# A Practical Approach to Quality Requirements Handling in Software Systems Development

Yuki Terawaki
Research Center for Computing and Multimedia Studies
Hosei University
Tokyo Japan
yuki.terawaki.dc@k.hosei.ac.jp

Tetsuo Tamai
Department of Advanced Sciences, Faculty of Science and Engineering
Hosei University
Tokyo Japan
tamai@hosei.ac.jp

*Abstract*— **Quality requirements (non-functional requirements or NFR) are vital for the success of software systems. Therefore, to define the quality requirements and to check the quality attributes carefully is necessary for bringing good-quality software and ensuring quality of the service. This paper proposes a framework that measures the quality attributes in the requirement document. The output of proposed framework shows where quality attributes are contained and how much. The framework's objectives are to assist defect detection of requirements statements with focus on quality requirements. We analyzed the requirement specification document using the framework, and confirm the efficacy of this framework.**

***Keywords; Quality Requirements; Non-Functional Requirements;Text-mining Approach***

## I. INTRODUCTION

Many of the problems in a software development project are caused by faults in the requirements that properly the customer (or user) should determine. It is said that items of re-work due to faults in the requirements account for 30~50% of the total cost of development, resulting in cost overruns [1]. Furthermore, the expenses for correction of defects discovered in the latter half of the lifecycle of software development become more massive than the expenses for correction during the first half [2]. For these reasons, it is important to ensure quality of description content of documents that created in upper process of software development such as RFP (Request for Proposal) or SRS (Software Requirements Specification). Requirements that have to be written in these documents are functional requirements and non-functional requirements (or quality requirements) [3].

A functional requirement describes the software's behavior (what is to be executed by the software), and a non-functional requirement is a description which indicates how well the software's behavior is to be executed. It is widely recognized that in real systems, meeting the quality requirements often is more important than meeting the functional requirements in the determination of a system's perceived success or failure [4].

In IEEE830-1998 [15], the basic issues that the SRS writer(s) shall address are the following: Functionality, External interfaces, Performance, Attributes, Design constraints. Especially, IEEE830 insists on the importance of Attributes.

Software quality attributes have to be specified so that their achievement can be objectively verified because there are a number of attributes of software that can serve as requirements. Therefore, this paper proposes a framework that measures the quality attributes in the requirement document. The proposed framework supports to improve the SRS and to lighten the human workload in carrying out Inspection as well.

The paper is organized as follows. In section 2, we point out the current problem. In section 3, our approach is described. The proposed framework and the implementation of tool are described in section 4. Section 5 introduces case studies. The related works are stated briefly in section 6. In section 7 conclusion and future works are provided.

## II. THE CURRENT PROBLEM

Despite a general awareness of software quality attributes, functional requirements are highly focused and the quality attributes are not necessarily sufficiently defined. [4]. Quality attributes are difficult to define [6], because customers generally don't present their quality expectations explicitly. An independent administrative institution, Information-technology Promotion Agency (IPA), Japan study shows that NFR demand has a rate of documentation lower than functional requirements.

To address the difficulty of definition of NFR, ISO/IEC25010 is available. The quality model of ISO/IEC 25010 defines the quality attributes which should be considered in software and system development. ISO/IEC25010 defines seven ways to use the quality model [9]. Here are some of uses of the quality model of ISO/IEC25010: i) identifying software and system requirements, ii) validating the comprehensiveness of requirements definition, iii) identifying acceptance criteria for a software product and/or software-intensive computer system.

We can apply a sentence of requirements to each characteristic of the software quality attributes of ISO/IEC 25010 and can check the requirements for quality. However, it is difficult to evaluate correspondence with attribute and requirements for a general reason. This is because some quality requirements overlap two or more quality attributes. Also, when identifying attributes and requirements, human judgment may change over time. Thus, it is difficult to review every quality requirements in terms of coherent thinking.

 ISBN: 978-1-61208-246-2 

Though software development requires quick delivery today, it is not unusual for development documents (such as SRS or RFP) to be over several hundred pages long. As the scale of the SRS gets bigger, the structure of the SRS becomes complex. At present, despite the increasing number of documents which should be inspected, shortening of development time is desired. Additionally, almost all development documents are described in natural language. If the quality requirements needed are written to the document created to the upper process, quality can be measured at the time of the acceptance inspection. However, it is difficult to review every quality attributes in terms of time. These problems bring deterioration in the quality of development document and play a role in the failure of the project.

