2015 2nd International Conference on Trustworthy Systems and Their Applications

Quality Assessment Method for Software Requirements Specifications based on Document Characteristics and its Structure

Patra Thitisathienkul
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
patra.th@student.chula.ac.th

Nakornthip Prompoon
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
nakornthip.s@chula.ac.th

***Abstract*—The Software Requirements Specifications document (SRS) is used to collect the user requirements, which is used as an input for development process, and as a baseline for verifying the correctness of the software product occurring at each step throughout the software development process. It has been found that many organizations cannot deliver software products that satisfy the actual requirements of the customers, due to defects that frequently occur in the SRS, especially the use of ambiguous natural language in the requirements specifications and the inappropriate document structure, which negatively affects the software quality. Therefore, this research presents a method for objectively assessing the quality of the SRS for use of natural language in requirements specification, document structure, and overall document quality, by directly considering three characteristics of the document: unambiguous, verifiable and modifiable, to indicate the quality of the document and defects that appear during the software requirements engineering step. The process assessment model is applied as a framework for assessing the quality of the SRS, and the measurement process model and measurement information model are used as approaches for proposing a method for SRS quality assessment and defining the metric, respectively, using Pearson's correlation coefficient as the criterion for verifying the validity of the results obtained from assessing the SRS using the proposed method, indicating that the results obtained from the quality assessment reflect the quality of the SRS, as well as the apparent defects.**

Keywords—SRS Quality Assessment; Quality Assessment Method; Document Characteristics; Document Structure

I. Introduction

Currently, organizations developing software products give high importance to the delivery of quality software products under limited budget and time constraints. Software products need to be able to be used in the specified environment and satisfy the customers' requirements.

The customers' software requirements are often collected and specified as natural language, and are an important part of the software requirements specifications document. This document is one of the important software products that occurs during the collection and identification of software requirements from the users, which is in the early phases of the software development process, and is used as documentation of the commitment between the customer and the developer. This document is also used as an input to the software development process, in which the software development team uses the identified requirements to develop the software for delivery to customers, as well as a baseline for verifying the correctness of the various software products occurring throughout the software development process, including the completed software product. In addition, if there are changes in the software requirements after it has been delivered to the customer, this document may also be used to verify the correctness of the software product.

However, many organizations are unable to deliver software products that satisfy the actual requirements of the customers within the given timeframe and budget. One of the main causes comes from defects in the collection and identification of software requirements from the users, totaling 56% [1], especially in the use of natural language that may be ambiguous for identifying the customers' software requirements in the software requirements specifications document, which results in the relevant parties interpreting the software requirements in different ways, and also affects subsequent verification of the software requirements. For example, one requirement specifies that "The program shall produce output as fast as possible.". Each of the persons involved may interpret the speed of output differently. Another defect is inappropriate SRS document structure, for example, the lack of reference to the topics for the important contents of the document, or topics being in a different order from referenced, or topics not corresponding with other topics in the same category etc.. These may result in the difficulty of document understanding, and subsequent changes to the software requirements, not easily modified. These negatively affect software development, and make the software product to be delivered deviate from the actual customer requirements.

The aforementioned leads to the important question in this research of how to assess and indicate the quality of the SRS, and identify the defects of the document in the early stages of the software development process, as if the level of quality and defects appearing in the SRS can be determined in the software development process. It leads to the correction of those defects, making the SRS more complete, and also the risk reducing of defects in the later stages of the software development process.

Many researchers have proposed methods for assessing the quality of the SRS in order to determine the level of quality and

978-1-4673-9581-6/15 $31.00 © 2015 IEEE
DOI 10.1109/TSA.2015.19

defects in the early stages of the software development process. The earliest research works [2-4] applied machine learning. However, this group of research works had limitations in effectiveness in identifying the level of quality in the SRS, dependent on the training data set, and efficiency in identifying the level of quality, dependent on the applied algorithm of the learning machine. The other group of research works [5-14] studied the factors that affect the quality of the SRS, focusing on the quality of usage of natural language in identifying the requirements, then defining a metric for assessing the quality by directly considering various characteristics of the SRS. However, this group of research works also still had their limitations: some of the metrics presented in these research works still require human decisions. Also, there have been research works focusing on the quality of the document structure, which allows involved persons to use the identified software requirements for analysis, design and testing more easily, as well as easily change the contents of the document when only a small amount of change occurs in the requirements.

Therefore, this research presents a method for objectively assessing the quality of the SRS, by directly considering the characteristics of the SRS document: usage of natural language in identifying the requirements, document structure, and overall document quality, in order to identify the level of quality and apparent defects, which are information that allows the involved persons to know the quality of the SRS and the software project, as well as update the SRS to be more complete. This research will only consider three characteristics of the SRS: unambiguous, verifiable, and modifiable, as will be explained in detail subsequently. The process assessment model is applied as a framework for assessing the quality of the SRS. The model is referenced in ISO/IEC standard 15504-2 Information technology — Process assessment [15], providing recommendations on assessing the quality of the software development process. The measurement process model and the measurement information model are used as approaches for presenting the SRS quality assessment method, and defining the metric respectively. Both of these models are referenced in the ISO/IEC Standard 15939 Systems and software engineering — Measurement process [16], which is an industry- accepted standard relating to identifying essential activities and tasks for the measurement process.

