International Journal of Engineering & Technology, 7 (2.29)(2018) 772-775

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Requirements Elicitation in Software Engineering

Muhammad Ali Ramdhani[1*], Dian Sa'adillah Maylawati[2], Abdusy Syakur Amin[3], Hilmi Aulawi[4]

[1]Department Of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia
[2]Department Of Informatics, Sekolah Tinggi Teknologi Garut, Indonesia
[3]Department Of Industrial Engineering, Universitas Pasundan, Indonesia
[4]Department Of Industrial Engineering, Sekolah Tinggi Teknologi Garut, Indonesia
**Corresponding Author E-Mail: M_Ali_Ramdhani@Uinsgd.Ac.Id*

Abstract

Software Engineering (SE) is a discipline, concept, method that focus on all aspects related to software production. SE must adapt organized system and approach in building software. The objective of this article is to discuss needs elicitation in SE, so that the design can be done through utilizing appropriate tools and techniques based on the problem to be solved, restrictions and need that must be met, as well as resource availability. The method used in this article is literature review as a major source of problem analysis. In the next phase, the main topic of the article is clarified, validated, and verified by the model of the Focus Group Discussion (FGD). The respondents of FGD are lecturers of Informatics Department at UIN Sunan Gunung Djati Bandung. Requirements elicitation is in-dept and comprehensive process of finding information from all stakeholders in relation to the built software. The result of requirements elicitation is the explanation from every stakeholder with natural language related to the will and interest of the built software. The mechanism of finding information can be done through observation, questionnaire, interview, and documentary study, which are then tested with data validation and verification in FGD. The product of requirements elicitation is the User Requirement Document (URD) that can be used as an early artifact in software development process. Furthermore, URD is used as a reference in tracking the suitability between the function of software and the needs of stakeholder. This article formulated analysis approach of SE need, so that it is easily modified, adaptive to growing software complexity, and connected with the analysis of hardware.

Keywords: *elicitation; requirements analysis; software engineering.*

1. Introduction

Information technology has involved almost all aspects of life. Technology is built to alleviate human complex jobs to become much simpler. Due to this purpose, most organizations use technology as a facility in doing their jobs and achieving organizational goals[1]–[3]. Almost all organizations today have utilized information system application (software) as a means to help their jobs in solving various problems. Software design should be able to work holistically to handle problems the software face.

Information systems (IS) is a combination of information technology utilizations and human activity upon a set of agreed procedure[4], generally is used to support management and operation[5]. IS is an organized data process[6], IS has a high level of flexibilities to develop and scalable[7]. Refers to several research, the information system has an accurate data accessibility and efficient run-time [8], high accuracy [9], and to support a proper decision [10], low cost [11], extended accessibility [3], intensify user knowledge [12], increase productivity [13], provide a better data and information [14], and used as data storage [15].

In designing software design, needs analysis is the first stage to be the basis for software design. As whole, the smoothness of software design and the completeness of system feature resulted is highly dependent on the results of the needs analysis. The analysis phase aims to find the user and organization's needs and to analyze the existing conditions. The incomplete result of needs analysis potentially causes some unexpected problems, namely: the time of making system is longer and the system cannot meet all the user's needs. To minimize this risk, it is recommended to make confirmation by asking the resume of analysis results needed and complete the result of analysis[16].

This article is intended to discuss the concept of needs elicitation in software engineering. Elicitation is a process of searching, browsing, and understanding with more detailed, deeper, and comprehensive so it is likely to be little information left [17], [18]. Thus, needs elicitation is a process in which the search for information is more deeply and comprehensive from all stakeholders associated with the built software. The result of needs elicitation is every stakeholder's explanation in natural language dealing with the wishes and interests to the built software. The product of needs elicitation is the User Requirements Document (URD) that can be used as an early artifact in the process of software development. URD may also be used as a reference in tracking the function of software whether it is already in line with the need of stakeholders.

2. Methodology/ Materials

The methodology used in this article is literature review as a major source of problem analysis. The analysis methodology followed an analysis model recommended by Ramdhani et al., 2014[19], and Ramdhani et al., 2014 [20]. In the next phase, there have been done clarification, validation, and verification of the main topic of this article through the model of Focus Group Discussion (FGD). The participants of FGD are seven lecturers of Informatics Department of UIN Sunan Gunung Djati Bandung. The discussion in this article is limited to needs elicitation in software engineering.

