Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 6 Issue 9, September - 2019

A Participatory Requirement Engineering Process (PREP) Model for Software Development Projects

Akpasam Joseph Ekanem

Department of Electrical/Electronic Engineering, Akwa Ibom State University Mkpat Enin, Akwa Ibom State, Nigeria

ikechukwu4per@yahoo.com

Udegbe Ikechukwu Valentine
Department of Computer Science Imo State Polytechnic Umuagwo, Imo State Nigeria

Ibe Perpetual Nwakaego
Department of Computer Science Imo State Polytechnic Umuagwo, Imo State Nigeria

***Abstract*— Requirement engineering is one of the most critical and fundamental phases in software development upon which other phases in software development life cycle depend on. In this paper, The a participatory requirement engineering process model is proposed. The model is mearnt to handle some of the problems inherent in the existing requirement engineering process models. These includes requirement pre-processing, risk management and application specific elicitation techniques. It is iterative in nature for better requirement engineering and can be used in larger as well as complex software development process where requirements changes continuously. The details of the components of the participatory requirement engineering process model is presented along with the tools used in modelling the process components.**

Keywords— Requirement Engineering, Software Development, Requirement Elicitation Software Development Life Cycle, Requirement Pre-Processing, Risk Management

I. INTRODUCTION

Requirement engineering phase is one of the most essential phases for an effective software development [1] .According to [2,3], Requirement engineering is recognized as the first phase of software engineering process and is considered as one of the key tasks in software development. Requirement engineering is a crucial factor which influences software product quality and productivity [4].Also, [5,6] reported that ambiguous requirement and product defect are the top reasons for software project failure. According to [7], "Requirements Engineering is a sub discipline of systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems". Also [8] pointed out the criticality of Requirement engineering in his work; that the hardest part of building a software system is deciding precisely what to build, that no other part of the work cripples the resulting system if done wrong and no other part is more difficult to rectify later like Requirement engineering.

Similarly, [9,10] identified Requirement engineering as the most critical and fundamental phase in software development and that other phases in software development life cycle depends on requirement engineering phase. [11] in his research showed that requirement errors produces 60% of errors in critical systems and a survey conducted by [12] in European companies showed that more than 60% of them considered Requirement engineering problem as very significant. These established facts highlights requirements engineering as a critical phase in software development projects. Nowadays, many requirement engineering techniques and models are available for software developers. Furthermore, some of the techniques which are proven successful to one system may not work well for another. It means that methods and techniques selection can be difficult and poor selection of the requisite methods and techniques may lead to software system failure. Also the requirements engineering process can be affected by changes in business environment and risk management. However, in situations where time to market is more critical to the release of the software than the software of quality, requirement engineering models must provide appropriate guidelines for developers to help them produce high quality software in a timely manner which ultimately leads to a successful software project development[13].So as to develop high quality software in timely manner, while trying to avoid traditional requirement engineering problems, requirements engineering process along with appropriate guidelines is inevitably required [13].

In this paper, a Participatory Requirement Engineering Process (PREP) Model is proposed. The proposed PREP model will afford the system stakeholders opportunity to participate effectively in the software design process and will overcome some of the problems inherent in the existing models such as: Pre-processing activity; application specific requirement elicitation techniques; and risk identification, assessment and management problems.

II. REVIEW OF EXISTING REQUIREMENT ENGINEERING PROCESS MODELS.

A. Spiral Model of Requirement Engineering Process

This model, as recommended by [14] has the key element "Spiral". Each spiral has four major variants which are: requirements elicitation, requirement analysis and negotiation, requirement documentations and requirement validation. The main objective of the model is to overcome the problems that affect the quality and cost of the project which occur in various stages of software development [15].The strength and weakness of the model as pinpointed by [16] are:

Strength(s):

- It incorporates client feedback and facilitates active user contribution.
- Fault can be detected early in software development
- It supports an effective risk management strategy.
- It supports effective reverse engineering process.

