VOL. 3, NO. 5, May 2012 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

The Role of Requirement Engineering in Software Development Life Cycle

[1] Abhijit Chakraborty, [2] Mrinal Kanti Baowaly, [3] Ashraful Arefin, [4] Ali Newaz Bahar

[1, 2] Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Bangladesh.
[3] Departments of Electrical and Electronic Engineering, Northern University Bangladesh, Bangladesh.
[4] Institute of Information Technology, University of Dhaka, Bangladesh.
[1] abhijit.cse@gmail.com, [2] baowaly@gmail.com, [3] arefin.mist@gmail.com, [4] bahar_mitdu@yahoo.com.

ABSTRACT

The Requirement Engineering (RE) is the most important phase of the software development life cycle (SDLC). This phase is used to translate the imprecise, incomplete needs and wishes of the potential users of software into complete, precise and formal specifications. The specifications act as the contract between the software users and the developers. Therefore the importance of Requirement Engineering is enormous to develop effective software and in reducing software errors at the early stage of the development of software. Since Requirement Engineering (RE) has great role in different stages of the SDLC, its consideration in software development is crucial. There exist a number of approaches for requirement engineering. Among the approaches, object-based and problem domain-based approaches are widely used. An effective analysis of methods is essential for the appropriate capturing of requirements. Taking the above viewpoint into account, this paper demonstrates an effective method of requirement engineering, which plays an important role in different phases of the SDLC. Hospital can be seen as an example of a complex system. Therefore, the paper considers Hospital as a case study for which a software system has been developed taking the mentioned approach into account.

Keywords: *Object-based Method, Problem Domain, Requirement Engineering, Software Development Life Cycle.*

1. INTRODUCTION

Requirements engineering (RE) is the most important area of Software engineering and possibly of the entire software life cycle. Because errors produced at this stage, if undetected until a later stage of software development, can be very costly. If it can specify the requirement specification correctly, errors of later stages will be less. Hence software developments and maintenance cost will be less moreover customer will get their desire system. Object model which is part of requirement Analysis, functional model and dynamic model shows the requirement specifications in the requirement engineering stage. These three kinds of models describe a system. Object model describes the static structure of the object in a system and their relationship; the dynamic model describing the interaction among objects in the system; and the functional model which describes the data transformation of system.

It is seen that this method captures requirement specifications by building models from the problem statement. Hence it is concentrated on building models rather than analysis. The domain-based approach concentrated on the understanding of the system problem domain and defining the boundary of system. Hence, it provides more precise way of transferring of information from the user-domain to analyst domain. Thus, an appropriate capturing of information from the user-domain can be ensured.

This information through a series of processes can be transformed into more concrete software requirement specifications through the analysis of its environmental, behavioral and solution characteristics. In **[** HYPERLINK \l "IBM" **1] 2] [** HYPERLINK \l "YWa02" **3]** describe a robust requirement engineering approach which has the positive effect on Software Development Life Cycle (SDLC) phases including requirement specification, system specification, system design, programming and coding, testing and maintenance. Among the phases software requirement specifications can be considered as the most important because the reliability, efficiency, usability and maintainability of a software system mainly depends on this phase. This is especially true when the CBIS for a large complex system like hospital to be developed. Inappropriate capturing of requirement specification will cost a large amount of money in the operational environment of a software system. Therefore, this paper presents the capturing procedures of requirement specification for software to be used to support the activities of a hospital which has the positive effect on requirement engineering phases of SDLC.

2. SYSTEM DEVELOPMENT METHODOLOGY

The Figure 1 shows the broad stages of the SDLC. As can be seen, it is a closed loop of 8 (eight) steps, initiated or triggered by one of two inputs. The concept of a life cycle for software was first proposed (IBM, 1996) in **1].** The aim of the software lifecycle is to provide an overall framework in which the development and lifetime of a software system

can be described. The software lifecycle does not tell the developer how to produce the various outputs of the different stages. Their objectives and the outputs each stage should deliver.