## III. OUR APPROACH

When we manage the quality, we have to know the object of management. As mentioned in section 2, in order to improve a SRS with low quality, we focused on software quality attribute (ISO/IEC25010). If the quality attributes can be quantitatively measured, then they could potentially help the author of SRS decide if a revision is needed.

In order to solve the problem described in section 2 (in terms of coherent thinking and time), text-mining technique is effective. The Requirements Process has 4 processes. There are requirements elicitation, evaluation, specification (documentation) and quality assurance. This process is iteration on successive increments according to a spiral model [5]. In the spiral process, when requirements document will become elaborate, the error of requirements may be made.

In spiral process, the revised document (SRS) can check without spending hours as much as possible using text-mining technique. The quality attributes contained in SRS are showed quantitatively because the text mining analyzes where quality attributes are contained, and how much. The rate of documentation of quality attribute can be showed using the output of text mining. Thus, the quality requirements are checked by coherent thinking. Therefore, the workload for verification of SRS will be decreased.

## IV. PROPOSED FRAMEWORK

### A. Overview of Framework

We propose a framework to improve the quality of SRS through the requirements definition process. As criteria for evaluating the quality requirement, the quality model of ISO/IEC25010 is used. The proposed framework contains a text mining tool which can specify the statement related to quality attribute of ISO/IEC 25010.

The proposed framework analyzes the SRS by text mining to identify where quality characteristics are contained, and how much. The output of this framework provides the consistent evaluation criterion of quality requirements for revising the requirements specification.

The conceptual diagram of the framework is shown in Figure 1. When the document such as SRS or RFP is inputted to the proposed framework, the quality attributes in the document are identified. There are two outputs. One is goodness of fit to each quality characteristic of arbitrary sentences. The other is goodness of fit to each quality characteristic of whole document. From these results, it is shown where sentences expressing quality characteristics occur, and how well they fit the characteristics. Thus, the output shows sentences that need improvement in the SRS and the quality attribute which is not written becomes clear.

This framework can be used after the SRS is documented. However, SRS does not need to be completed. In the documentation process, if a elementary error [7] were to detect even as the author is making the SRS, the perfectibility of SRS would increase before quality assurance process. Moreover, in the quality assurance process, the inspector can reduce the inspection time and implement a more effective inspection. Inspection is a clearly defined procedure performed by a number of inspectors [8]. Before meetings by all the inspectors, each inspector has to read and understand all documents. So, it requires a great investment of time to complete the whole inspection. If this framework used in quality assurance process, it is possible to aim at improving the efficiency of review.

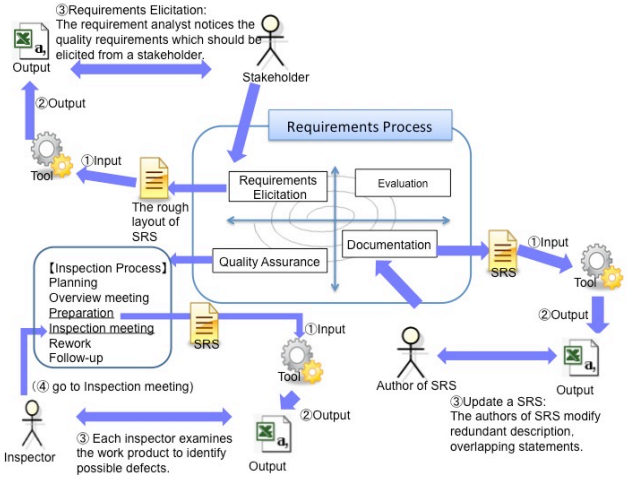


Figure 1 The Conceptual Diagram of Framework

### B. Tool with Text Mining

This Tool consists of two comportments: 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.

The IG have some files. Each file includes statements that represented the quality attributes of ISO / IEC 25010. IG has eight files so that ISO / IEC 25010 consists of eight quality attributes. The sentence in each file is reviewed by the specialist. 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

 ISBN: 978-1-61208-246-2 

the word "user" is able to represent "customer" "End-user", or "consumer". Moreover, the Japanese word for "user" is "riyousya (kanji)" but "u-zer (katakana)" is also recognized as Japanese. These meanings are identical, and such synonyms need to be considered within the proposed method. After that, IG creates Word Article Matrix (WAM) file by executing mkw commands of Generic Engine for Transposable Association (GETA) [10]. Finally, this method analyzes the rate of content of the quality attributes in the requirements documents using the term-document matrix in MA and the WAM file created in IG.