Weaknesses:

- It does not compute efforts in requirement phase.
- There is no support for requirements pre-processing.
- There is no support for concept of requirement prioritization.
- There is no criteria of using application specific elicitation technique

B. Bee Hive Model

In this work, Application Domain, organizational factors, Market, scale check, safety and security are some areas in which background research was carried out [17].The model increases the swiftness and examines the actual time required for eliciting the requirements from the stakeholders for designing the prototype. The model assures the appropriateness of the well-timed produced code and can be used in evolutionary and conventional prototyping [16].According to the author [17] Background Research, requirement Elicitation and Analysis, prototyping, Requirement verification, validation and requirement specification are the key components of Bee Hive Model. The strength and weakness [16] of the model include:

Strength(s):

- It does not combine both parallel and serial model prototyping
- Feasibility study phase carried out parallel with all other phases.
- It identifies and focuses on only the vital requirements.

Weaknesses:

- There is no concept of requirements prioritization
- There is no requirements pre-processing activity involved in the model.
- There is no concept of effort estimation
- Does not grant effective risk management policy.
- No criterion for application specific requirements elicitation technique

C. Kotonya and Sommerville Linear Requirements Engineering Process model.

The linear Requirement Engineering process model was suggested by Kotonya and Sommerville [32]. The model as depicted in Figure 1 overlaps and encompasses repetition among activities like requirement elicitation, requirement analysis and negotiation, requirements documentation and requirements validation [15].

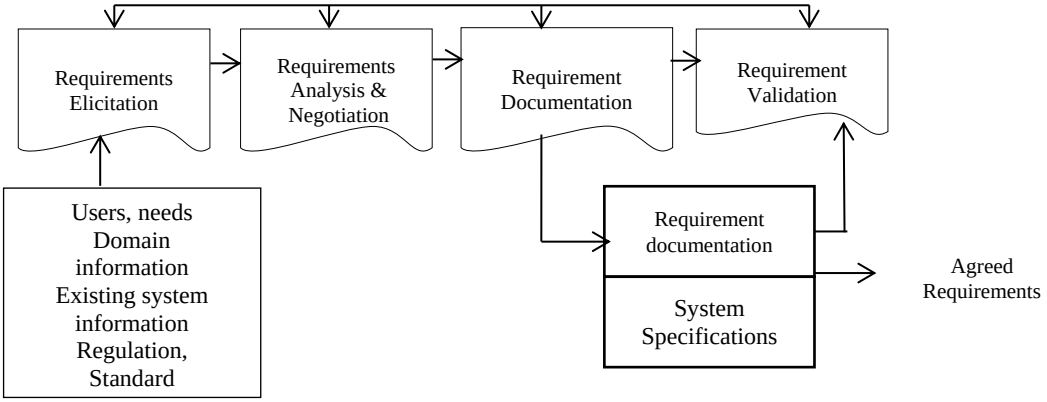

Figure 1: Kotonya and Sommerville Linear Requirements Engineering Process model

The strengths and weaknesses of this model as reviewed in [16] of the model includes:

Strengths:

- It can work well with small projects
- It is the basis of other Requirement process model.

Weaknesses:

- No support for user feedback
- No policy that performs risk management
- No concept of effort estimation on the basis of requirements.
- No requirements pre-processing activity.

- No criterion for using application specific requirement elicitation technique.

D. Macaulay Linear Requirements Engineering process Model

The linear requirement engineering process model as presented in Fig.2 was proposed by Linda Macaulay in 1996 [33]. There are five activities arranged sequentially in this model namely. Concept, Problem analysis, feasibility and choice of option, analysis and modelling and requirement documentation [18].

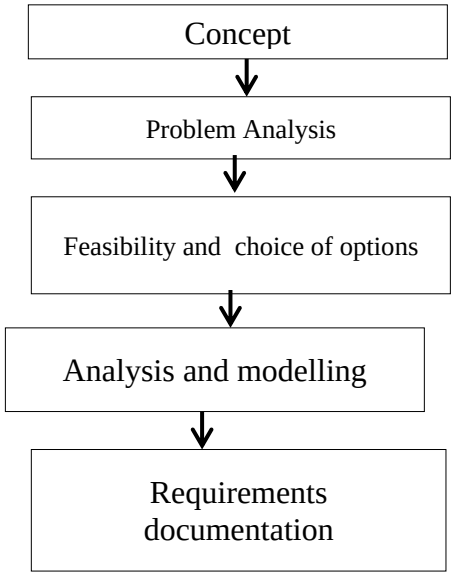

Fig.2: Macaulay Linear Requirements Engineering process Model

The strength and weaknesses of Macaulay Linear model according to [16] are:

Strength(s):

- it is good for small projects
- it provide support for analysing system's feasibility
- it is linear and does not overlap activities.

