Supporting of Requirements Elicitation for Ensuring Services of Information Systems Used for Education

Yuki Terawaki

The University of Tokyo Graduate School of Arts and Sciences, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
Research Center for Computing and Multimedia Studies, Hosei University, 3-7-2 Kajinocho, Koganei, Tokyo 184-8584, Japan
terawaki@graco.c.u-tokyo.ac.jp

Abstract. This paper describes a method to ensure the quality requirements from service receiver in the requirement definition phase of system development. The proposed method measures the quality characteristics that are in the requirement document using the text-mining technique and concept dictionary and identifies requirements of document with quality characteristics of the International Standards Organization (ISO) / the International Electrical technical Commission (IEC) 9126-1:2001[1]. The case study shows that the quality characteristics are contained in the requirements document.

Keywords: Requirements Engineering; RE; Non-Functional Requirements; ISO/IEC 9126; text-mining approach.

1 Introduction

This paper describes a method to identify the service receiver's expectation of quality for developing system. Service receiver's needs are usually focused on not only the functional requirements to be met but also on quality characteristics, such as how soon the system can be run, how easy it is to use, or how often a system failure can occur etc. Such characteristics, collectively known as software quality characteristic or quality factors, are part of the system's non-functional requirements. It is widely recognized that in real systems, meeting the functional requirements often is more important than meeting the functional requirements in the determination of a system's perceived success or failure [11]. Therefore, in order to ensure services of developing system, the essential elements in the requirement definition are followings; The IT system-specific issues from various needs that were acquired from the stakeholders need to be selected. And then quality characteristics as much as possible need to be acquired and defined.

However, in the requirements acquisition phase of requirement definition, functional requirements are highly focused, non-functional requirements are not necessarily sufficiently defined. Moreover, some of the quality characteristics that are supposed to be fulfilled can occasionally be left unstated and implicit requirements because service receiver's requirements can be contradicted in requirements definition. Thus, to check

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part I, HCII 2011, LNCS 6771, pp. 58–65, 2011.

the quality characteristics carefully is necessary for preventing of the implicit requirements but it also avoids a situation where users complain that the requirements are not all met when the developed system is delivered.

Therefore, this paper proposes a method measures the quality characteristics that are in the requirement document using the text-mining technique and concept dictionary and identifies requirements of document with quality characteristics of the ISO/IEC 9126-1.

The paper takes the following format. In section 2 describes the process of the requirements definition, and the non-functional requirements. In section 3 outlines the ISO/IEC 9126. The proposed method and its implementation are described in section 4. In section 5 introduces case studies. In section 6 conclusion and future works are provided.

2 Requirements Process and Non-functional Requirements

The RE Process has 4 processes. There is requirements elicitation, evaluation, specification (documentation) and quality assurance. This process is iteration on successive increments according to a spiral model [5,6]. A new iteration may take place at different stages of the software lifecycle. This spiral process model is fairly general and flexible. By participating both of provider and receiver (stakeholder) in the spiral process, requirements document will become elaborate. A stakeholder is a group or individual affected by the system-to-be, who may influence the way this system is shaped and has some responsibility in its acceptance. Stakeholders play an important role in the RE process. Because the problem of system embedded in human activity arises within some broader context. By stakeholder, requirements that has different viewpoint are elicited. Some of elicited requirements may include goal or problem on providing service. These are requirements against property of activity outside software. This is a non-functional requirement in the broad sense. Non-functional requirements define constraints on the way the software-to-be should satisfy its functional requirements or on the way it should be developed. Characteristics that fall into this category include how quickly it runs, how often it fails, and how easy it is to use.

Such characteristics, known as software quality attributes, as defined by ISO/IEC9126. Excellent software products reflect an optimum balance of competing quality characteristics. Quality attributes are difficult to define [10], because customers generally don't present their quality expectations explicitly. However in order to ensure services of developing information system, non-functional requirements should be well defined. Therefore, it is necessary to ascertain whether quality attributes were elicited during requirements definition.

3 ISO/IEC9126

ISO and IEC have developed the ISO/IEC 9126 Standards for Software Engineering – Product Quality [1-4] to provide a comprehensive specification and evaluation model for the quality of software products. The ISO/IEC 9126 standard is divided into four parts, under the general title Software engineering- Product quality:

- Part 1. Quality model
- Part 2. External metrics
- Part 3. Internal metrics
- Part 4. Quality in use metrics

The first three parts are concerned with describing and measuring the quality of the software product; the fourth part evaluates the product from the user point of view. We focus on part 1 (ISO/IEC9126 -1) which defined the quality model for external and internal quality.