Copyright © 2016 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

3. Results and Findings

Software Engineering (SE) is a discipline, concept, method that focuses on all aspects related to software production [17], [18]. Software engineer must be able to read and adapt organized system and approach in building software. This includes the use of appropriate tools and techniques based on the problem to be solved, limitations, user's needs and desires and the availability of resources. The focus of software engineering is related to various theories, methods, and tools used for professional software development (systematic, disciplined, and measured toward the development, operation and maintenance of software), in order to produce excellent software.

In general, the information required in needs analysis of making software consists of information regarding the functional and non-functional requirements of the designed software which covers: (a). system architecture; (b). data flow architecture; (c). Data Flow Diagrams (DFD); (d) System user; (e) Main needs of system; and (f) Specifications of device[21]. To obtain such information, the concept of needs elicitation in software engineering is presented in Figure 1.

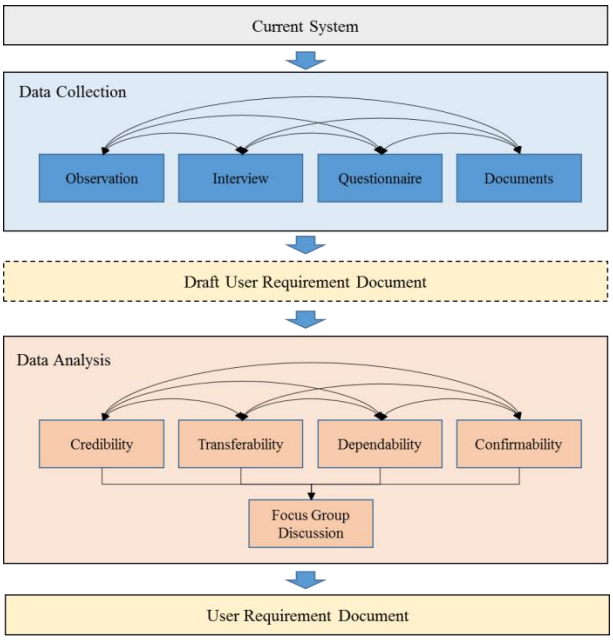


Fig. 1. Concept of needs elicitation in software engineering

3.1. Data Collection Technique

Data collection technique for needs elicitation in software engineering was done to find the user's desires, needs, and limitation, through the collection of data which URD draft was subsequently drawn up to get the validation and verification. There are several data collection techniques, namely: observation, interviews, documentary study and questionnaire. The author recommends these four techniques to be done well, intact, and comprehensively. Here is a brief description of the technique of data collection:

a. Observation:

Observation is a method of data collection used to get data through observation and sensing where the observer actually stays in everyday life of the subject of study or informant, the observer can engage in active and inactive manner[22]. Observation method is done by observing the behavior, event or activity of a person or group of people in current system. Then he records the result of the observation to find out what really happens. Through observation, software designer can see an event as the subject of study experiences, catches, feels the phenomenon in accordance with the understanding of the subject and object being observed.

b. Interview:

The interview is an activity performed to directly obtain information by asking the software designer and interviewees. The success of the interview depends on the skills the software designer has in gaining the trust of interviewees. The skill covers how to ask questions, make the order of questions, how to listen, how to express verbally such as intonation and speed of sound, and non-verbal expression such as eye contact, patience and attention to listening the answers and conditioning the comfortable situation[22].

c. **Document**:

In designing software design needs, there was required secondary data from relevant documents to the needs of system that aims to enrich data and information[21]. Document is written record/ stored image on something that has already happened. Documents are the fact and the stored data in various materials in the form of documentation. Most of the available data are letter, report, regulation, diary, biography, symbol, artifact, photograph, sketch and other stored data. Documents is not restricted in space and time to give an opportunity to interested people to know something that has ever happened[22], which can then be used as the user's needs design in designing software.

d. Questionnaire:

Questionnaire is a method of data collection by providing or distributing a list of questions/ statements to the competent people hoping that respondents give the correct response to the question [23][24].

3.2. Data Analysis

Mechanism of data analysis/ needs information in designing software is basically a valid test for the information has been obtained. To establish the validity of data/ needs information in designing software can be made by concept modification of assessment technique in qualitative research[25], where the draft User Requirement Document (URD) was analyzed based on four criteria, namely;

a. Credibility:

There are some activities undertaken to increase the degree of trust, namely; (1) extending the time of needs analysis of data/ information; (2), making continuous observation of detail; (3) doing triangulation or checking data with various sources as a comparison of the data; (4) exposing the interim or final results obtained in an analytical discussion with colleagues; (5) making negative case studies by collecting cases that are not in accordance with the existing pattern as a comparison; (6) comparing with the results of similar analyzes in a case/ another organization and; (7) checking data, interpretation, and conclusion with fellow members of the software designers.