Weaknesses:

- No criteria for selecting application specific requirements elicitation technique.
- No support for changing requirements
- No concept of requirements pre-processing
- No support for risk management activity
- No support for user feedback
- No support for effort estimation.

E. Loucopoulos and Karakostas Iterative Requirements Engineering process model

This iterative and cycle requirements process model as presented in Figure 3 exhibits connections between phases of requirements engineering such as gathering (elicitation), requirements specification and validation to the problem domain [15]. This iterative model was proposed by Loucopoulos and Karakostas in 1995 [34]

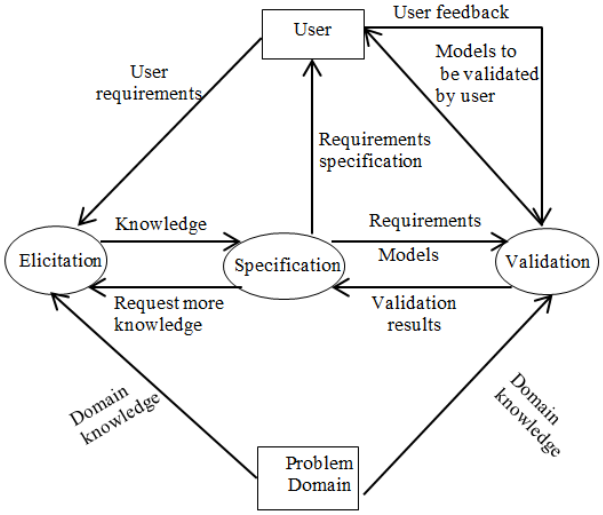

Fig 3: Loucopoulos and Kanakostas Iterative Requirements Engineering process model

The strength and weaknesses of this model as reviewed by [16] includes:

Strength(s):

- Provides support for user feedback
- Client basically validates the prototype of a system to be developed.

Weaknesses:

- No support for risk management in software development.
- No support for the concept of effort estimation
- There is no support for requirements pre-processing
- No criteria for using application specific requirements elicitation technique.

F. Tools Cost Benefit Analysis (TCBA) Requirement Engineering process Model

Tools Cost and Benefit Analysis (TCBA) Requirement Engineering Model suggested by shams-UI-Arif, Mr.Qadeem Khan, S.A.K Gahyyur [18], uses survey method for elicitation when the users are in excess and to use interview method if the users are limited to a particular department or office. According to [16], the model as presented in Figure 4 will figure out Return On Investment (ROI) prior to the start of the project i.e. computation of costs involving in staff payments, hard/software, maintenance, recreational, library, networking, employee pension and health facility.

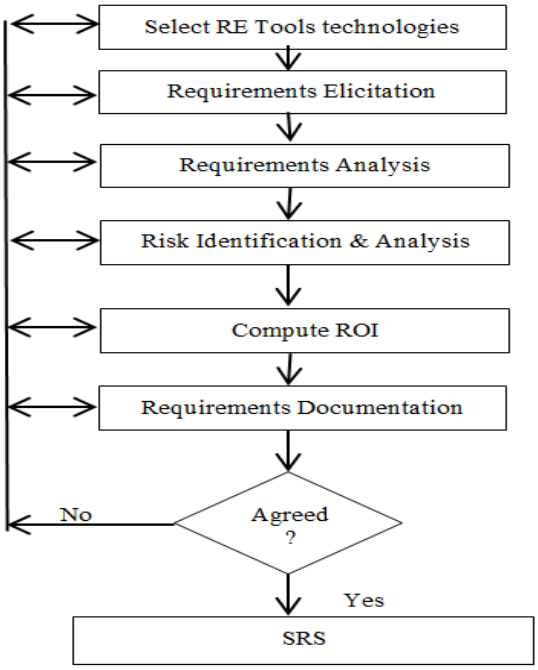

Fig. 4: Tools Cost Benefit Analysis (TCBA) Requirement Engineering process Model

The strength and weaknesses are as reviewed by [16] are:

Strength(s):

- It incorporates the feature of client feedback
- It grants an effective risk management
- The major strength of this model is computation of ROI.i.e resources and budget estimation is done prior to the project initiation.

Weaknesses:

- No support for requirement pre-processing
- No support for using application specific requirement elicitation technique
- No concept of requirements prioritization

G. An Effective Requirements Engineering process model

The model proposed by [20] relates requirements engineering process to software development process and introduces some significant but unseen viewpoints of some requirement engineering such as business requirements, customer requirements, user requirements, constraints, security requirements, information requirement and so on. The model as presented in Figure 5 also incorporates requirements management and planning phase to overcome issue of changing requirements [16,20].

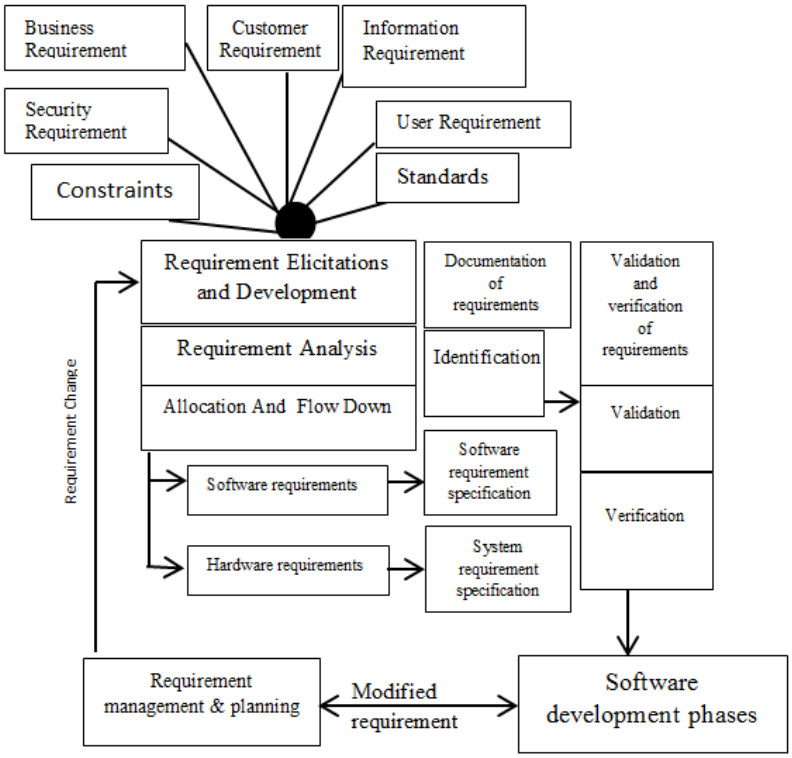

Fig. 5: An Effective Requirements Engineering process model

The strength(s) and weaknesses as pinpointed by [16] includes:

Strength(s):

- Support different viewpoints like business, customer, user, constraints, security and information requirements.
- It incorporates requirements management and planning phase for software developments.
- It support changing requirements.

Weaknesses:

- No concept of effort estimation
- No support for requirements pre-processing
- No effective risk management
- No criteria for application specific requirement elicitation technique.

III. Proposed Requirement Engineering process

Based on the existing literature, the strength and weaknesses of requirements engineering process models reviewed are described in section 2. It can be deduced that further work is required in the area of requirement pre-processing, risk management, requirement prioritization and application specific elicitation technique. It is against this drawback that Participatory Requirement Engineering Process model is proposed to overcome identified major weaknesses of existing models which are requirement pre-processing, risk management and application specific elicitation techniques. The proposed model (PREP) will raise the performance. Time frame, cost and quality of software development.