ISO/IEC 9126-1 describes two-part model for software product quality: a) internal quality and external quality, and b) quality in use. The first part identifies the quality of a software product through six quality characteristics, namely: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. Each characteristic is subdivided into related sub-characteristics. These sub-characteristics are manifested externally when the software is used as a part of a computer system, and are a result of internal software attributes.

ISO/IEC9126-1 can be used not only the purpose to evaluate the quality of the developed software but also to define the non-functional requirements. Moreover, ISO/IEC 9126-1 also is used for infostructures' design. Examples of uses of the quality model are followings:

- Validate the completeness of a requirements definition;
- Identify software design objectives;
- Identify software testing objectives;
- Identify quality assurance criteria;
- Identify acceptance criteria for a completed software product.

4 Proposed Method

This method analyzes the rate of content of the quality characteristics in the requirements documents such as Request For Proposal (RFP). This method is able to calculate rate of content of whole document and each statement. The overview is described in Fig 1.

4.1 Morphological Analyzer and Index Generator

This method consists of two tools: one is the morphological analyzer (MA) and the other is the index generator (IG). MA breaks down the requirements specifications into sentences, and then each sentence is separated by morphological analysis. After that, MA tally the number of word and frequency of appearance, then makes term-document matrix.

In IG has some files. Each file include statements that represented the quality characteristics of ISO / IEC 9126. ISO / IEC 9126 consists of six quality characteristics so that IG also has six files. Each file of IG will contain the statements reviewed by specialists. IG first performs a morphological analysis against each file and removes characteristic words. Then, IG creates the frequency file. When frequency file are created, synonyms are added by using the concept dictionary. The reasons for adding

synonyms are as follows. For example the word "user" is able to represent "customer" "End-user", or "consumer". Moreover The Japanese word for "user" is "riyousya" but "u-zer" is also recognized as Japanese. These meanings are identical, and such synonyms need to be considered within proposed method. After that, IG creates Word Article Matrix (WAM) file by executing mkw commands of Generic Engine for Transposable Association (GETA) [8]. Finally, this method analyzes the rate of content of the quality characteristics in the requirements documents using the term-document matrix in MA and the WAM file created in IG.

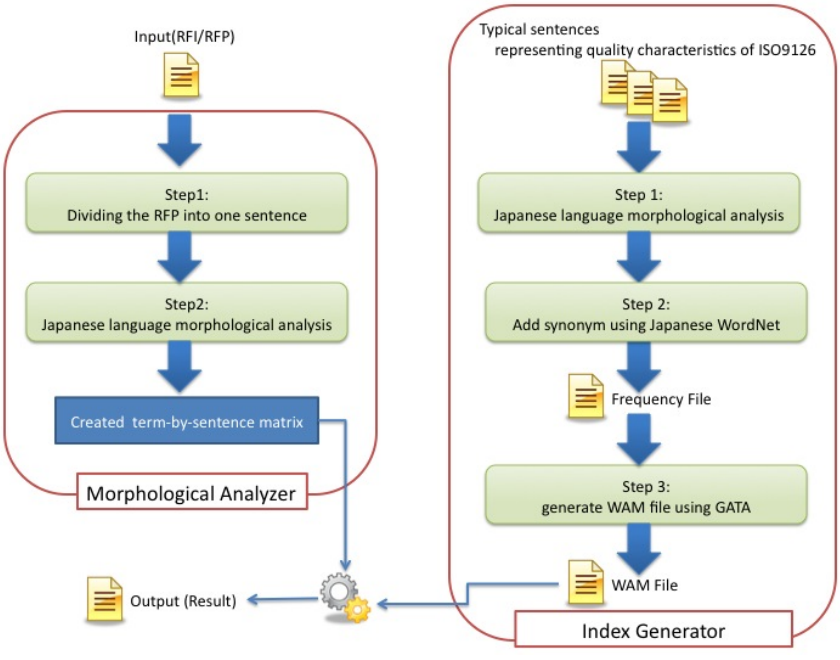

Fig. 1. Conceptual Diagram

4.2 Implementation

This method was developed as a CUI application by using Java programming language and shell scripts. The Japanese morphological analyzer Sen [7] is employed for morphological analysis in MA and IG. Synonyms are added to IG through Japanese WordNet [9], a concept dictionary, and extracted through a method that first acquires a lower level of the word group against the original words and then acquires the sum of the sets of each upper level word group. For the WAM file creation in IG and search operations in MA, a generic engine for transposable association (GETA) is employed. In IG, the mkw GETA command is used in the script. Since the search parts in GETA are provided only as the C library, GETA must create an execute format to wrap the I/O for the connection by using the Java program and standard I/O. The results will be output in CSV format, which facilitates easy use of the scores included in the search results.

