Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 4, Issue 1; January-March, 2017, pp. 16-19

http://www.krishisanskriti.org/Publication.html

Re-Defined Requirement Elicitation

Swati Goel

Department of Information Technology Bhararti Vidyapeeth College of Engineering Pune-411043, India
E-mail: swatigoel96@gmail.com

Abstract—*In Today's world Software Companies are facing many problems while gathering requirements from the users, customers or stakeholders. Hence, resulting in controversial delivery of software product to the client side which ultimately affects the reputation of software companies and leads the customers not fully satisfied with the final software product. Requirements elicitation is considered one of the most crucial part of project's lifecycle. The final software developed at the end is directly proportional to how correctly the requirements have been gathered in requirement elicitation phase. Hence, to simply the task of requirement gathering this paper defines a new method of elicitation i.e. Stack-In method which states that whether the requirement can be practically implemented or not in the first stage of SDLC itself .Hence, the final product delivered will be 90% correct.*

Keywords*: SDLC, Stack-In*

1. INTRODUCTION

The success or failure of a software is highly dependent on the quality of requirements gathered in the first stage of SDLC i.e. Requirement Elicitation stage. The quality of requirements is dependent on the methodology that is used for gathering these requirements[1].Correctly gathering of these requirements significantly improves the quality of requirements and in turn the quality of end software product delivered. In recent years the Requirement Engineering community has dedicated a lot of efforts in order to handle the Reuirements Elicitation(RE) problems. Although many important results have been discovered, the RE processes still present challenges that has to be resolved in upcoming years[2].

There are many problems associated with requirements engineering like Problems of scope which means The boundary of the system is ill-defined or the customers/users specify unnecessary technical detail that may confuse, rather than clarify, overall system objectives.

Problems of understanding is the second problem which means The customers/users are not completely sure of what is needed, have a poor understanding of the capabilities and limitations of their computing environment, don't have a full understanding of the problem domain, have trouble communicating needs to the system engineer, omit information that is believed to be "obvious," specify requirements that conflict with the needs of other customers/users, or specify requirements that are ambiguous or untestable. Problems of volatility is the last problem which means the requirements change over time. The rate of change is sometimes referred to as the level of requirement volatility. Problems of volatility is considered as one of the most critical problem that the engineers face while gathering requirements from the users or customers.

Requirements elicitation is also considered as the most critical and error prone phase in any software system development cycle[3].Hence, extra attention should be taken while gathering requirements. Requirements elicitation is an repeated process which is greatly affected by the communication and understanding skills of requirement engineers and the commitment and cooperation of the system stakeholders. This implies that the communication barriers and agreement about the requirements are major issues involved with requirement elicitation[3].

The Requirements are gathered by studying the existing or obsolete system and software, conducting interviews of users and developers, referring to the database or collecting answers from the questionnaires .They are all traditional practices that are used by requirement engineers to capture the requirements from the users. For example, suppose that we are interested in eliciting new requirements for smart traffic manager system to be installed in urban cities which is capable of automated warning systems, Real-time traffic monitoring, Active traffic management, traffic camera monitoring and many other functionalities based on the condition of the traffic in the area. Use of traditional requirements elicitation approaches in these circumstances will be difficult, time consuming and sometimes even impossible.

Requirements can be gathered by oraganising one-to-interviews, group interviews, facilitated sessions, joint application development(JAD), Questionaires, Prototyping, Use Cases, Following people around, Request for proposal, Brainstorming etc. All these techniques involves active participation of user to deliver the quality end product to the customer.

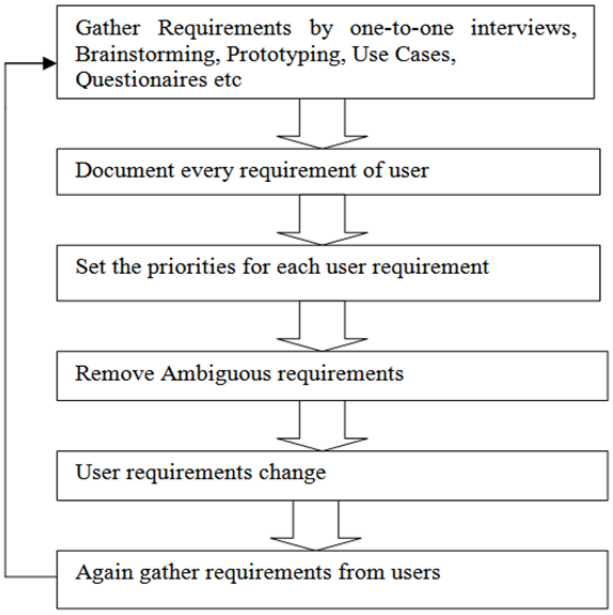

Fig. 1: Traditional Requirement Elicitation Process

Fig. 1 clearly shows that traditional method of requirement elicitation is time consuming. Hence**,** This paper proposed a novel method for requirement elicitation i.e. Stack-In method which will solve almost all the above stated problems.