In the field of software engineering, the aim of requirement engineering is to discover clear, consistent, modifiable and traceable requirements that can be implemented to produce high quality software. The proposed model as presented in Fig 6 is design with this in mind.

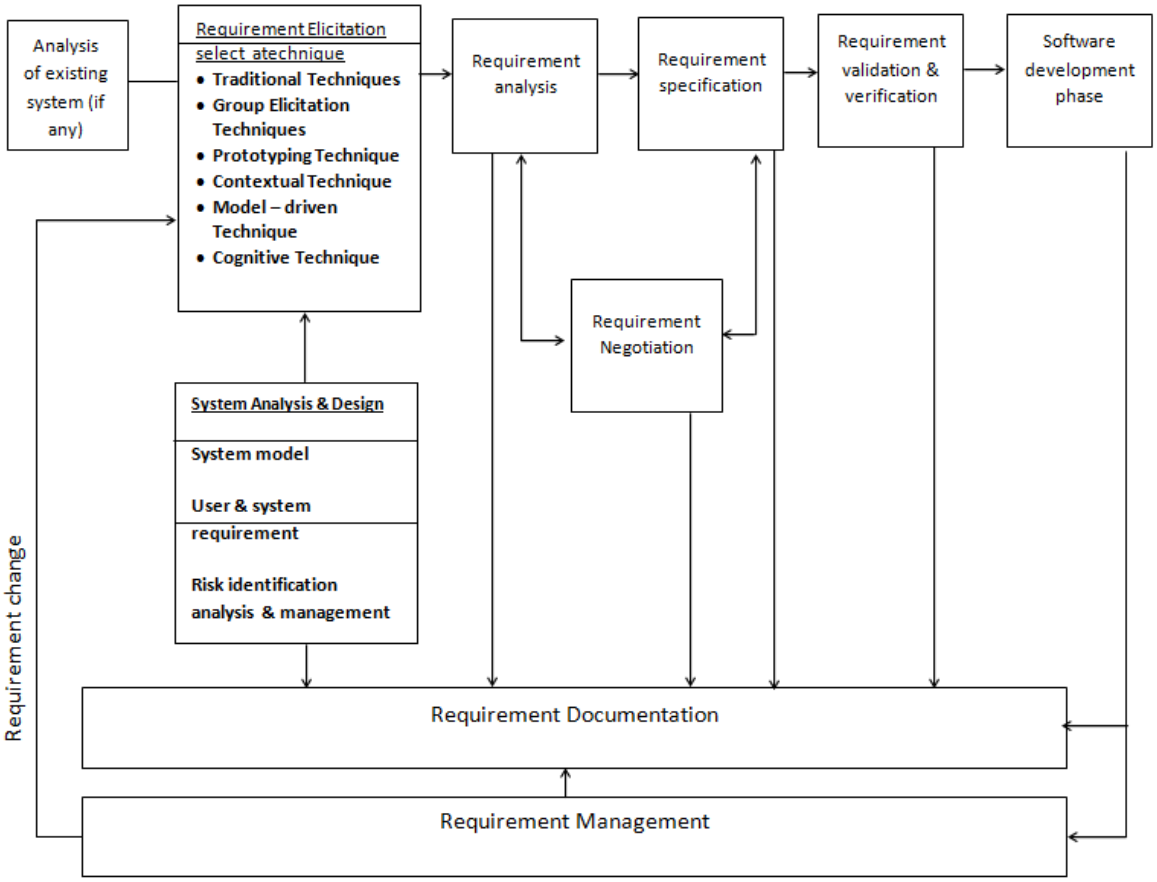

Fig 6: Participatory Requirement Engineering Process (PREP) Model

The model consist of the following phases: Analysis of existing system, Requirement elicitation, Requirement analysis, Requirement specification, Requirement validation and verification, software development phase, requirement identification, requirement negotiation, requirement documentation, requirement management, system models, Risk analysis and management, user and system requirement.

A. Requirement Elicitation

Requirement elicitation phase allow the analyst to select the application specific elicitation technique for the development process.