5 Case Study

5.1 Overview

This section describes the trial practice using the proposed method. The RFP used for this trial practice was created in the university. This RFP involves replacing the network infrastructure and getting software (web-based application) across the university. In 2006, this RFP was already created by their system implementation committee after multiple reviews. However, author of this RFP is not the expert in Requirements Engineering (RE). In short, author didn't know the ISO/IEC 9126.

In this trial, the section relating to software, particularly Learning Management System (LMS) was picked out the RFP. This portion has about 4,600 characters.

Initially, This portion was analyzed manually by two experts of RE. After that it was analyzed by proposed method.

5.2 Result

Fig2 shows result of analysis by manually. Fig3 shows result of analysis by proposed method. The tendency of whole document denotes the same tendency. In this RFP, usability and functionality are describing most frequently requirements. However, portability was less described.

On the other hand, Fig4 shows quality characteristics contained in each sentence. For example, sentence of No.27 consists of Functionality, Reliability, and Usability. In their quality characteristics, Functionality is most remarkable tendency. Sentence of No.52 consists of Functionality, Usability. Functionality has much more higher than Usability.

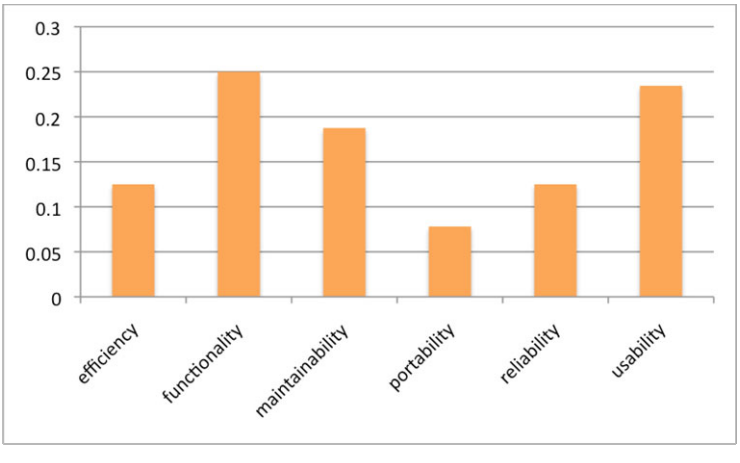

Fig. 2. Result of analysis by manually

5.3 Discussion

The results of analysis by manually and by proposed method were compared, and then these tendencies were similar. Let's take a look at each sentence Fig 4.