b. Transferability:

Transferability means whether the result of the acquisition of data/ information can be applied to other situations (or not), so that the software can be adaptive and developed according to the needs.

c. Dependability:

Dependability means whether the result of the acquisition of data/ information refers to the consistency of the designer's system (software) to collect data, establish, and use concepts when making interpretation to draw conclusions.

d. Confirmability:

Confirmability or certainty means whether the result of the collection of data/ information can be substantiated, where draft of URD is in accordance with the collected data. This is done by discussing the draft of URD with software engineering experts who do not participate and are not interested in designing software, expected that the result of the analysis of data/ information can be more objective.

One of quick techniques used for clarifying the result of the acquisition of information can be done through Focus Group Discussion (FGD). FGD is one of the data collection techniques designed to gain the information of needs, desires, beliefs, perspectives and experiences of participants on a topic, which is carried out with a facilitator or moderator's guidance[26]. In this FGD, draft user required system is analysed jointly by the competent and interested parties.

3.3. User Requirements Document and Software Requirements Specifications

URD (User Requirement Document) contains detailed information obtained from the elicitation from all stakeholders with natural language. Here is an example of a natural language as outlined in URD:

"I want the visualization of the report that concludes how many resellers, dropshipper, and active agents to sell every day appear in the application. If you can make its shape like dashboard to be more eye-cacthing so that the movement is more visible, including the control or evaluation of our everyday income. "

In fact, URD may be the result of conversations resulted from communication with stakeholders regarding needs that must exist in the software to be built. Within it, there is the business of process, policy, and legal rule of the organizations associated with the software. Besides as an early artifact, URD may also be used as a reference for the contract agreement between developer and stakeholder. The search of entire good software development process must achieve stakeholders' needs described in the URD.

After performing requirements elicitation and producing URD, natural language presented is mapped in the language of software requirement with the viewpoint of system or software. Furthermore, it is grouped into functional requirement and non-functional requirement, and given the scale of priorities for each software requirement. Needs engineering is useful to help software developers better understand the problems to be solved, which results in written software requirements specification from communication to modeling.

Software requirement can be a statement that is expressed in natural sentences to more detailed mathematical sentences[17], [18]. Where, software requirement will be the basis for the scope of the software design implementation. In addition to the user's needs, there are some types of needs, namely: Functional Needs, Non-Functional needs, and domain issues based-needs

3.4.1 Functional Requirement

Functional requirement is a statement of service the software will provide, how software will react to input from outside (Sommerville, 2011). The statement of need must be provided by software in the requirements specification document of software. The statement of needs presented by the software point of view should be complete and detailed enough so that it can be used as the basis for the next design. Software requirements specification describes what the software must provide, not how the software is built. Here are presented the examples of functional requirements of software in the library information system:

- P/ L provides the facility to manage the data of borrowing books, including data addition, deletion, and modification
- P/ L provides the facility to search books by title, author, or publisher

There are some ways to describe the software functional requirements, as shown in Table 1. (There is additional column of the priority of needs)

3.4.1 Non-Functional Requirement

In contrast to the functional requirement, non-functional requirement stated limitation to the services that software serves, such as time limits, restriction of process model, standard limit, and others that can be measured. Non-functional requirement can also be said the measurable needs that support functional requirement of the software (Table 2). For example:

- searching book must be done in a short time, ie within a maximum of 5 seconds
- constructing P/ L should utilize MySQL DBMS

Table 1. Sample of Software Functional Requirements[27]

