., [4] describes the handling of obstacles in Goal Oriented Requirements Engineering in which the requirements are visualized as goals or objectives which are refined based on the software systems specifications tested and validated across major constraints in order to achieve the best outcome of the software that is developed.Venketesh et al., [5] describes the novel strategy for risk analysis in goal oriented requirements engineering by using selective approach for the software goals using Tropos Model for Goal Selection for impactful and highly effective requirement engineering. Taiseera et al., [6] have proposed an ontological approach for requirements elicitation and prioritization to regulate and in turn maintain the quality of requirements. A quality driven requirements engineering model is proposed using ElicitO framework for requirement quality control. Joaquin Nicolas et al., [7] proposed a systematic review literature for generating requirements in the textual format from the analysis of software engineering models such that the gap between the Graphical Models of Software Engineering and Text Oriented Requirements is bridged in this approach.

III. PROBLEM DEFINITIONS

Requirement Engineering is a highly challenging and a recursive process which involves proper interaction of the stakeholders in order to achieve consensus and stability in requirements gathered. The requirements for the same product may differ from individual to individual. Every individual has their own style of defining or describing requirements. This makes the process of requirement engineering more complex and time consuming. The clarity in the requirement definition is totally neglected and the defined requirements are never available centrally in the conventional requirement engineering methodologies. The four major problems requirement engineering can be summarized as:

3.1 Dynamism of the requirements with changing perspectives of the stakeholders

The requirements are highly dynamic and volatile. This is because the final software product is visualized in several perspectives with every individual who are involved in requirement engineering. The developer may have a technology oriented perspective whereas the client may have a different perspective which more possibly can be visualized as a behavioral perspective or a business oriented perspective.

3.2 Lack of a centralized requirement engineering system for the interaction of stakeholders

If the model for requirement engineering was based on a centralized model where the stakeholders take part in the requirement engineering process through collaboration and review,the defined requirements can be implemented in the resulting end software with a lot of ease without any conflict. A high degree of Transparency among the stakeholders of requirement definition will set in. This would in turn add to the final quality of the resulting software. Due to a lack of centralization, there is no proper collaborative platform for Requirement Engineering.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

3.3 Absence of security in the requirement modeling systems

Collection of requirements is quite a basic task but still it requires high end security measures without which requirements can be intruded by third party agencies and they may tend to engineer a better software model for the client involved. Thus the authenticity of the stakeholders must be verified at least by a basic first line security mechanism.

3.4 Lack of a proper change monitoring and management mechanism in the requirement engineering automation models

An efficient model for the tracking of the changes which the requirements undergo right from the stage of their conception to the stage where they are actually finalized for their inclusion into the software that is to be developed.

IV. PROPOSED ARCHITECTURE

The proposed architecture of the systemfor centralized cloud oriented requirement engineering is depicted in the Figure 1 where in initially secure access mechanism is instituted for the stakeholders to log into the system. Secure access is mandatory as without security, the identity of the requirement contributors, ie, the stakeholders can be masked. In order to incorporate security to the system, a secure access scheme is introduced into the Dynamic Requirement Engineering Model. Once the stakeholders are authenticated into the system, then approval of stakeholders is done .Stake holders who are designated to define requirements are uniquely identified by an Id with which every individual stakeholder can be referred till the finalization of requirements take place. The generation of unique Id for individual participant in the process of requirement definition and validation keeps the system free from illegal and fake stakeholders thus ensuring more privacy. This also facilitates easy reviewing process of the defined requirements.