The remainder of this document will explain various details as follows: The 2nd section explains the theory and knowledge related to SRS quality assessment. The 3rd section describes the presented method for assessing the quality of the SRS. The 4th section illustrates the verification of the validity of the results obtained from assessing the quality of the SRS using the presented method. The 5th section explains the important issues that should be considered when applying the presented method to assessing the quality of the SRS, and the 6th section describes the conclusions and future development approaches.

II. Background

A. *Quality Assessment*

One challenge of assessing the quality of SRS is that quality is hard to control, manage, or convert to quantitative data. The IEEE standard 610 - Computer Dictionary [17] provides two definitions for quality: the level at which the system, program component, or process follows the identified requirements and satisfies the customers' or users' needs or expectations. This research applies the ISO/IEC 15504-2 standard [15], which provides recommendations on assessing the quality of the software development process, as a framework in assessing the quality of the SRS. This standard describes a process assessment model that specifies four actions for quality assessment: 1) specifying the scope of quality assessment 2) specifying the indicators for quality assessment 3) linking the expected results to the actual production, and 4) analyzing and displaying the results of the quality assessment.

B. *Measurement Process Model*

The measurement process is an important tool in managing various activities in the software development process, such as feasibility assessment and software project progress etc., allowing the involved persons to know the state of the software project and help drive the project to success. This research uses the ISO/IEC 15939 standard [16], which integrates software requirements engineering knowledge and management to specify the necessary activities and tasks in the measurement process. The measurement process model consists of 4 required main activities: 1) establish and sustain measurement commitment 2) plan the measurement process 3) perform the measurement process and 4) evaluate measurement. This model is used as an approach for presenting the SRS quality assessment method.

C. *Measurement Information Model*

The ISO/IEC 15939 standard [16] describes the measurement information model, a model that explains the conversion of attributes, properties of entities relating to information needs, to qualitative data or information products that can be used to support a decision. This research applies the measurement information model as an approach to convert the attribute related to the quality of the SRS into information products that can indicate the quality of the SRS and its apparent defects.

D. *Metric Validation Method*

The consideration of the validity of the proposed metric, that the metric can be used for assessing what is to be measured, and the results from the assessment correspond to its actual values, is described in IEEE standard 1061 - Standard for a Software Quality Metrics Methodology [18]. This standard describes 6 criteria for assessing the validity of the metric and application of the criteria, namely 1) correlation 2) tracking 3) consistency 4) predictability 5) discriminative and 6) reliability. This research will consider the correlation between the results from applying the metric to assessing the quality of the SRS and the expected results from the experts.

E. *Characteristics of a good SRS*

There are many characteristics of a good SRS, which may be identified in various standards relating to the characteristics of a good SRS, such as IEEE standard 830 - Recommended Practice for Software Requirements Specifications [19] or IEEE standard 1233 - Guide for Developing System Requirements Specifications [20], organizational best practices concerning the SRS, or expert knowledge and experience. In this research, the IEEE 830 [19] and IEEE 1233 [20] standards are used as a basis for considering the quality of the SRS. The standards describe the 8 common characteristics of good SRS: 1) correct 2) unambiguous 3) complete 4) consistent 5) ranked for importance and/or stability 6) verifiable 7) modifiable and 8) traceable.

This research focuses only on three characteristics: unambiguous, verifiable and modifiable, as the unambiguous and verifiable characteristics relate directly to the usage of natural language in specifying software requirements, and modifiable relates directly to the structure of the SRS document. In addition, an automated tool for assessing the quality of the SRS based on considering these 3 characteristics can be developed. Assessing document quality based on other characteristics requires analysis and human decisions, therefore, a tool development for automatic assessment of document quality cannot be easily done. The IEEE 830 standard [19] has defined the 3 characteristics as follows:

1) Unambiguous: A SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation.

2) Verifiable: A SRS is verifiable if, and only if, there exists some finite cost-effective process with which a person or machine can check that the software product meets the requirement. In general any ambiguous requirement is not verifiable.

3) Modifiable: A SRS is modifiable if, and only if, it has table of contents, index and explicit cross-referencing.

Moreover, each requirement should be expressed separately and appear in one place in the SRS.