Elicitation technique can be grouped into six classes according to [21] as follows:

- Traditional techniques; these are long-established and well-known techniques such as document analysis, questionnaires, interviews, meetings.
- Group elicitation techniques; encourage stakeholders to communicate requirements more willingly. The most used techniques include focus groups and brainstorming sessions.
- Prototyping is one of the most used techniques to elicit hidden requirements and to draw some feedbacks. It is often used in combination with other techniques.
- Contextual techniques- these involves studying users in the context and environments to system implementation such as observation or conversation analysis.

- Model-driven techniques – includes a specific model and specific type of information to guide the elicitation process.
- Cognitive techniques- are used to attain knowledge by means of various techniques.

According to [18], the tool selection depends upon the type and complexity of the project to be developed. If the project is general purpose market project then survey is the best option and if the project is specific for the particular organization, then interview is the best option. Other tools can be selected according to their pros and cons applied on the project. [21,22] have investigated the elicitation technique selection and state that a particular elicitation technique may be selected for a variety of reasons:

- The technique selected is the only one the analyst knows.
- The technique selected is the analyst's favourite.
- The selected technique is the one prescribed by a specific methodology that is being followed for the system development and
- The choice of technique is governed solely by the intuition of the analyst to be effective in the current context.

B. Requirement Analysis phase

Elicited requirements are often incorrect, incomplete, inconsistent and ambiguous [26]. The requirements gathered at the elicitation stage is analysed at the analysis phase and triangulated to ensure that the requirement specification is free of inconsistency, ambiguity redundancy etc. the analysed requirements needs to be documented to enable communication with stakeholders and future maintenance of requirements and the system. Apart from conflicting the raw data, prioritization of software, software allocation and builds models are part of software requirement analysis.

C. Requirement Specification

Requirement specification is the final blueprint for the system realization, it consist of functional and non-functional requirement of the system. It includes sets of use cases that describe all the interactions that users will have with the system/software. It also fully describes what the software will do and how it will be expected to perform. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluation in the Requirement specification [19].

Requirements are documented in a formal artifact called Requirement specification (RS). Nevertheless, it will become official only after validation. A RS can contain both written and graphical (Models) information if necessary [26]

D. Requirement Verification and validation

The nature of the entire requirements described in the requirement specification needs to be agreed upon by different particles involved.

Requirement verification ensure that requirements are stated correctly, consistently, unambiguously, and comprehensively in the requirement specification document.

Requirement validation ascertains that the correct requirements are stated. It is at this stage that it checks and ensures that the documented requirements and models are consistent and meet the needs of the stakeholder. A formal technical review team should be used it is only if the final to carry out the validation process, the team will examine the system specification looking for errors in draft passes the validation process that the content or interpretation, areas where clarification is required, Requirement specification becomes official. [24]missing information, inconsistencies, conflicting requirements or unrealistic requirements.

E. Requirement Identification and Requirement management

Requirement Identification :The requirements identification practices focuses on the assignment of a unique identifier for each requirement [29] these identifiers are used to refer to requirements during product development and management.

According to [29], the requirement identification activities consist of three sub-activities which include: labelling, structure based identification and symbolic identification. The basic numbering activities include significant numbering and non-significant numbering.

Requirement management: Requirement management manages all the activities related to the requirements since inception, supervising as the system is developed until it is completed and put to use. Requirement management is a continuous activity that can be performed during and after development and during maintenance because requirements may continue to change [10,25]. During this phase, a set of activities to identify, control and track requirements and changes to the requirements at any time as the projects proceeds will be performed. Each requirements will be assigned a unique identifier. The requirements will then be placed into one or more traceability tables and these tables may be stored in a database that relate features, sources, dependencies, subsystems and interfaces to the system requirements.[10]

F. Requirement Negotiation and Requirement Documentation

Requirement Negotiation **:**Every conflict with the system should be reconciled through the process of negotiation. Stakeholders must rank their requirement are identified and analysed at this level.

Requirement Documentation :A document must be prepared during and after collecting requirements, which contains a complete description of the external and internal behaviour of the software system.