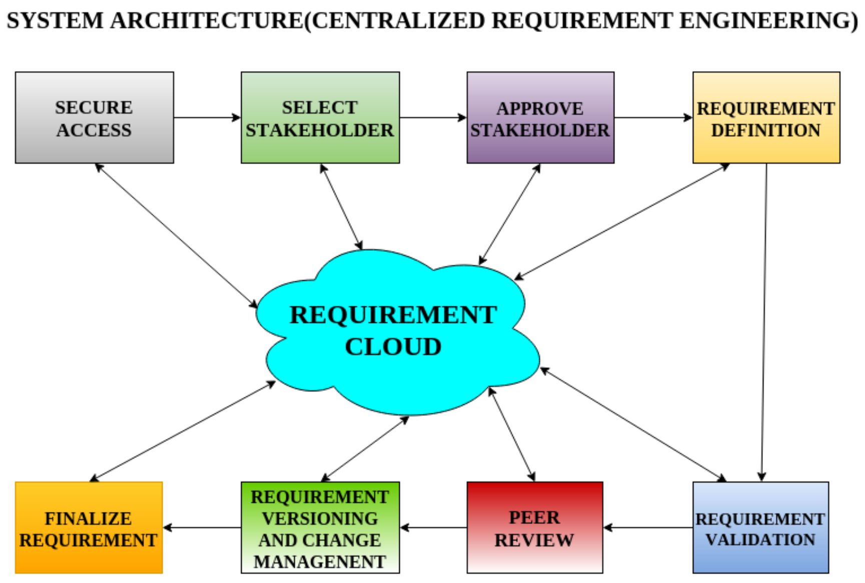

Figure 1: Proposed System Architecture

When the stakeholders are identified to be authentic and when their identities are verified, the stake holders are allowed to define requirements. The requirements must be defined in a highly precise manner; the system then allowsdetailed definition and description of the requirements. The requirements defined are directly stored in the cloud. The usage of Cloud for defining and describing requirements enables the provision for the centralization of requirement engineering. The facilitation of a centralized cloud for requirement engineering enables transparency for requirement definition between participating individual stakeholders. The requirements defined can be reviewed dynamically as they are being defined by other stakeholders. Every requirement that is defined needs to undergo a review and a validation process.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

Review of requirement can be done either by the client participants or by the development organization participants but validation of requirements must be done by individuals who are from the developers side and a technically sound. Requirement Validation is based on the following primary factors:

- The requirement defined is realizable or not.
- The requirement defined is in the context of the software to be developed or not.
- The implementation of the requirement is critical or not.
- The requirement implementation fits into the allocated budget of the project or not.

Requirement Validation is followed by prioritization of requirement where the priority *high* is assigned for requirements that need to be compulsorily implemented where as a priority *medium* is assigned for the requirements that have lesser priority to be implemented; priority*low* is assigned to the requirements that may or may not be implemented. The validated requirement undergoes a peer review process where a dual review phenomenon for requirements is followed until the requirement is finalized. Once the Requirement is finalized it undergoes versioning for the first time. Until this step, versioning is not implemented, ie, dynamism is not recorded but if there is a conflict in the requirements after the first versioning and peer review, then any change from now needs to be versioned every time a new change is incorporated. This is the underlying principle for this system of Dynamic Requirement Engineering. The variations in the versioning of the requirements are the key basis for Change tracking and Management. Every major change and minor change once the finalization of requirements is done needs to be tracked. The incorporation of change management and tracking is another contribution to the concept of requirement engineering. Since the requirements keep changing dynamically, this model can also be referred to as requirement re-engineering. At the end of peer-review and rigorous changes in the requirements, the final requirements are gathered.

V. IMPLEMENTATION

The proposed architecture is implemented as a Cloud based model (CBM) where a Cloud Space was rented in order to achieve the best in-class look and feel of realizing a centralized system. The Cloud which is a universal storage for the requirements is visualized as a storage and analysis area for the requirements defined. The Cloud which comprises of the requirements is termed as the Requirement Cloud since the cloud storage holds all the analytical procedures for requirement storage. The primary functionalities for the requirement cloud are to store the requirements and provide a centralized platform for requirements prioritization, requirement validation and a centralized review.A distributed computing paradigm is adopted where the individual users are authenticated through a web based remote access mechanism for accessing into the cloud. Each participator involved in definition and analysis of requirements is assigned with Unique Id for identification for future reference in order to prevent ambiguity and redundancy of participators. The Unique Id generated is prefixed with CLI for client side participators and suffixed with DEV for developer side stakeholders, thus providing a clear cut demarcation between the client end stakeholders and developer end stakeholders. The visibility of the changes in the defined requirements is in the scope of all the logged in stakeholders for a specific project where individual requirement engineering stakeholders are allowed to enter the project Id for which they are supposed to define the requirements. The Unique Id of the client side stakeholder comprises of the following details *(CLI, projectId, first_name_participant, unique key)* and the Unique Id for developer side stakeholder looks like *(unique key, projectId, first_name_participant, DEV)*.
The finalized requirements after the peer review is automatically extracted into the XML structured finalized Requirements Document that is maintained separately. If a stake holder is not peer-reviewing a requirement within 8 hours, an e-mail is dispatched to the stakeholder to either give a review or interact with the system and intimate the system that stakeholder has no reviews. The Prioritization of the requirements is done again by a re-review of the requirements by the stakeholders developer side alone. This time, the client side stakeholders are not allowed to re-review and the new validated requirements with respect to the project budget and the requirement feasibility, a new requirement feasibility report is generated again as an XML file. Finally for the requirement prioritization review the client side stakeholders are allowed to analyze the requirement and vote for the priorities. Based on individual prioritization score, the requirements are prioritized and the finalized priority reordered requirement document is generated and mailed to every individual stakeholder. Thus there is a clear visibility of requirements and their priorities for both the client side and developer side stakeholders. Versioning is made simple and there is no concept of major changes and minor changes whereas every change is considered as a new change and the versioning is done as V 1.0,V 2.0, V 3.0, . . . , V n.0, where n is any positive integral number from 1,2,3….. .In this approach, the versioning of the