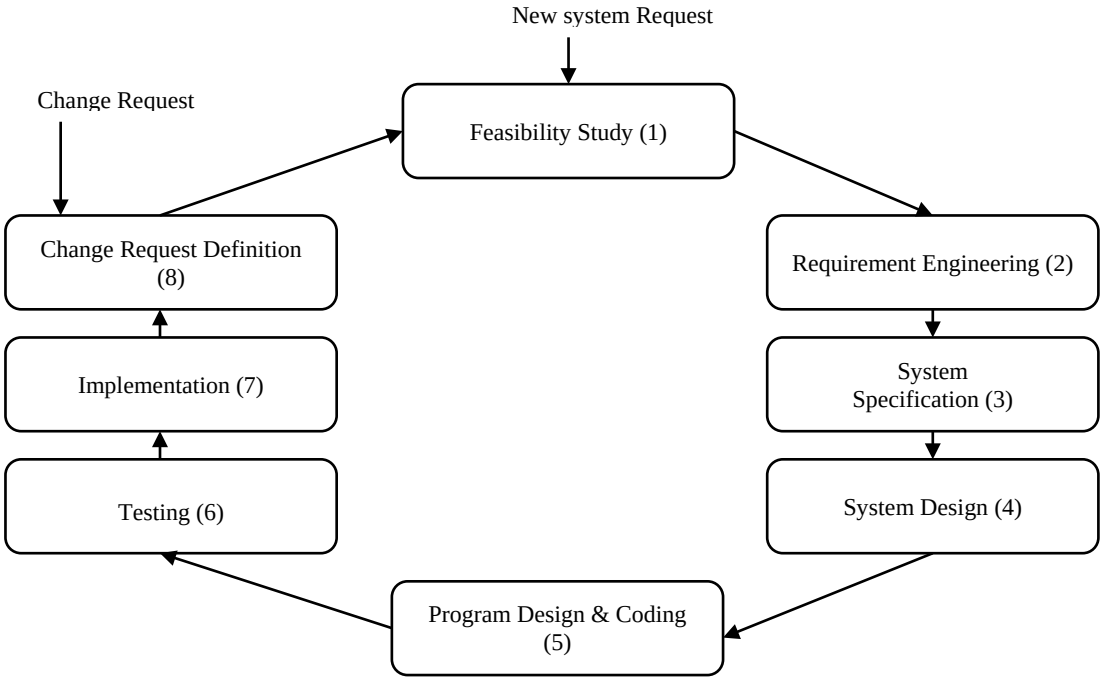

Figure 1: Royce's Software Development Life Cycle.

3. RELATED WORKS

There are many methodologies to construct these steps but there is no fixed methodology for each step. It may vary in respect of the problem domain. In **[**HYPERLINK \l "Bru03" **4] 5] [** HYPERLINK \l "Wan96" **6]** OOSDM is widely used approach takes an object oriented viewpoint. Here the problem domain is characterized as a set of objects that have specific attributes and behaviors and that are categorized into classes and subclasses. A formal definition of object oriented is: Object-Oriented = Objects + Classification + Inheritance + Communication. Function-oriented (or procedural) design decomposes the design into a set of interaction functions, which act on a centralized state. This approach concentrates on the processing and algorithms of the system. Function-oriented design has been practiced since programming began, and there is a large body of design methods based on functional decomposition. However, while the functions hide the details of the algorithms, the shared state can be a particular problem as a function could change the state in ways not anticipated by other functions. In **7]** User-centered design has numerous benefits for business. Firstly, UCD methods result in higher-quality screen-based systems with increased customer satisfaction and confidence. In **[** HYPERLINK \l "Sur04" **8] 9]** the competitive online market, a high-quality design can mean the difference between success and failure, Secondly, under UCD methodologies, software development is more efficient. The system will likely go to market faster and cost less.