Also, the IEEE 830 standard [19] provides recommendations on the SRS document structure, that is, identifying the topics of important content that are essential to include in the document, consisting of 1) table of contents 2) introduction; purpose, scope, definitions, acronyms, and abbreviations, references and overview 3) overall description; product perspective, product functions, user characteristics, constraints and assumptions and dependencies 4) specific requirements 5) appendixes and 6) index. However, software requirements have a tendency to expand. Therefore, the document format must be managed under specific requirements, to allow the involved persons to use the SRS more easily. This research has chosen to manage the document format by feature, which is the consideration of software requirements as groups of functions according to feature, consisting of: 1) external interface requirements; user interfaces, hardware interfaces, software interfaces and communications interfaces 2) system features; introduction/purpose of feature, stimulus/response sequence and associated functional requirements 3) performance requirements 4) design constraints 5) software system attributes and 6) other requirements, as generally, customers often specify their software requirements as groups of required functions. These topics of important content will be used as criteria for considering the quality of the SRS as well.

III. THE PROPOSED SRS QUALITY ASSESSMENT METHOD

This research has the purpose of presenting a method for objectively assess the quality of the SRS, by applying the process assessment model [15] as a framework for quality assessment. The measurement process model and research [21] referring to methods for assessing the quality of documents related to the software development process are used as an approach for presenting the methods for assessing the quality of the SRS. The overall picture of the necessary activities and tasks for document quality assessment are as shown in Fig. 1.

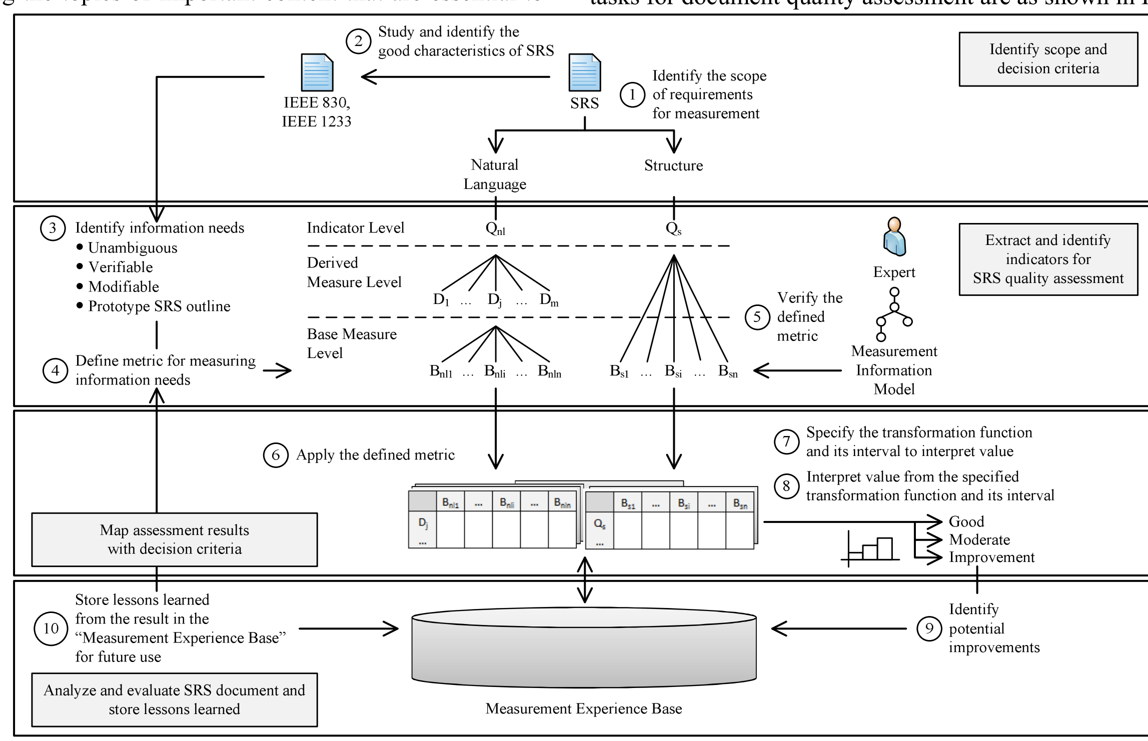

Fig. 1 The Detail of the Proposed Method

Fig. 1 shows that the overall picture of the presented method consists of 4 main steps, consisting of 10 sub-steps, with details as follows:

A. Identify Scope and Decision Criteria

This step specifies the purpose and scope of the measurement, which in this research is to specify the purpose and scope of the SRS quality assessment. The theories and knowledge related to the good characteristics of a SRS are also studied in order to select the candidate characteristic of a SRS for this research.

1) Identify the Scope of Requirements for Measurement:

As mentioned earlier, the quality of the SRS affects the quality of the software product to be delivered to the customer. Therefore, this step has specified quality assessment for the scope of the SRS in usage of natural language in specifying the requirements and document structure and structure as the purpose and scope for SRS document quality assessment, in order to obtain an information product that satisfies the need to know the quality level and defects in the document.