requirements is implemented and the finalized requirements document is not versioned. The last and final document is addressed and automatically watermarked as "FINALIZED REQUIREMENTS".

VI. SURVEY AND ANALYSIS RESULTS

In order to compare the traditional requirement engineering methodologies with the Centralized Cloud Model for Dynamic Requirement Engineering, an experiment along with a comprehensive survey was conducted. To take part in the experimental survey eight young and energetic software development literate persons were considered. They were segregated into two teams such that four in each group. The first group served as client stakeholders whereas the second served as developers. The experiment was divided into two phases. Phase 1 comprised of traditional requirement engineering techniques through e-mails, phone calls, Skype and videoconferencing. A detailed design of the front end of a web page was given to the clients where they had to describe them as requirements to the developers' team and they had to implement it. It actually took 5 long hours approximately for the client team and the developer team to come in agreement and develop the Web Page.
On the contrary, a much complicated front end with the database design prototype was given to the client team for Phase 2 of the experiment using the Centralized Cloud Requirement Engineering Model. They were asked to register into the developed system which was hosted into a series of systems that were well connected. Now the entire requirement engineering process was centralized such that they dynamically reviewed the requirements then and there. This led to more clarity and facilitated better understanding of the requirements to the stake holders. The requirements were also validated by a peer review mechanism and further more prioritization of requirements was done until they were versioned before their finalization. All the stakeholders were able to relate the requirements and were able to track and manage changes due to their full version control. The clear definition of the requirements enabled the stakeholders to have a proper vision of the requirements objectives and their purpose for including them into the software. Henceforth, a software model which was twice difficult than the Phase 1 web page was developed in approximately 1 hour 22 minutes.This saved almost around 3 hours 30 minutes for a software that is twice difficult than the one that is implemented in the phase 1.A survey was further conducted based on question and answer format where the stakeholders involved in the requirement engineering were asked a few questions and their experience in using the centralized cloud model Vs. Traditional Requirement Engineering Models. The survey results were plotted against several parameters that were considered and the results are depicted in Figure 2.

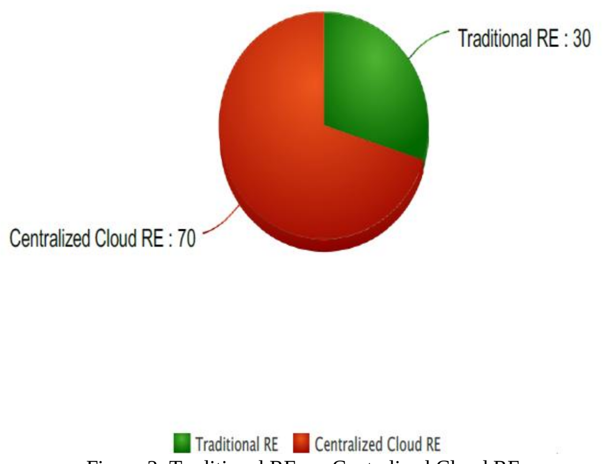

Figure 2: Traditional RE vs. Centralized Cloud RE

The results were that approximately 30% of the participators in both the phases of requirement engineering said that traditional method for requirement engineering was well suited for them. 70% of the survey participators voted for the migration to Cloud Based Requirement Engineering paradigm as they felt constant monitoring of the requirements with loads of time saving capability is preferred rather than the age old model of conventional requirement engineering. The

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

Requirement Engineering Methodology vs. Impact Distribution is depicted in the Figure 3. The impact distribution of the Centralized Cloud Based Requirement Engineering Paradigm is definitely four times better than that of the traditional Requirement Engineering Paradigms. This definitely proves that the proposed Cloud Based Requirement Engineering Model is the best suited model for Software Requirement Engineering that is well associated with several advantages.