4. REQUIREMENT ENGINEERING FRAMEWORK

Requirement Engineering (RE) is the first major activity which involves with understanding problem domain (described in a statement of needs), solution determination, and specification of a solution that is testable, understandable, maintainable, and that satisfies project quantity guidelines **[** HYPERLINK \l "ISO89" **10]** (ISO:9001, 1989). Figure 2 produces feasibility report and goes ahead to the output as definition of requirements and specification of requirement through the process of Requirement Analysis (RA), Requirement Definition (RD) and Requirement Specification (RS). The feasibility study process gives feasibility study report. After getting requirement acquisition which is the input of the statement of requirements, requirement modeling is performed and last

of all go to the requirement model process and finally the requirement is validated.

5. ANALYSIS OF VORD METHOD FOR THIS CASE STUDY

There are many methods for Requirement Engineering (RE) such as: VORD Method, OMT, and Domain Based Approach. VORD (Figure 3) method is introduced by Kotonya and Sommerville in **1]**. Its principle objective is requirement detection and analysis that helps translating this analysis into Object Oriented system model. Figure 4 illustrates the domain components that are related to the medical services provided to the patients.

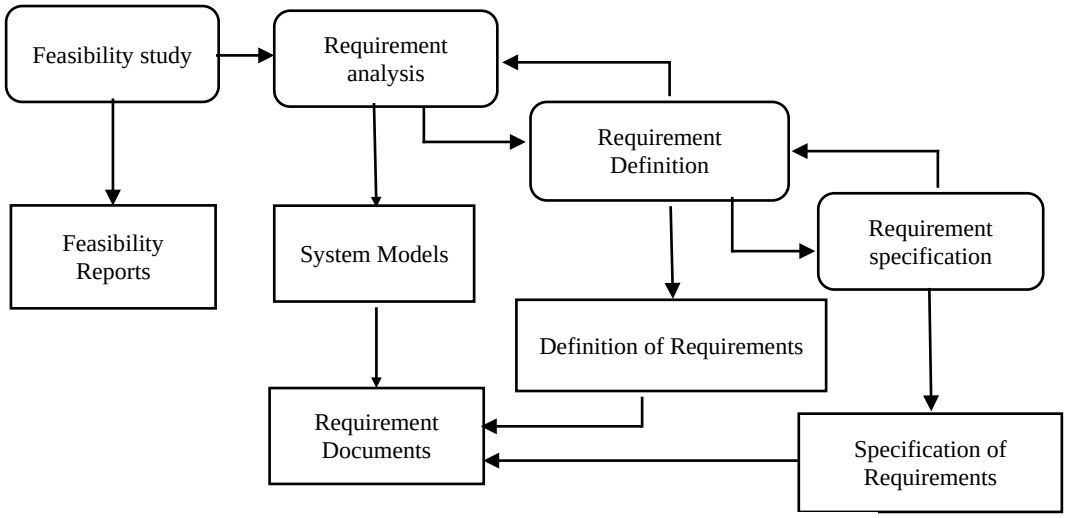

Figure 2: A Framework for Requirement Engineering Process

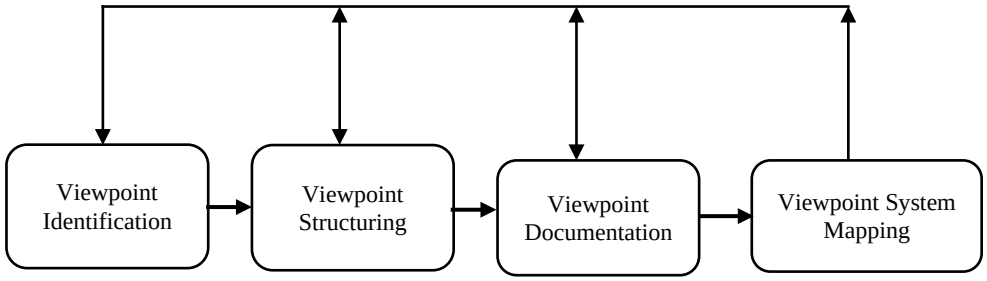

Figure 3: VORD Method of Requirement Analysis

VOL. 3, NO. 5, May 2012 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