### C. Implementation of Tool

This method was developed as a CUI application by using Java programming language and shell scripts. The Japanese morphological analyzer Sen [11] is employed for morphological analysis in MA and IG. Synonyms are added to IG through Japanese WordNet [12], 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.

## V. CASE STUDY

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

In this trial, the section relating to software, particularly Learning Management System (LMS) was picked out the RS. This portion has about 4,600 characters. Initially, this portion was analyzed manually by two experts of RE. After that it was analyzed by the proposed method.

### A. Result

Figure 2 shows the result of the manual analysis. Figure 3 shows the result of the analysis by the proposed framework. It is clear from the graph that the manual analysis and the analysis by the proposed framework denote the same tendency. In this RS, usability, functional suitability and security are describing most frequently requirements. Figure 4 shows that the quality attributes are distributed throughout. In figure 4, there is a sentence showing two or more quality attributes. For example, sentence of No.45 consists of Functional Suitability (FS), Performance Efficiency (PE), and Functional Requirements (FR).

### B. Discussion

The results obtained by the manual analysis and by the proposed method were compared, and it was concluded that the tendencies were similar. Thus, it was concluded that the output of this framework is close to a reviewer's evaluation.

Let's take a look at Figure 4 of each sentence. For example, the sentence of No. 45 has described two requirements in one sentence. The first half of No.45 described, "being able to time on the use of the LMS (to answer a questionnaire)", and is "about security, performance and stability which can respond to concurrent access" in the second half. The first half of No.45 represents functional requirements and the second half represents performance efficiency. The sentence of No. 45 is undesirable sentence because one sentence described two requirements. So, this sentence needs modification. If the output of the proposed framework is provided to the author of the specification, it will contribute to its improvement.

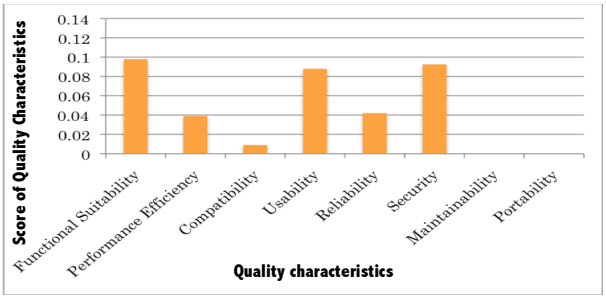


Figure 2 Result of Manual Analysis

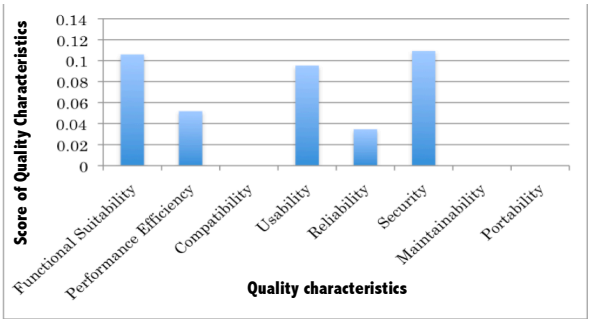


Figure 3 Result of the Analysis by the Proposed Framework

## VI. RELATED WORKS

The following researches are developing the tool which detects the defect of requirements. William M. Wilson et al proposed the Automated Requirements Measurement (ARM) [13]. The Quality Analyzer for Requirements Specifications (QuARS) was proposed by A. Fantechi et al [14]. These researches aim at pointing out the inaccuracy of the requirement specification document written by natural

language. The advantage of this research over these researches is as follows. This research provides stronger the support function for quality requirements. This framework gives the evaluation criterion of quality requirements (development documents) to the author of RS. The author of RS can focus on improving the quality requirements.

## VII. CONCLUSION

In this research, a framework for quantitatively evaluating the quality requirements was proposed. The proposed framework analyzes the requirements specification by text mining to identify where quality attributes are contained, and how much. The case study shows that the quality attributes are contained in the requirements specification. As reflected by our case study, the proposed method shows mostly good performance. This paper could give an initial evaluation of the extent to which the framework's objectives are met. It helps improve the requirements specification because framework can show the consistent evaluation criterion. And it helps improve the next requirements elicitation in the requirements definition of spiral process. We began to analyze several requirements specifications using the proposed framework. These RSs were used for the real system development. In future work we will explain that the proposed framework provides a consistent evaluation criterion of quality requirements for revising the requirements specification.