2) Study and Identify the Good Characteristics of SRS:

This step studies the good characteristics of the SRS. This research has studied the IEEE 830 [19] and IEEE 1233 standards [20], which are standards that mention the characteristics of a good SRS, defects that should be avoided, and topics of important and necessary content to be specified in the document, and applied the characteristics specified in both standards as criteria for considering the quality of the SRS, in use of natural language in specifying requirements and document structure, which is the purpose and scope of SRS quality assessment, as mentioned in III.A.1.

B. Extract and Identify indicators for SRS Quality Assessment

The characteristics requiring quality assessment are identified, from studying theory and knowledge relating to characteristics a good SRS, which must also correspond to the scope of the quality assessment specified in the first step. The necessary activities and tasks for quality assessment are also specified, by defining metrics for usage of natural language in specifying requirements and document structure. Also, past lessons collected in database can be used for defining the metric.

3) Identify Information Needs:

From studying the IEEE 830 [19] and the IEEE 1233 standards [20], which describe the characteristics of a good SRS, this research will consider only 3 characteristics of a good SRS, which are considered as entities relating to the SRS quality: 1) Unambiguous 2) Verifiable and 3) Modifiable, for the reasons mentioned in II.E., and also in order to correspond to the purpose and scope as mentioned in III.A.1. The three characteristics chosen for consideration have attributes, properties of entities for which examples are provided as follows:

- Unambiguous attribute: For example, specifying software requirements in the form of personal opinions or ambiguous words, such as the software requirement "Changing screens will require very little computation and thus will occur very quickly" [22] in which there is an ambiguous word "quickly", for which each person may interpret the speed in changing the screen differently, resulting in this specified software requirement being ambiguous.
- Verifiable attribute: For example, specifying software requirements by using adjectives or adverbs that cannot be properly defined, such as the software requirement "The application must utilize the PayPal API to provide the appropriate functionalities" [22] in which there is the adjective "appropriate", for which it cannot be defined what an appropriate functionality means. Therefore, this software requirement cannot be verified.
- Modifiable attribute: For example, specifying software requirements using punctuation marks, as the use of punctuation may result in the specification of those software requirements tends towards referencing more than one software requirement in a single sentence, and may also be complicated and hard to understand. If there is a subsequent need to change the software requirements, it is therefore not easy to modify, such as the software requirement "The only potential safety concern associated with this application applies to virtually all handset apps: SplitPay should not be used while operating a vehicle or in any other situation where the user's attention must be focused elsewhere" [22] in which punctuation appears, namely ":".

4) Define Metrics for Measuring Information Needs:

The metric defined by this research was specified through using the measurement information model, as explained in detail in II.C, as an approach for defining the metric, by considering the attributes, properties of the characteristics, as explained with the examples in III.B.3. The results obtained from using the metric to assess the quality of the SRS must satisfy the purpose of the SRS quality assessment as mentioned in III.A.1. Therefore, this research has defined a metric for assessing the quality of the SRS in the usage of natural language in specifying requirements, document structure, and overall quality of the document, with details as follows:

4.1) Defining the metric for assessing the SRS quality in usage of natural language in specifying requirements:

a.) Base measure level: Extracting the attributes used for calculations according to the 3 characteristics specified. Calculations can be made by counting the frequency of appearance for the attributes in each software requirements appearing in the SRS document. Before the frequencies may be counted, the words and parts of speech in the software requirements sentence must determined. In this research, the Stanford Parser tool [23, 24] is used for analysis of sentences, and identifying words and their parts of speech in the sentences. The relationships between the attributes affecting the quality of the SRS in the usage of natural language in specifying requirements and characteristics of a good SRS are shown in Table I.

However, counting the frequency of attributes in each software requirement may affect the correctness in interpreting the software requirements quality level. Therefore, the results must undergo normalization using equation (1):

$$B_{nl_i} = \frac{NLM_i}{L_j} \quad (1)$$

Where B_{nl_i} is the base measure result for the i^{th} attribute that has been normalized, NLM_i is the frequency of appearance of the i^{th} attribute, and L_j is the length of the j^{th} software requirement.

TABLE I. THE RELATIONSHIPS BETWEEN THE ATTRIBUTES AFFECTING THE QUALITY IN USAGE OF NATURAL LANGUAGE IN IDENTIFYING REQUIREMENTS AND CHARACTERISTICS OF A GOOD SRS

No.	Quality Attributes	Characteristics of good SRS		
		Unambiguous (C_1)	*Verifiable (C_2)*	*Modifiable (C_3)*
NLM_1	Ambiguous word	✓	✓	
NLM_2	Abbreviation	✓	✓	
NLM_3	Acronym	✓	✓	
NLM_4	Subjectivity	✓	✓	
NLM_5	Optional	✓	✓	
NLM_6	Modal verb	✓	✓	
NLM_7	More than one verbs	✓	✓	✓
NLM_8	Pronoun	✓	✓	
NLM_9	Demonstrative adjective	✓	✓	
NLM_{10}	Quantifier	✓	✓	
NLM_{11}	Coordinating conjunction	✓	✓	✓
NLM_{12}	Punctuation	✓	✓	✓