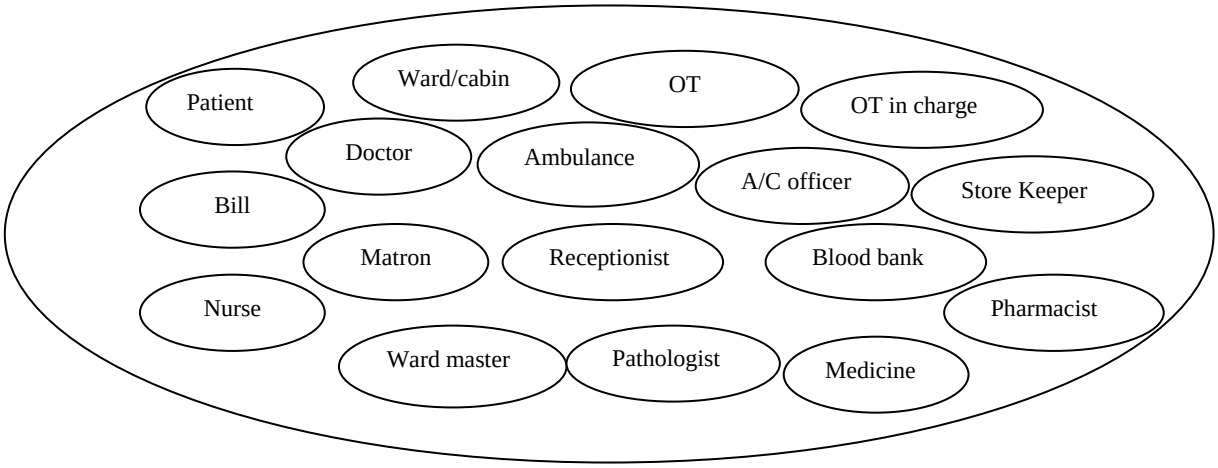

Figure 4: The Domain Components of a Hospital

6. PROBLEM SCOPE OF THIS CASE STUDY

Figure 5 illustrates the problem area of Hospital Automation. The input of the record keeping process of this manual system is patients' profile, roistering information, staff profile, doctor profile, booking details, inventory information, selling information, distributing information, scheduling information. Based on problem scope the above Figure 6 illustrates the viewpoints and their services where the white ovals are viewpoints and the shadow ovals are services. Figure 7 shows an event trace for a patient registration scenario. If more than one object of the same class participates in the scenario assign a separate column of each object by scanning a particular in the trace, we can see the events that directly affect a particular object. Only these events can appear in the state diagram for the object. The Figure 8 is the demonstration of the structure of the viewpoints that describes the services according to the viewpoints. Here the services may pick up successfully that helps the developers or analyst to identify the events, functions and processes with objects. The Figure 9 demonstrates scenarios of an event, patient registration, that realize the total operation with inputs, processors and outputs.

During the first stage/Analysis phase, VORD method does not explicitly identified the objects. But at fourth stage, system mapping it transfers into viewpoint documentation into object model for requirement specification.

.