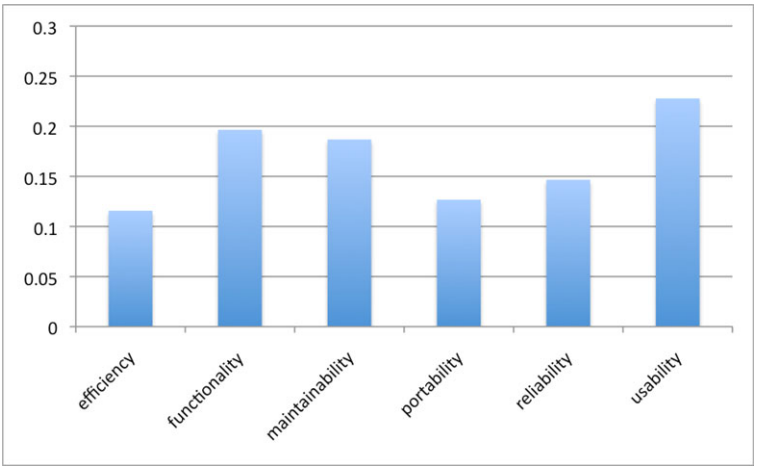

Fig. 3. Result of Analysis by Proposed Method

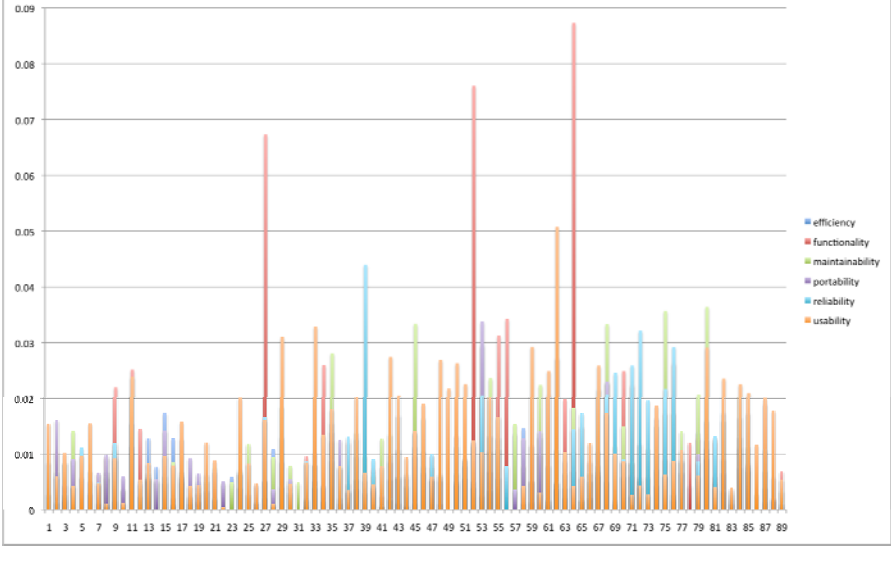

Fig. 4. Quality Characteristics in Each Sentence

For example, sentence of No.27 and No. 52 has features of Functionality. The requirement of No.27 is "The teaching material corresponding to international standard SCORM1.2 or more should be able to be published." Manually powered analyzing determined that the requirements were represented Functionality. The requirement of No. 52 is " When there is a difference in the method for SSO between LMS and University, method of SSO for LMS should be customized." This requirements was

deemed Functionality. This result also matched the analysis by manually. There are great examples. When two results are different, statement of target and the frequency file in IG need to be closely examined as to how different features were produced. At this time, when two results are different, we found that it is often the case that requirements sentence is subjective. These sentences supplies during requirements elicitation some clues about what they have in mind. On the other hands, some previous studies pointed out quality characteristics of ISO/IEC 9126 has ambiguity and omission [12]. Thus, Which characteristics of ISO9126 are most crucial to project success, we need to carefully consider.

6 Conclusion

In this paper, we illustrated a method and its implementation to ensure quality requirements from service receiver in the requirement definition phase of system development. The quality characteristics that are in the requirement document can be measured by using proposal method. Result of case study, the proposed method shows mostly good performance. Several issues are found through the case study in this paper.

- Remove some specific words

In any specifications there are some specific words that users wish to avoid considering as a morpheme. (E.g. 「情報メディア教育研究センター」 is 「情報:メディア:教育:研究:センター」). As such, it is not appropriate that there are some words, which are supposed to be treated as specific words, that are broken down into morphemes and reflected in the output of this framework. Therefore, we would like to review some words further for removal as specific words.

- Acquire synonyms

Though we acquire synonyms from the Japanese WordNet, acquiring them with no restrictions may create gaps between the original meanings. Currently, the method acquires lower level word groups against the original words and then acquires the upper level word group for each word in the lower level groups to determine the sum of the sets in the upper level word group. The top ten synonyms are then added. The number of synonyms added should be determined by continuously conducting actual case studies. Alternatively, the desirable number should be determined by examining the previous studies in the text-mining field.

Quality attributes are difficult to define, yet often they distinguish a product that merely does what it's supposed to from one that delights its user. However, service receivers generally don't present their quality expectations explicitly. Quality, in its many dimensions, must be defined from some clues about what they have in mind. In order to ensure service, the visible characteristics may be important. By gaining a lot of case study, the proposed method will be refined and then we'd like to learn which quality characteristics are most important to ensure service of information systems used for education.

References

1. ISO/IEC, ISO/IEC 9126-1 Software engineering- Product quality- Part 1: Quality model (2001)
2. ISO/IEC, ISO/IEC 9126-2 Software engineering- Product quality- Part 2: External metrics (2002)
3. ISO/IEC, ISO/IEC 9126-3 Software engineering- Product quality- Part 3: Internal metrics (2002)
4. ISO/IEC, ISO/IEC 9126-4 Software engineering- Product quality- Part 4: In Use metrics (2002)
5. Boehm, B.W.: A Spiral Model of Software Development and Enhancement. IEEE Computer 21(5), 61–72 (1988)
6. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley & Sons Ltd., Chichester (1997)
7. Sen (December 2010), http://ultimania.org/sen/
8. GETA (December 2010), http://geta.ex.nii.ac.jp/e/index.html
9. WordNet (December 2010), http://nlpwww.nict.go.jp/wn-ja/index.en.html
10. Bratthall, L., Wohlin, C.: Understanding Some Software Quality Aspects from Architecture and Design Models. In: 8th IEEE International Workshop on Program Comprehension (2000)
11. Charette, R.N.: Applications Strategies for Risk Analysis. Hill Software Engineering Series (1991)
12. Al-Kilidar, H., Cox, K., Kitchenham, B.: The Use and Usefulness of the ISO/IEO 9126 Quality Standard. In: International Symposium on Empirical Software Engineering (2005)