2. PREVIOUS WORK

There have been many techniques proposed in the recent years for requirement elicitation. One of such technique was based on web server log analysis. In this technique after collecting the sample data from logs, a simple analysis is performed over the source data .The metrics of interest for this analysis are the number of unique users, page hits, average and maximum time taken, percentile of measured time taken[4].This technique does not only gather requirements but also establishes a priority ranking for them. The log based approach took considerably less effort than traditional method of requirement elicitation (Fig. 1).

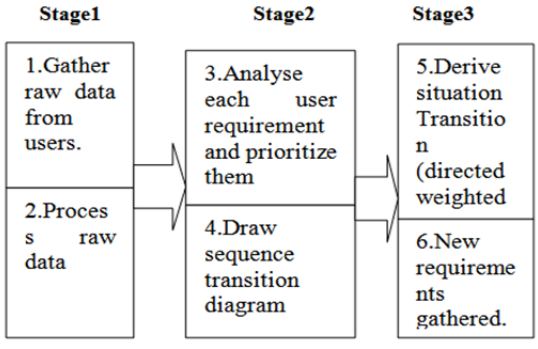

Fig. 2: Requirement Elicitation Methodology

The other technique was based on human –centered requirements elicitation methodology that considers end-user's desire, behavior and environmental context and represented in the form of 3-tuple<d,a,e>[3]where d denotes user expectation, a denotes transition vector and e denotes environment context vector. This methodology of requirement elicitation is divided into three stages(Stage1,Stage2and Stage3).

In this methodology requirements elicitation was based on transition diagram drawn using raw data that targets particular end-user group in a specific domain. Transition structure gives the overall view of a system domain that has to be operated.

3. PROPOSED WORK

In this paper a new technique is used for requirement elicitation which will make the task of requirements gathering easy for requirements engineers . The overall process of the proposed method shown in Fig.3. can be divided into three main phases where each phase consists of several steps. During the first stage, (1) All the users requirements is collected as stated by user with no modification, and (2) In this step repeated requirements has to be removed. In the second phase,(3)dependent requirements has to be placed simultaneously, (4) proposed Stack-In algorithm has to be applied. In the third phase (5) Again start with step 1 in phase1.

1) Capture all the end-user requirements

In this step all the user requirements are written in natural language supplemented by appropriate diagrams and tables in the requirements document[5].Ideally, the user and system requirements should be clear, easily understandable, complete and consistent. In practice, this is tedious task for stakeholders to achieve as they analyse the requirements in different dimensions and they are often inherent conflicts and inconsistencies in the requirements.

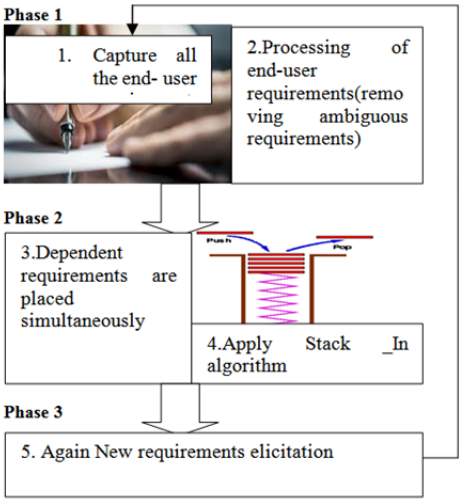

Fig. 3: Proposed requirement elicitation methodology

2) Processing of end-user requirements (removing ambiguous requirements)

In this step all the end-user requirements that have been written by the requirement engineers in step 1 has to be analysed and processed properly in order to remove ambiguous requirements (repeated). Customers/users are not technically sound hence thee specify unnecessary requirements that may confuse the requirement engineers rather than clarify, overall system objective. Hence, In this step all these such requirements has to be removed in order to developed reliable end software product.

3) Dependent requirements are placed simultaneously

In this step dependent requirements means the requirements which are related to one another has to be placed simultaneously because it will be easy for the requirements engineers to apply proposed Stack-In in next step it will be time saving.

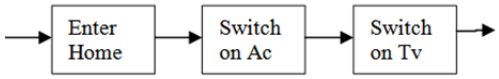

Fig. 4: Dependent Requirements in Smart Home

Fig.4. is the example of dependent requirements in smart home. As the user enters home he/she can switch on ac, Tv, Lights, Music system etc. Hence, all these dependent requirements should be placed simultaneously in one module.