G. System Analysis & Design

According to [27,28], this technique has been in use since 1970s and is a functional oriented approach. In the proposed model, the following activities will be carried out under system analysis and design:

- Requirement identification
- System model
- User and system requirement

H. System Model and User/System Requirements

The raw data taken can be used to design the system model and the user/system requirements using the following approaches

a) Flow diagrams

Examples of this approach is Data Flow Diagram (DFD) which shows the flow of data among the proceeding elements. Functional decomposition is achieved using this technique also

Flowcharts (which is a type of diagram that represents an algorithms, workflow or process and illustrates a solution model to a given problem) can also be used

b) Entity Relationship Diagram (ERD)

ERD shows the complex information in the form of entities, attributes and their mutual relationships.

c) **Unified Modelling Language (UML)**

UML describes the user requirements by using several techniques such as use case diagrams Activity diagrams, deployment diagrams and so on. All of these define the functional behaviour of the system which defines the customer requirements [18]

IV CONCLUSION

In this paper, a participatory requirement engineering process (PREP) model for software development projects is proposed. The proposed model is more effective to product quality requirements. The major existing requirement engineering processes have been reviewed and they are limited to cover only certain aspect of requirement engineering like requirement elicitation, requirement analysis, requirement verification and validation and requirement specification [14], leaking out other important and hidden aspect. The proposed model introduces other aspects of requirement engineering into the software development process for producing high quality software products. The new phases are:

- Requirement pre processing activity – this is the first phase in our model where the existing system is analysed (if any) before the commencement of requirement elicitation and other aspect of requirement engineering.
- Criteria for selecting application specific requirement elicitation technique – in the proposed model, different elicitation techniques available is presented and the criteria for selecting the best technique for a particular project is also presented.
- The proposed model presents new insight of requirement preprocessing activity and criteria for selecting application specific requirement elicitation techniques.

The proposed model is iterative in nature which can be used in requirement prototyping with more stakeholders participation. The model can be used in larger as well as complex software development process, where requirement changes continuously.

REFERENCES

[1] *IndikaPerera: "Impact of Poor Requirement Engineering in software Outsourcing: A* study on Software Developers' Experience", International Journal of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. Vi(2011), No. 2 (June), pp. 337-348

[2] Wahono, R. S (2003): Analyzing Requirements Engineering Problems, IJW-2003, pp.55-58

[3] Asghar, S, Umar, M. (2010): Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-shelf (COTS) Components, IJSE, **1**(1)

[4] Fernandez D. M, Lochmann, K Penzenstadler, B; Wagner, S (2011): A case study in the application of an artefact –based requirements engineering approach. EASE'11, IET.

[5] Hofmann, H.F, Lehner, F (2001): Requirements Practice Addison- Wesley.

[6] Young , R. R. (2001): Effective Requirements Practices, Addison-Wesley.

[7] Philip A. Laplante, 2007. What Every Engineer should Know about Software Engineering.

[8] F.P. Brooks, "Mythical Man-Month: Essays on Software Engineering", 20th anniversary edition. Addison-wesley Professional, 1995. 5791CSEA 2011 : The Sixth International Conference on Software Engineering Advances.

[9] Saiediana H, Dalab R Requirements engineering : making the connection between the software, developer and customer(2000).

[10] Stevens, R., Brook, P., Jackson, K. & Arnold, S. Systems Engineering : Coping with Complexity. Prentice Hall Europe. (1998).

[11] Jantunen, S., "Exploring software engineering practices in small and medium –sized organizations", Proceedings of the 2010 ICSE Workshop on Cooperative and Human aspects of Software engineering (CHASE'10), pp. 99 -101,2010.