NLM abbreviated from Natural Language Metrics

b.) Derived measure level: Assessing the quality of each software requirement. Quality can be considered in 3 different aspects according to the 3 characteristics. Each characteristic can have its quality level indicated with many base measure metrics, and can be calculated using equation (2):

$$D_{j_{C_x}} = \frac{\sum_{i=1}^{n} w_i B_{nl_i}}{\sum_{i=1}^{n} w_i} \tag{2}$$

Where $D_{j_{C_x}}$ is the derived measure result of the j^{th} software requirement, when considering characteristic x, w_i is the weight of the i^{th} attribute, and B_{nl_i} is base measure result of the i^{th} attribute.

c.) Indicator level: Assessing the quality of the SRS in the usage of natural language in specifying requirements, for which quality can be considered in 3 aspects according to the 3 characteristics and can be calculated using equation (3), as each requirement may have different significance to the quality in usage of natural language in specifying requirements.

$$Q_{nl_{C_x}} = \frac{\sum_{j=1}^{m} w_j D_{j_{C_x}}}{\sum_{j=1}^{m} w_j} \tag{3}$$

Where $Q_{nl_{C_x}}$ is the result indicating the quality of the SRS in usage of natural language in specifying requirements when considering characteristic x, w_j is the weight of the j^{th} software requirements, and $D_{j_{C_x}}$ is derived measure result of the j^{th} software requirements when considering characteristic x.

As mentioned earlier, example definitions for the metrics for assessing the quality of the SRS in usage of natural language in specifying requirements can be shown when considering the unambiguous characteristic as in Table II.

TABLE II. MEASUREMENT CONSTRUCT FOR QUALITY OF USING NATURAL LANGUAGE TO SPECIFY SOFTWARE REQUIREMENTS IN SRS (UNAMBIGUOUS CHARACTERISTIC)

Information Need	Assess quality of SRS
Measurable Concept	Quality of using natural language to specify software requirements
Relevant Entities	Unambiguous
Attributes	As shown in Table I
Base Measures	As shown in equation (1)
Measurement Method	Applied equation (1)
Type of Measurement Method	Objective
Scale	Real numbers from zero to one
Type of Scale	Ratio
Unit of Measurement	The number of defect found/ length of software requirements
Derived Measure	As shown in equation (2)
Measurement Function	Applied equation (2)
Indicator	Equation (3)
Model	Applied equation (3)
Decision Criteria	If the result obtained from applying equation (3) is below acceptable criteria, the usage of natural language in specifying the software requirements must be considered in order to improve the quality of natural language usage.

4.2) Defining the metric for assessing the SRS quality in document structure:

a) Base measure level: Detecting the appearance of topics of important content in the SRS. In identifying the topics in the document, other words with the same meaning may be used. Therefore, synonyms of the topics must be defined for verification. In this research, the WordNet tool [25, 26], a large database collecting English language synonyms, and the Oxford Advanced Learner's Dictionary [27], have been used for adding groups of synonyms. The relationships between attributes affecting the quality of the SRS in document structure and characteristics of good SRS are shown in Table III, and the appearance of topics can be converted to quantitative data by using equation (4):

TABLE III. THE RELATIONSHIPS BETWEEN THE ATTRIBUTES AFFECTING THE QUALITY IN DOCUMENT STRUCTURE AND CHARACTERISTICS OF A GOOD SRS

No.	Quality Attributes	Characteristics of good SRS		
		Unambiguous (C_1)	*Verifiable (C_2)*	*Modifiable (C_3)*
SM_1	Presence of topic "Table of Contents"			✓
SM_2	Presence of topic "Purpose"		✓	
SM_3	Presence of topic "Scope"		✓	
SM_4	Presence of topic "Definitions, acronyms, and abbreviations"	✓	✓	
SM_5	Presence of topic "References"			✓
SM_6	Presence of topic "Overview"			✓
SM_7	Presence of topic "Product perspective"		✓	
SM_8	Presence of topic "Product functions"		✓	
SM_9	Presence of topic "User characteristics"			
SM_{10}	Presence of topic "Constraints"	✓	✓	
SM_{11}	Presence of topic			✓

	“Assumption and dependencies”			
SM_{12}	Presence of topic “User interfaces”			✓
SM_{13}	Presence of topic “Hardware interfaces”			✓
SM_{14}	Presence of topic “Software interfaces”	✓	✓	✓
SM_{15}	Presence of topic “Communication interfaces”		✓	
SM_{16}	Presence of topic “Functional requirement”		✓	
SM_{17}	Presence of topic “Performance requirements”	✓	✓	
SM_{18}	Presence of topic “Design constraints”			
SM_{19}	Presence of topic “Software system attributes”	✓		
SM_{20}	Presence of topic “Index”	✓	✓	✓