## REFERENCES


[1] Boehm Barry W., and Philip N. Papaccio, "Understanding and Controlling Software Costs", IEEE Transactions on Software Engineering 14(10), 1998, pp. 1462-1476.

[2] Grady Robert B., "An economic Release Decision Model: Insights into Software Project Management", In Proceedings of the Applications of Software Measurement Conference, , pp. 227-239, 1999.

[3] Robert N Charette, "Applications Strategies for Risk Analysis", Mcgraw Hill Software Engineering Series, 1991.

[4] Karl E. Wiegers, "Software Requirements", Microsoft Press, 2003.

[5] Kotonya, G. and Sommerville, I., "Requirements Engineering: Processes and Techniques", John Wiley & Sons Ltd. 1997.

[6] L. Bratthall and C. Wohlin, "Understanding Some Software Quality Aspects from Architecture and Design Models", 8th IEEE International workshop on Program Comprehension, 2000.

[7] Gursimran S. Walia et al., "Requirement Error Abstraction and Classification: An Empirical Study", In proceedings of the ACM/IEEE international symposium on International symposium on empirical software engineering, pp. 336 – 345, 2006.

[8] Michael Fagan, "Design and Code Inspections to Reduce Errors in Program Development", IBM Systems Jornal 15(3), pp. 182-211, 1976.

[9] ISO/IEC, "ISO/IEC 25010 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) -System and software quality models", 2011.

[10] GETA, http://geta.ex.nii.ac.jp/e/index.html, December 2010.

[11] Sen, "http://ultimania.org/sen/", December 2010.

[12] WordNet, " http://nlpwww.nict.go.jp/wn-ja/index.en.html " , December 2010 (In Japanese) .

[13] William M. Wilson et al., "Automated Analysis of Requirement Specifications", in Proceedings of the International Conference on Software Engineering ICSE97, pp. 161-171, 1997.

[14] A. Fantechi et al., "Application of Linguistic Techniques for Use Case Analysis", in Proceedings of the IEEE Joint International Conference on Requirements Engineering, pp. 157-164, 2002.

[15] IEEE, "IEEE Recommended Practice for Software Requirement Specifications IEEE Std 830-1998", 1998.


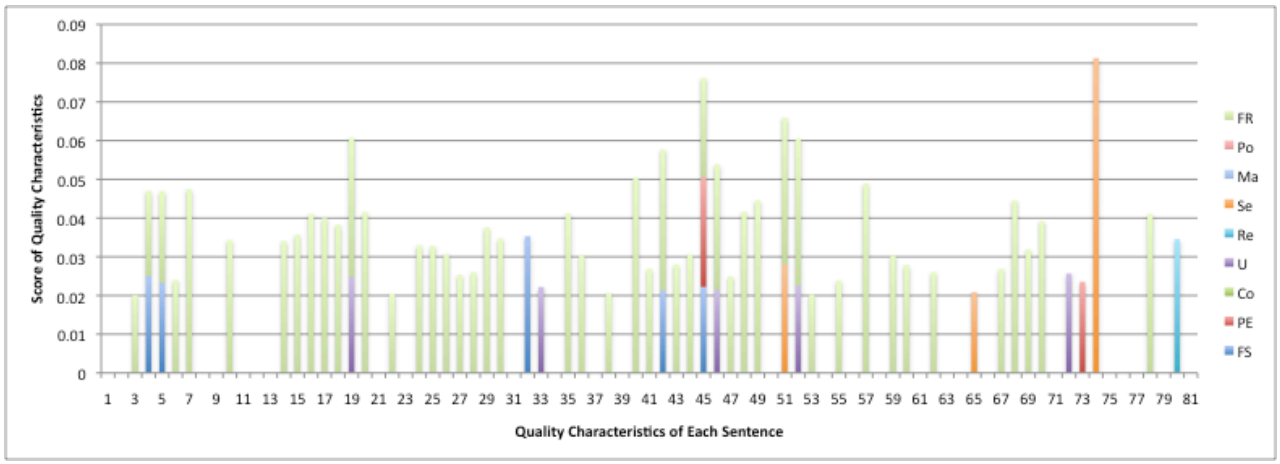


Figure 4 Quality Attributes of Each Sentence