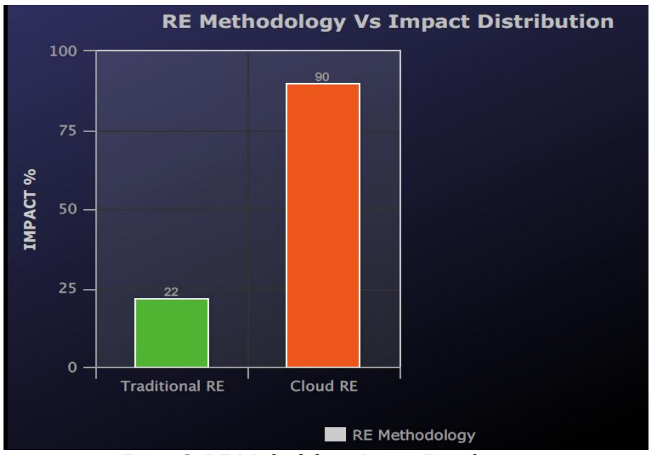

Figure 3: RE Methodology-Impact Distribution

The graph in the Figure 3 depicts the requirement engineering Methodology-Impact Distribution where in the impact of overall phenomenon of Requirement Engineering is plotted against the strategies involved in Requirement Engineering.The overall impact is found out to be 22% for Traditional Requirement Engineering Strategy whereas the impact for Cloud Requirement Engineering strategy is 90% owing to the fact that a fully centralized approach involving tracking of the changes in the requirement is a much better and is a highly significant approach in correlation to the traditional conventional model for requirement engineering.

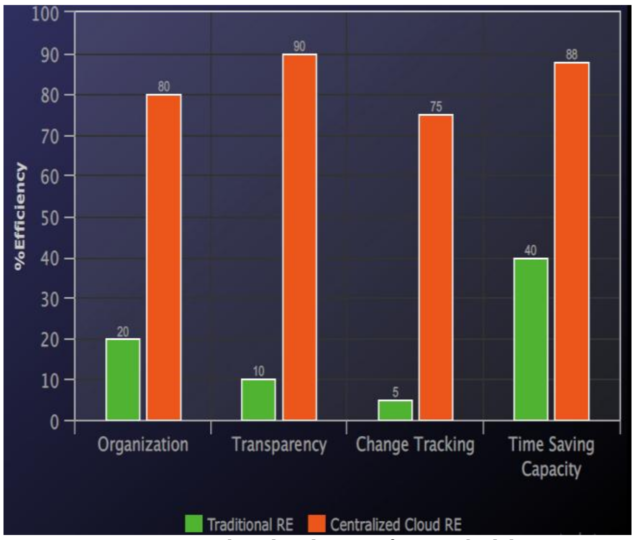

Figure 4: Feature based evaluation of RE Methodologies

Several features were considered like Organization, Transparency, Change Tracking and Time Saving Capacity and the efficiency was evaluated for both Traditional Requirements Engineering and Centralized Cloud RE and is depicted in Figure 4.It is found that, the feature Organization that is involved in Requirement Engineering is 20% for traditional RE while 80% for Centralized Cloud RE. Similarly the Transparency of the requirements between the stakeholders is 10 %

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

and 90% for Traditional RE and Centralized Cloud RE respectively. The Change tracking is almost absent and is completely manual in the Traditional RE approach contributing to only 5% where as in Centralized Cloud approach Change Tracking constitutes to 75%. Finally the time saving capacity is 40% for Traditional RE where as it hits 88% for Centralized Cloud RE.From Figure 4, it is clearly evident that there is a huge difference in Efficiency for Various Features of Requirement Engineering Methodologies for Traditional RE and Centralized Cloud RE. The average difference of efficiency % between the Cloud RE and Traditional RE is 64.5 which can be interpreted as the Centralized Cloud RE Methodology is 64.5 times better than the Traditional RE Methodology.

VII. HEURISTICS APPROACH

A Heuristic Approach is adopted in the Cent CORE as depicted in Figure 5 mainly in order to arrive at the solutions quite early. Heuristics is followed here for Requirement Engineering Elicitation. The Heuristics starts with Requirement Definition and then allows Validation of Requirements with multiple reviews until a proper consensus is achieved until requirements validation. Prioritization of the Requirements is also allowed in the Heuristic and furthermore the heuristics allows versioning. The versioning is highly recursive until the finalization of requirements takes place. The heuristics ends only if the requirements are finalized. Heuristics is a continuously changing model until the dynamism of the requirements is achieved. The Heuristics is followed for every requirement that is defined in the entire process of Requirements Engineering.