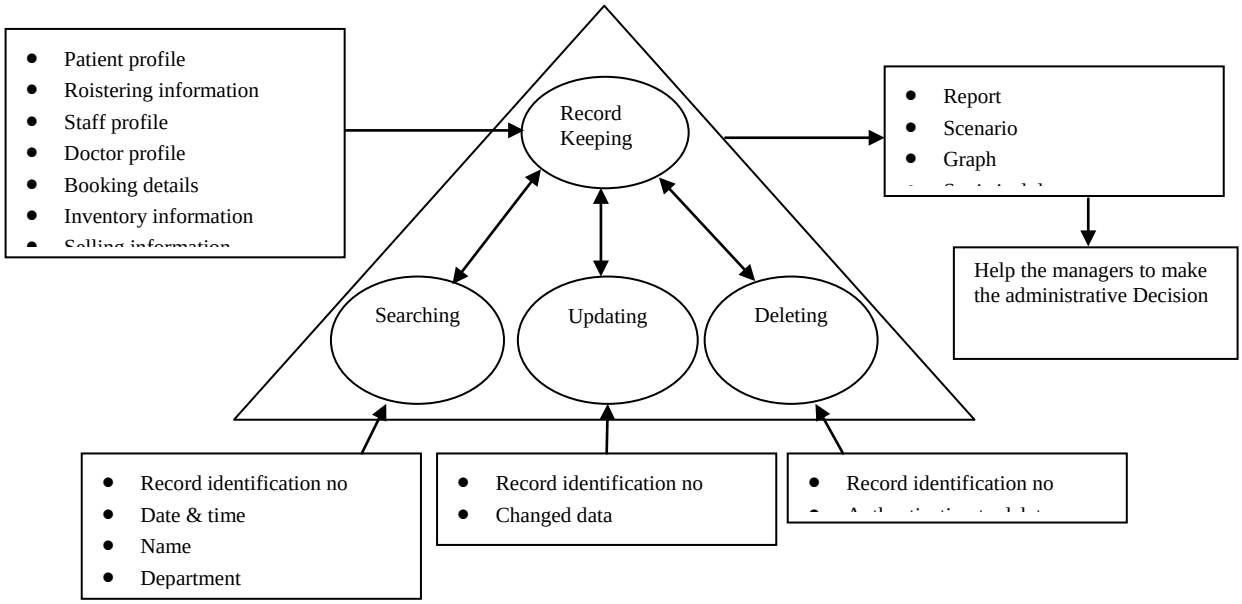

Figure 5: Problem Scope of a Hospital

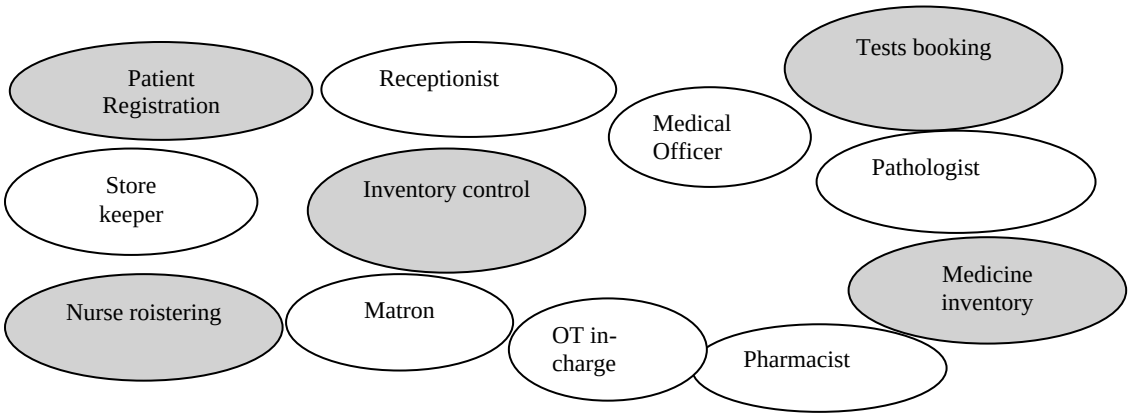

Figure 6: Brainstorming for Viewpoint Identification on Hospital

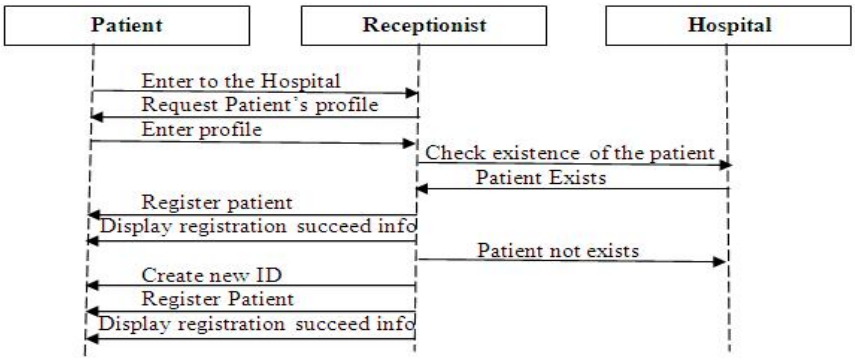

Figure 7: Event Trace Diagram of Reception Desk of Hospital

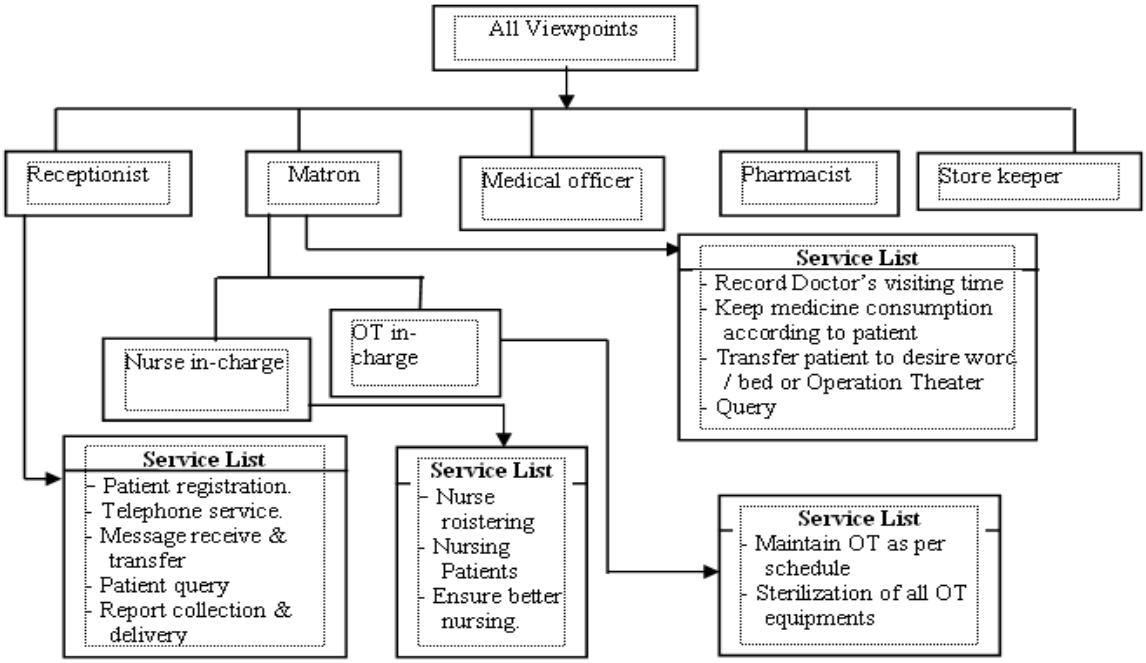

Figure 8: Structuring Viewpoints in the System

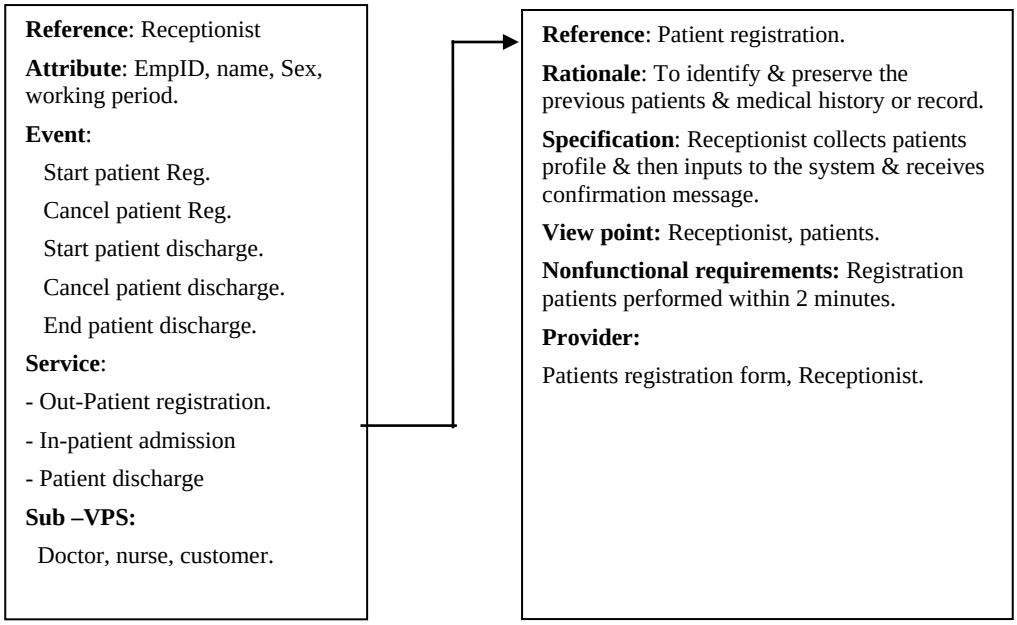

Figure 9: Viewpoint Template and Viewpoint Service Template

7. DISCUSSION & CONCLUSION

As Requirements engineering, second step of SDLC, which involves with requirement analysis is performed by requirement engineering (RE). As it is discussed earlier, requirement analysis is the most significant part of the SDLC, it is reviewed that the three Requirement Engineering (RE) frameworks to measure which framework may be the better approach for RE process. From the observation, it was seen that other two frameworks have some limitations which was tried to cover up in Somerville framework; it considers two important inputs user and problem domain. The user knows better his need and wishes of desire system and problem domain where system will be applied. Moreover, it has some extraordinary scope such as its cyclic process, iterative and nature of real world approach etc. It