4) Apply Stack-In Algorithm

After step 3 Stack-In algorithm has to be applied by the requirements engineers on concised set of requirements. In this algorithm user requirement is placed on the top of the stack and various sub requirements has to be placed on further levels of the stack. And, PUSH and POP operations has to applied on each user requirement placed in stack.

5) Again new requirements elicitation

In this step again start with step 1 in order to find to some missed out requirements.

4. ALGORITHM

In this paper Stack-In algorithm is used to implement the proposed technique. By applying this algorithm requirements engineers can easy tell the users/customers out of all the requirements how many them can be practically implemented.

4.1 Stack-In Algorithm

a. Place the user requirement on TOS.
b. Place various subways or sub requirements on further levels of the stack through which user requirement can be achieved.(Apply PUSH operation)
c. Check whether each sub requirement can be possible to implement practically or not.
d. If yes, this means user req. in step a. can be possible to implement.
e. If no, any one of sub req. in step c. fails that means user req. in step a. can't be implemented practically.

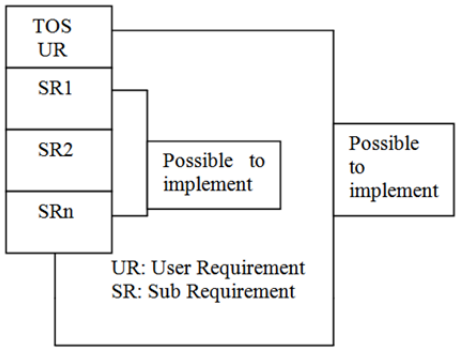

Fig. 5: Stack-In Algorithm (1)

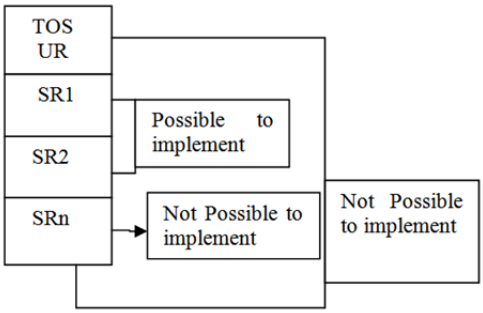

Fig. 6: Stack-In Algorithm (2)

5. RESULT

I observed that different techniques and approaches have different and relative strengths and weakness, and may be more or less suited to particular types of situations and environments. Likewise some techniques and approaches are more appropriate for specific elicitation activities and the types of information that needs to be acquired during those activities[6]. Table 1 below presents which of the selected core group of techniques and approaches are best suited.

Table 1: Comparison of various techniques

	Inter-views	**Team work**	**Prototyping**	**Goals**	**Proposed Algorithm (Stack-In)**
Understanding the domain	Good	Good	Average	Good	Excellent
Identifying sources of requirements	Good	Good	Average	Good	Excellent
Analysing the stakeholders	Good	Good	Good	Good	Excellent
Eliciting the requirements	Good	Good	Good	Good	Excellent

Table 1 shows that the proposed algorithm i.e. Stack –In algorithm give best results in all cases.

6. CONCLUSION

The stack-In has following advantages:

a. Accurate SRS will be developed.
b. No controversies between clients and developers.
c. 80% of problems will get eliminated.
d. Eliminates the need of acceptance testing.
e. Reliable software will be developed at the end.

With this proposed technique in this paper I conclude the process of requirement elicitation will definitely get much easier and it will be possible for requirements engineers to tell the customers whether the requirements can be practically possible to implement or not in first stage of software development itself. Hence, the proposed methodology is very time saving also.

REFRENCES

[1] Neetu Kumari.S, Anitha S. Pillai, "A Survey On Global Requirements Elicitation Issues And Proposed Research Framework" 2013 IEEE

[2] Aldrin Jaramillo Franco, "Requirements Elicitation Approaches : A systematic Review" 2015 IEEE

[3] Nimanthi L. Atukorala, Carl K. Chang, Katsunori Oyama "Situation-Oriented Requirements Elicitation" 2016 IEEE 40th Annual Computer Software And Applications Conference

[4] Odorico MaChado Mendizabal, Martin Spier, Rodrigo Saad "Log-Based Approach for Performance Requirements Elicitation and Prioritization" 2012 IEEE

[5] Ian Sommerville, "Software Engineering" Dorling Kindersley(India) publishers(2011)

[6] Didar Zowghi, Chad Coulin, " Requirements Elicitation : A survey of techniques, Approaches, and tools"