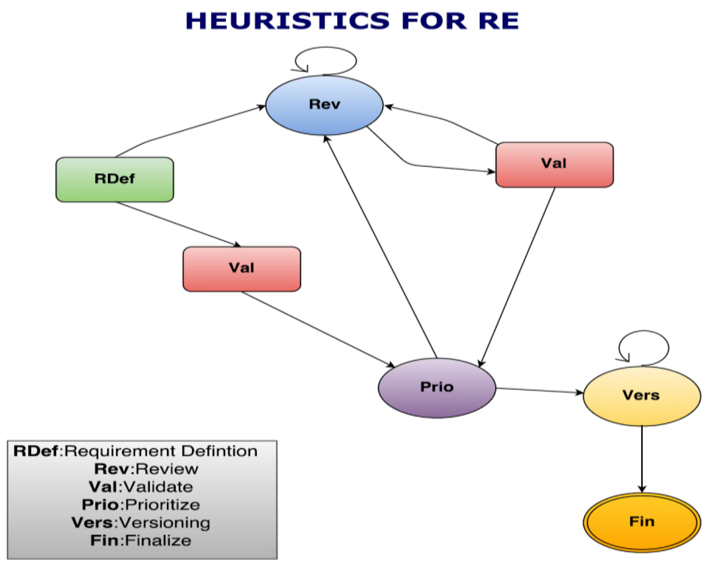

Figure 5: Heuristics for Cent CORE

There is a high degree of randomness and dynamism in defining the requirements in the process of Requirement Elicitation and Requirement Engineering. In order to compensate, the randomness a Heuristic Approach is followed rather than a traditional Algorithmic Approach. Heuristics is the best suited paradigm to be followed for Requirement Engineering as the states of the requirements changes constantly until they are validated, prioritized and finalized.

VIII. CONCLUSIONS AND FUTURE WORK

A strategy for Requirement Engineering over the cloud is implemented in order to provide a Centralized Platform and Clear Transparency of the Stakeholders involved in Requirement Engineering. In this Methodology not just a simple strategy for Requirement Collection is provided but also a platform for Requirement Validation with a constant peer review.The strategy that is implemented here further provides automated requirements prioritization based on the interaction history of the stakeholders while prioritizing the requirements. The generation of Requirements Document is also automated and individual stakeholders are intimated with an e-mail of the finalized requirements. Henceforth a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

high degree of clarity is achieved among the stakeholders to come in proper consensus with the requirements in this approach. The Dynamism in the requirements is tracked easily through centralized versioning control strategy that is made simple in our implementation. A secure access mechanism authenticates the participants in the Requirement Engineering process.Henceforth, our system also has quite a significant measure of security awareness. The survey based evaluation clearly infers that the Proposed Cloud Oriented Requirement Engineering Methodology-Cent CORE overtakes the previous methodology of Traditional RE in almost every aspect. The systems efficiency is evaluated against four prominent features namely Organization, Transparency, Change Tracking and Time Saving Capacity. The results prove that the proposed Cloud RE system is definitely much better than the existing Traditional RE Methodology.

REFERENCES

[1]Wikipedia, "Requirements Engineering (RE)" https://en.wikipedia.org/wiki/Requirements_engineering, Last Accessed on 18th Oct 2015.
[2] Daniel Mendez Fernandez, Birgit Penzenstadler, "Artefact-Based Requirements Engineering: the AMDiRE approach" in *Requirements Engineering Springer-Verlag*, London 2014, pp. 405-434.
[3] Michael U, Tony G, Robert F and Eriks K, "Assessing Requirements Engineering and software test alignment-Five case studies," in *The Journal of Systems and Software(109)*, 2015, pp. 62-67.
[4] Axel van L, "Handling obstacles in goal oriented Requirements Engineering," in IEEE Transactions of Software Engineering, Vol. 26 No. 10, Oct 2000, pp. 978–1004.
[5] K Venketesh S, P V Kumar , "A Method to Risk Analysis in Requirement Engineering Through Optimized Goal Selection Tropos Goal Layer, *Journal for Theoretical and Applied Information Technology*, Vol. 61, No.2, 2014,pp. 270-280.
[6] Taiseera H A B, Pedro R F S and Pericles L, "Eliciting and Prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool" in the *Article of Expert Systems*, May 2013, Vol. 30, No.2, Wiley Publishing Limited, pp. 129-151.
[7] Joaquin N, Ambrosio T, "On the Generation of Requirement Specification from Software Engineering Models: A Systematic Literature Review," in *Journal of Information and Software Technology*, 2001, pp. 1291-1307.