SM abbreviated from Structure Metrics

$$B_{s_i} = \begin{cases} 1; Found \\ 0; Not\ Found \end{cases} \quad (4)$$

Where B_{s_i} is the base measure result of the i^{th} attribute.

b.) Indicator level: Assessing the quality of the SRS in document structure. Quality can be considered in 3 different aspects according to the 3 focused characteristics. It can be calculated using equation (5), as each topic may have different significant to the document structure quality.

$$Q_{s_{C_x}} = \frac{\sum_{i=1}^{n} w_i B_{s_i}}{\sum_{i=1}^{n} w_i} \quad (5)$$

Where $Q_{s_{C_x}}$ is the result indicating the quality level of the SRS in document structure when considering characteristic x, w_i is the weight of the i^{th} attribute, and B_{s_i} is base measure result of the i^{th} attribute.

As mentioned earlier, example definitions for the metrics for assessing the quality of the SRS in document structure can be shown when considering the unambiguous characteristic as in Table IV.

TABLE IV. MEASUREMENT CONSTRUCT FOR QUALITY OF SRS DOCUMENT STRUCTURE (UNAMBIGUOUS CHARACTERISTIC)

Information Need	Assess quality of SRS
Measurable Concept	Quality of SRS document structure
Relevant Entities	Unambiguous
Attributes	As shown in Table III
Base Measures	As shown in equation (4)
Measurement Method	Applied equation (4)
Type of Measurement Method	Objective
Scale	Zero or one
Type of Scale	Ratio
Unit of Measurement	-
Indicator	As shown in equation (5)
Model	Applied equation (5)
Decision Criteria	If the result obtained from applying equation (5) is below acceptable criteria, the structure of the SRS document must be considered in order to improve the quality of document structure.

4.3) Defining the metric for overall quality assessment of the SRS:

a) Indicator level: An overall quality assessment of the SRS, for which the quality can be considered in 3 aspects according to the 3 characteristics. It can be calculated using equation (6), as the quality in usage of natural language in specifying requirements and in document structure may have different significance to the overall quality of the SRS:

$$Q_{C_x} = \frac{w_{nl} Q_{nl_{C_x}} + w_s Q_{s_{C_x}}}{w_{nl} + w_s} \quad (6)$$

Where Q_{C_x} $Q_{nl_{C_x}}$ and $Q_{S_{C_x}}$ are the results indicating the overall quality of the SRS, the quality level of usage of natural language in specifying requirements, and quality level of document structure respectively, when considering characteristic x; w_{nl} is the weight for the quality of usage of natural language in specifying requirements, and w_s is the weight for the quality of the document structure.

5) Verify the Defined Metrics:

The metrics defined in III.B.4 are verified for 3 properties: 1) metric is completely defined 2) metric is correctly defined and 3) metric is reasonably defined, in order to be able to efficiently use the defined metric in assessing the quality of SRS. The first two properties can be verified with the approach and meaning recommended by the measurement information model. For example, verifying that the metric has been completely defined at the base measure level is a verification of whether the attributes, measurement method, and base measure metrics have been defined, and verifying that the metric has been correctly defined at the base measure level is a verification of whether the scale type of the result has been correctly specified. For example, the result obtained from quality assessment at the base measure level is quantitative data that can be used in higher level processing, and thus must have its scale category specified as ratio scale, etc.

The last one is verification of the validity of the defined metric, by sending questionnaires relating to the factors affecting the quality of the SRS to graduate level students in the Software Engineering Program in the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, who are currently studying or have studied Software Requirements Engineering, for a total of 53 questionnaires. 35 questionnaires were returned, for a rate of 66.04%. Most respondents, 62.86%, have had 3-5 years of experience working in software development, with 54.76% working as programmers. Also, more than half of the respondents have had experience in identifying software requirements for SRS.

In verifying the validity of the defined metric, the Five-Point Bipolar Likert Scales [28] will be used. The respondents gave a score level based on their opinions about factors affecting the quality of the SRS: 12 factors in usage of natural language in specifying requirements, and 20 factors in document structure, ranging from strongly disagree (1) to strongly agree (5). The results for each factor were obtained, which can be considered according to the 3 characteristics as shown in Table VI. When the criteria for interpreting the data obtained from Likert’s metric [28] was used to interpret, it indicated that on average, respondents agreed with the presented factors, and that the presented factors strongly affect the quality of SRS, as almost every factor has an average opinion level higher than 3.41.

TABLE V. RESULTS OBTAINED FROM VERIFICATION OF DEFINED METRIC WITH EXPERTS, CATEGORISED BY THE 3 CHARACTERISTICS

Quality Aspects	Characteristics of good SRS		
	Unambiguous	***Verifiable***	***Modifiable***
Natural Language	3.80	3.74	3.55
Document Structure	3.91	4.20	3.97

C. Map Assessment Results with Decision Criteria

The defined metric is used in assessing the quality of the SRS, and specifying criteria for interpreting the result values obtained from assessing document quality, as well as linking the assessed results with the specified criteria, in order to obtain an information product that satisfies the purpose of assessing the quality of the SRS, in which past lessons collected in the data banks may be used as approaches for quality assessment and results analysis.