[12] Espiti 1996. Software process improvement on the right road with ESPITI-The ESPITI European Survey Results.ESPITI Newsletter Issue2. Available at: http://www.cse.duc.ie/cse/international/trispin/New2.html

[13] LachanaRamingworg et al. (2012): Review of Requirements Engineering Process, Problems and Models International Journal of Engineering science and Technology (IJEST). Vol 4. No. 06 pp 2999-3002

[14] Kotanya, Gerald; *Sommerville, Ian* (September 1998). Requirements Engineering: Process and Techniques. *John Wiley & Sons.*ISBN 0-471-97208-8.UK

[15] Sacha Martin, AybukeAurum, Ross Jeffery and Barbara Paech: "Requirements Engineering Process Models in Practice" (2002): The Seventh Australian workshop on Requirements Engineering, AWRE 2002 : The seventh Australian workshop on Requirement Engineering : proceedings, Melbourne, Victoria, 2-3 December 2002, Deakin university, School of Information Systems, Deakin University .

[16] Mona Baha, ArchanaBhantnagan, A comparatives study of requirements engineering process model international journed of advanced research in computer science vol 8, No. 3, march – april 2017 ISSN NO 0976-5697 PP 740-745

[17] Swarnalatha K S, G.N Srinivasan, Pooja S Bhandary, "A constructive And Dynamic Frame work For Requirement Engineering Process Model(Bee Hive Model)" International Journal Of Computer Engineering And Technology (Ijcet), Volume 5, Issue 7, July(2014), pp. 48-54

[18] Shams-Ul-Arif, Mr.Qadeem Khan, S. A. K. Gahyyur, "Requirement Engineering processes, Tools/Technologies, & Methodologies" International Journal of Reviews in computing, 2009-2010. http://www.ijric.org/volumes/vol2/6vol2.pdf

[19] DhirendraPandey& U. Suman, "An Effective Requirement Engineering Process Model for Software Development and Requirements Management", International Conference on Advances in Recent Technologies in Communication and Computing, IEEE society,2010. http://www.tamps.cinvestav.mx/~ertello/civis/s01-ReqEngmanage.pdf

[20] Nuseibeh, B; Easterbrook, B. (2000): Requirements engineering: a roadmap. ICSE '00, pp. 35-46, ACM, New York, NY, USA.

[21] Hickey AM, Davis AM ,Elication techniques selection: *Howdo experts do it?* In: Proceedings of the 11th IEEE International requirements engineering conference,, Monterey Bay, CA, (2003)

[22] Herela D, Greenberg S, Using a groupware space for distributed requirements engineering. In: Proceedings of the 7th workshop on enabling technologies: Infrastructure for collaborative enterprises, June 17-19, Stanford, CA, (1998)

[23] Pandey, U. Suman, A. K. Ramani, "Social Organization Participation difficulties in Requirement Engineering Process- A Study", National Conference on Emerging Trends in Software Engineering and Information Technology, Gwalior Engineering College, Gwalior, 2009.

[24] Sommerville, Ian (2009). Software Engineering (9th Ed.). Addison Wesley. ISBN987-0-13-703515-1.

[25] S. Blanchard & W. J. Fabrycky, Systems Engineering and Analysis. Prentice-Hall , 1981.

[26] Palmer, J.D; Liang ,Y, Wang, L. (1990): Classification as an approach to requirements Analysis, "Advances in classification research and application: proc. 1st ASIS SIG/CR Classification research Workshop, S.M Humphrey and B.H Kwasnik, eds, Learned Information, Medford, N.J.

[27] DeMarco T, Plauger PJ Structured analysis and system specification. Prentice Hall, New York, NY (1979)

[28] Yourdon E., Modern structured analysis Prentice Hall, Englewood Cliffs, NJ, (1989)

[29] J. Siddiqi, "Requirement Engineering: The Emerging Wisdom", IEEE Software, 1996, pp. 15-19.

[30] S. Lanesen, Software Requirements: Styles and Techniques, Addison Wesley, 2002.

[31] Pandey, U. Suman, A.K. Ramani, "Design and Development of Requirements Specification Document for making Quality Software Products" National Conference on ICIS, D.P. Vipra College, Bilaspur, 2009.

[32] Kotonya, G. and Sommerville, I. (1998): Requirements Engineering-Processes and Techniques, John Wiley and Sons, UK

[33] Macaulay, L. A. (1996): Requirement Engineering, Springer-verlag

[34] Loucopoulos, P. and Karakostas V. (1995): System Requirements Engineering, McGraw-Hill Book Company, Europe.