is crucial decision to choose a suitable requirement engineering method. As other methods such as abstraction and portioning are concentrated with creating models which are helpful for developers but are not concentrated on customer need and wishes. In this point of view, domain based approach is appropriate for capturing users' need and wishes and its boundary that reduce error on next steps, consequently maintenance cost will be low. Further work needs to be carried out in order to adjust and refine existing measures that satisfy the most critical phases in SDLC, especially in the architecture and program design phases.

REFERENCES

[1] IBM, "Software Development Performance and Practices in Europe: A Benchmark of Software Development in Europe - Self Assessment Questionnaire," IBM Euro-coordination, vol. 2, pp. 1-11, 1996.

[2] G. Kotonya, "Practical Experience with Viewpoint-Oriented Requirements Specification," Journal of Requirement engineering, Springer London, vol. 4, no. 3, pp. 115-133, September 1999.

[3] Yingxu Wang and Antony Bryant, "Process-Based Software Engineering: Building the Infrastructures," Annals of Software Engineering, vol. 14, no. 1-4, pp. 9-37, December 2002.

[4] B. Bruegge and A.H. Dutoit, Object-Oriented Software Engineering Using UML(Conquering complex & changing systems), 2nd ed.: Prentice Hall, Englewood Cliffs, NJ, 2003.

[5] Elli Georgiadou, "Software process and product improvement: a historical perspective," Cybernetics and systems analysis, vol. 39 (1), pp. 125-142, 2003.

[6] Court I., Ross M. and Staples G. Wang Y., "Towards a Software Process Reference Model (SPRM)," in Proceedings of International Conference on Software Process Improvement (SPI'96), Brighton, UK, November, 1996, pp. 145-166.

[7] J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. Rumbaugh, Object-Oriented Modeling and Design.: Englewoor Cliffs, NJ: Prentice-Hall, 1991.

[8] Abran A., and Laporte C. Y. Suryn W., "An integrated life cycle quality model for general public market software products," in Proceedings of 12th International Software Quality Management & INSPIRE Conference (BSI) Canterbury, Kent, UK, 2004.

[9] Court I., Ross M. Staples G. King G. and Dorling A. Wang Y., "Quantitative Analysis of Compatibility and Correlation of the Current SPA Models," in Proceedings of the IEEE International Symposium on Software Engineering Standards (IEEE ISESS'97), 1997.

[10] ISO: 9001, Quality Systems - Model for Quality Assurance in Design, Development, Production, Installation, and Servicing. Geneva: International Organization for Standardization, 1989.