6) Apply the Defined Metrics:

The defined metric, as verified in III.B.5 is used to assess the quality of 4 actual industrial SRS, both in usage of natural language in specifying requirements, and in document structure: 1) The eNotebook system SRS document [29] 2) The WARC Tools system SRS document [30] 3) The Time Monitoring system SRS document [31] and 4) The SplitPay system SRS document [22], in order to show that the defined metric can be used to assess the quality of the SRS, and can also indicate the level of quality and apparent defects. The results obtained from assessing the quality of the SRS using the defined metrics in usage of natural language in specifying requirements and in document structure are shown in Tables VI and VII respectively.

TABLE VI. RESULTS OBTAINED FROM ASSESSING THE QUALITY OF SRS IN USAGE OF NATURAL LANGUAGE IN SPECIFYING REQUIREMENTS USING THE DEFINED METRIC

No.	Frequency of Defect Occurrences for each Metric												Total Defects (a)	Total Requirements	Total words (b)	Ratio (a/b)
	NLM_1	NLM_2	NLM_3	NLM_4	NLM_5	NLM_6	NLM_7	NLM_8	NLM_9	NLM_{10}	NLM_{11}	NLM_{12}				
1	16	1	17	0	0	13	181	17	11	68	183	22	529	114	1,680	0.31
2	29	12	91	0	0	17	217	22	12	78	240	16	734	89	1,827	0.40
3	10	0	5	0	5	23	130	9	5	66	149	7	409	73	1,177	0.35
4	22	1	12	0	4	40	184	18	22	85	149	30	567	95	1,423	0.40
Total Defects	77	14	125	0	9	93	712	66	50	297	721	75	2,239	371	6,107	0.37

NLM abbreviated from Natural Language Metrics

TABLE VII. RESULTS OBTAINED FROM ASSESSING THE QUALITY OF SRS IN DOCUMENT STRUCTURE USING THE DEFINED METRIC

No.	Frequency of Defect Occurrences for each Metric							Total Defects
	SM_1	SM_2	SM_3	…	SM_{18}	SM_{19}	SM_{20}	
1	✓	✓	✓	…	✓	✓	×	2
2	✓	×	✓	…	×	×	×	16
3	✓	✓	✓	…	×	×	×	4
4	✓	✓	✓	…	×	✓	×	4
Total Defects	0	1	0	…	3	2	4	26

SM abbreviated from Structure Metrics

From Table VI, it is observed that the SRS documents for the WARC Tools system and the SplitPay system have the highest ratio of defects detected by the metric per total words of the software requirements: 0.40, with most defects resulting from conjunctions. Therefore, there are tendencies to reference more than a single software requirement in one sentence, which affects subsequent verification and modification of software requirements.

From Table VII, it is observed that the SRS document for the WARC Tools systems has the most topics of important contents missing: 16 topics, and it has also been observed that the important content that is missing in all 4 documents is the Index part. This topic affects the quality of all characteristics, as it allows developers to easily use the SRS document.

The obtained results allow software engineers to know the cause of the defects appearing in the SRS, leading to the defects being corrected early in the software development process.

7) Specify the Transformation Function and Its Interval to interpret value:

The criteria for interpreting the results obtained from the defined metric for use in assessing document quality are specified, by having the involved persons specify 3 ranges of values: 1) good 2) moderate and 3) improvement. For example, good level requires a total lack of defects, moderate level has 1-2 defects, and improvement level has 3 defects upwards etc. This is because SRS for different systems will accept different levels of document quality based on the criticalness level of the system.

8) Interpret Value from the Specified Transformation Function and Its Interval:

The criteria for interpreting the results as defined in the previous section are used to interpret the results obtained from the SRS quality assessment in III.C.6. The interpreted results are qualitative information that can indicate the SRS quality, such as good, moderate, and improvement etc.

D. Analyze and Evaluate SRS Document and Store Lessons Learned

The results obtained from the quality assessment are analyzed for their strong points and weak points, and specify the portions of the SRS requiring improvement, allowing software engineers to know the weak points and defects that need to be corrected, as well as storing the obtained results and defined metric as knowledge captured in the database, in order to be used as an approach for future SRS quality assessment.

9) Identify Potential Improvements:

When the results obtained from the SRS quality assessment have been interpreted using the specified interpretation criteria, if the results from interpretation are in the improvement level, the cause can be identified based on the base measure results that indicate more defects out of the acceptable criteria, such as using more than 2 ambiguous words in a software requirements sentence, or more than 4 topics of important content being missing etc. The defects must be subsequently corrected, such as by reducing the ambiguous words in the software requirements sentence, or specifying important topics and content completely etc. The correction of the defects is to the discretion of the involved persons.