Code of Functional Requirement	Requirement Name	Description
LP-F-01	Receiving documents collection	Library can receive collection input of text documents to be prepared as an input of structure representation formation. Hereinafter it is named to input 1.
LP-F-02	Receiving preprocessing result	Library can receive the results of preprocessing as an input of structured text representation formation. Hereinafter it is named to input 2.
LP-F-03	Receiving support	Library can receive the input of support parameters from the user to make the representation of HFWI. Hereinafter it is named as input 3.
LP-F-04	Reading document collection	Library can read all files in input 1
LP-F-05	Separating sentence	Library can separate sentences in the document collection as representation needs of HFWI.
LP-F-06	Lowercase	Library can change the entire document to lowercase.
LP-F-07	Removing stop words	Library can remove stopwords that exist in the entire document including removing punctuation, characters other than letters, symbols, white space and white line excess in input 1.
LP-F-08	Stemming	Library can change the contents of the input 1 to the basic word form.
LP-F-09	Storing Preprocessing result	Library can store the result of preprocessing input 1 into a file as input 2. Hereinafter it is named file 1
LP-F-10	Reading input 2	Library can read the input 2 as input of HFWI representation formation.
LP-F-11	Creating FWI	Library can form FWI representation with FP-Growth algorithm of tools SPMF and algorithms CP-Tree. Hereinafter it is named output 1.
LP-F-12	Storing FWI	Library can store the representation of FWI, hereinafter it is named file 2.
LP-F-13	Selecting MFS from FWI	Library can select FWI that has been formed in file 2 and form sets of MFS.
LP-F-14	Storing MFS from FWI	Library can store MFS of feature selection results into a file. hereinafter it is named file 3.
LP-F-15	Creating HFWI	Library can form HFWI representation of the file 3. Hereinafter it is named output 2.
LP-F-16	Storing HFWI	Library can store the representations of HFWI to a file. Hereinafter it is named as file 4.
LP-F-17	Selecting MFS from HFWI	Library can select HFWI that has been formed in file 4 and form a set of MFS from HFWI.
LP-F-18	Storing MFS from HFWI	Library can store MFS of feature selection results into a file. Hereinafter it is called file 5.
LP-F-19	Natural Language Processing	Library can revise abbreviated words using NLP.

Table 2: Sample of Software Non-Functional Requirements[27]

Code of Non-Functional Requirement	Requirement Name	Description
LP-NF-01	Flexibility	Library can be used easily on the application of text mining as a tool for preprocessing by providing abstract classes that can be implemented according to the needs of diverse text mining applications based on the analysis.
LP-NF-02	Scalability	Library can scaleable text mining process with incremental mining technique in order to achieve an efficient library that uses algorithms CP-Tree as an incremental mining algorithm based on the analysis.
LP-NF-03	Data Validation	Library checks all correct data entries according to the provisions
LP-NF-04	Modularity	Library is modular so that it is easy to reuse, has a low coupling and high cohesion based on the results of the analysis with class and function designs that minimize dependence or interrelation between classes and between functions.

4. Conclusion

Requirements elicitation is the process of in-depth and thorough finding information from all stakeholders associated with the built software. The result of requirements elicitation is the explanation from every stakeholder with natural language related to the will and interest of the built software. The mechanism of finding information can be done through observation, questionnaire, interview, and documentary study, which are then tested with data validation and verification in FGD. The product of requirements elicitation is the User Requirement Document (URD) that can be used as an early artifact in software development process which is further mapped into the software requirements specification.

References

[1] Y. Bustomi, M. A. Ramdhani, and R. Cahyana, "Rancang Bangun Sistem Informasi Geografis Sebaran Tempat Riset Teknologi Informasi di Kota Garut," *J. Algoritm.*, vol. 9, no. 1, pp. 1–7, 2012.

[2] A. Tsabit, M. A. Ramdhani, and R. Cahyana, "Pengembangan Ganesha Digital Library untuk Membuat Situs Jurnal," *J. Algoritm.*, vol. 9, no. 1, pp. 1–10, 2012.

[3] C. Slamet, A. Rahman, M. A. Ramdhani, and W. Darmalaksana, "Clustering the Verses of the Holy Qur'an Using K-Means Algorithm," *Asian J. Inf. Technol.*, vol. 15, no. 24, pp. 5159–5162, 2016.

[4] A. Pamoragung, K. Suryadi, and M. A. Ramdhani, "Enhancing the implementation of e-Government in indonesia through the high-quality of virtual community and knowledge portal," in *Proceedings of the European Conference on e-Government, ECEG*, 2006, pp. 341–348.

[5] M. A. Ramdhani, *Metodologi Penelitian untuk Riset Teknologi Informasi*. Bandung: UIN Sunan Gunung Djati Bandung, 2013.

[6] D. S. Maylawati, W. Darmalaksana, and M. A. Ramdhani, "Systematic Design of Expert System Using Unified Modelling Language," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12047, 2018.

[7] H. Aulawi, M. A. Ramdhani, C. Slamet, H. Ainissyifa, and W. Darmalaksana, "Functional Need Analysis of Knowledge Portal Design in Higher Education Institution," *Int. Soft Comput.*, vol. 12, no. 2, pp. 132–141, 2017.

[8] C. Slamet, A. Rahman, A. Sutedi, W. Darmalaksana, M. A. Ramdhani, and D. S. Maylawati, "Social Media-Based Identifier for Natural Disaster," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12039, 2018.

[9] C. Slamet, R. Andrian, D. S. Maylawati, W. Darmalaksana, and M. A. Ramdhani, "Web Scraping and Naïve Bayes Classification for Job Search Engine," vol. 288, no. 1, pp. 1–7, 2018.

[10] Y. A. Gerhana, W. B. Zulfikar, A. H. Ramdani, and M. A. Ramdhani, "Implementation of Nearest Neighbor using HSV to Identify Skin Disease," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 012153 1234567890 Implementation, 2018.

[11] A. Rahman, C. Slamet, W. Darmalaksana, Y. A. Gerhana, and M. A. Ramdhani, "Expert System for Deciding a Solution of Mechanical Failure in a Car using Case-based Reasoning," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12011, 2018.

[12] D. S. Maylawati, M. A. Ramdhani, W. B. Zulfikar, I. Taufik, and W. Darmalaksana, "Expert system for predicting the early pregnancy with disorders using artificial neural network," *2017 5th Int. Conf. Cyber IT Serv. Manag. CITSM 2017*, 2017.

[13] W. B. Zulfikar, Jumadi, P. K. Prasetyo, and M. A. Ramdhani, "Implementation of Mamdani Fuzzy Method in Employee Promotion System," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12147, 2018.

[14] D. S. Maylawati, M. A. Ramdhani, A. Rahman, and W. Darmalaksana, "Incremental technique with set of frequent word item sets for mining large Indonesian text data," *2017 5th Int. Conf. Cyber IT Serv. Manag. CITSM 2017*, pp. 1–6, 2017.

[15] A. Taofik, N. Ismail, Y. A. Gerhana, K. Komarujaman, and M. A. Ramdhani, "Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 288, no. 1, p. 12018.

[16] A. Muhammad, E. Rianti, and I. Effendi, "Penerapan dan Perancangan Aplikasi Customer Relationship Management dalam Meningkatkan Fungsi Supplayer untuk Penjualan Sepeda Fixie Beserta Aksesoris Spare Partnya," *Maj. Ilm. UPI YPTK*, vol. 20, no. 2, pp. 52–61, 2013.

[17] R. S. Pressman, *Software-Engineering: A Practitioner's Appoarch*, Seventh. New York: McGraw-Hill, 2010.

[18] I. Sommerville, *Software Engineering*. Addison Wesley, 2011.

[19] A. Ramdhani, M. A. Ramdhani, and A. S. Amin, "Writing a Literature Review Research Paper: A step-by-step approach," *Int. J. Basic Appl. Sci.*, vol. 3, no. 1, pp. 47–56, 2014.

[20] M. A. Ramdhani and A. Ramdhani, "Verification of Research Logical Framework Based on Literature Review," *Int. J. Basics Appl. Sci.*, vol. 3, no. 2, pp. 1–9, 2014.

[21] M. N. Sasongko, K. Amaliah, M. A. Ardiansyah, and A. A. Wiratama, "Analisis Kebutuhan pada Perancangan Sistem Informasi Sebaran Usaha Mikro, Kecil dan Menengah di Kabupaten Klaten," *Rekayasa Teknol. Ind. dan Inf.*, pp. 108–112, 2016.

[22] A. R. Djaelani, "Teknik Pengumpulan Data dalam Penelitian Kualitatif," *Maj. Ilm. Pawiyatan*, vol. 20, no. 1, pp. 82–92, 2013.

[23] S. Muryani, P. D. Paramita, and A. Fathoni, "Pengaruh Pengalaman Kerja, Pengawasan Kerja, dan Spesialisasi Kerja terhadap Pemahaman Beban Kerja dengan Pemanfaatan Teknologi Informasi sebagai Variabel Intervening," *J. Manage.*, vol. 2, no. 2, 2016.

[24] T. N. Halimah, A. Fathoni, and M. M. Minarsih, "Pengaruh Job Insecurity, Kepuasan Kerja, dan Lingkungan Kerja terhadap Turnover Intention Pramuniaga di Gelael Supermarket," *J. Manage.*, vol. 2, no. 2, 2016.

[25] L. J. Moleong, *Metodologi Penelitian Kualitatif*. Bandung: Remaja Rosda Karya, 2005.

[26] A. Paramita and L. Kristiana, "Teknik Focus Group Discussion dalam Penelitian Kualitatif," *Bul. Penelit. Sist. Kesehat.*, vol. 16, no. 2, pp. 117–127, 2013.

[27] D. S. Maylawati, "Pembangunan Library Pre-processing untuk Text Mining dengan Representasi Himpunan Frequent Word Itemset: Studi Kasus Bahasa Gaul Indonesia," Bandung, 2015.