10) Store lessons learned from the result in the "Measurement Experience Base" for Future Use:

From the SRS quality assessment method as presented in the 9 previous steps, the following can be recorded: 1) the defined metric verified with software development expert 2) results indicating the SRS quality level, and 3) defects appearing in the SRS, as lessons in a database to be used as an approach for development and quality assessment of SRS in the future. The results obtained from quality assessment can indicate the strong points, weak points, and defects that appear in SRS, which software engineers can take care to prevent from appearing when developing SRS in the future. The defined metrics can also be used for SRS quality assessment in the future.

IV. Experiment and Result

This research tests the validity of the results obtained from the SRS quality assessment using the presented method, of whether the assessed results reflect the SRS quality, by comparing the assessed results with the results expected by experts. The details of the process of validity verification are as follows:

A. Metrics

The metric used by this research in indicating the validity of the results obtained from the SRS quality assessment is Pearson's correlation coefficient [32], which is a metric indicating the correlation or linear dependence between two sets of variables, which in this research are the results obtained from SRS quality assessment using the presented methods, and the results expected by experts.

B. Experiment Details

This research verifies the validity of the result obtained from SRS quality assessment by 13 experts, which are graduate students of in the Software Engineering course, Department of Software Engineering, Faculty of Engineering, Chulalongkorn University, assess the quality of the SRS in usage of natural language in specifying requirements and in document structure. The results from the assessment by experts are compared with the assessed results to determine if they correspond. Validity verification was performed on 9 actual industrial SRS, classified in 4 domains. Details are as shown in Table VIII, and the results obtained from validity verification, while the results expected by the experts in usage of natural language in specifying requirements and in document structure, are shown in Tables IX and X respectively. The experts that performed the SRS quality assessment must be experts with experience working in that domain or must have studied in subjects corresponding to that domain. For example, in the domain of security, experts must have studied the class of Computer Security, etc.

TABLE VIII. Details of the SRS used in the research

No.	Project Title	Domain	Total Requirements
1	CSC Multi-Utility [33]	Security	11
2	eNotbook [29]	Security	114
3	Vyasa [34]	Information Retrieval	29
4	WARC Tools [30]	Information Retrieval	89
5	Time Monitoring [31]	Resource Management	73
6	Libra Scheduler [35]	Resource Management	32
7	PeaZip [36]	Others	12
8	SplitPay [22]	Others	95
9	JHotDraw [37]	Others	30

TABLE IX. Pearson's correlation coefficient between the results obtained from SRS quality assessment using the presented method and the results expected by experts, in usage of natural language in specifying requirements

Natural Language Metrics	Domain				All Domain
	Security	*Information Retrieval*	*Resource Management*	*Others*	
NLM_1	0.41	0.23	0.49	0.56	0.43
NLM_2	0.51	0.58	-0.01	0.92	0.52
NLM_3	0.96	0.83	0.62	0.91	0.88
NLM_4	-	-	-	-	-
NLM_5	0.89	0.95	0.94	0.94	0.93
NLM_6	0.94	0.95	0.97	0.98	0.96
NLM_7	0.64	0.84	0.85	0.85	0.77
NLM_8	0.93	0.95	0.99	0.95	0.95
NLM_9	0.85	0.93	0.99	0.95	0.95
NLM_{10}	0.86	0.84	0.96	0.89	0.85
NLM_{11}	0.66	0.67	0.63	0.52	0.65
NLM_{12}	0.90	0.89	0.96	0.97	0.91

NLM abbreviated from Natural Language Metrics

TABLE X. Percent of consistency between the results obtained from the quality assessment of the SRS using the presented method and the results expected by experts, in document structure

Structure Metrics	Domain				All Domains
	Security	*Information Retrieval*	*Resource Management*	*Others*	
SM_1	100.00	100.00	100.00	100.00	100.00
SM_2	100.00	83.34	100.00	100.00	96.30
SM_3	100.00	83.34	100.00	100.00	96.30
SM_4	100.00	33.34	100.00	100.00	85.19
SM_5	100.00	50.00	100.00	91.67	86.11
SM_6	100.00	66.67	100.00	8.33	62.04
SM_7	100.00	100.00	100.00	100.00	100.00
SM_8	66.67	83.34	100.00	100.00	88.89
SM_9	100.00	100.00	100.00	100.00	100.00
SM_{10}	100.00	83.34	100.00	100.00	96.30
SM_{11}	50.00	100.00	100.00	91.67	86.11
SM_{12}	66.67	100.00	100.00	100.00	92.59
SM_{13}	100.00	100.00	100.00	100.00	100.00
SM_{14}	100.00	100.00	100.00	100.00	100.00
SM_{15}	83.34	100.00	100.00	100.00	96.30
SM_{16}	100.00	100.00	100.00	75.00	91.67
SM_{17}	100.00	83.34	100.00	83.33	90.74
SM_{18}	50.00	66.67	100.00	8.33	50.93
SM_{19}	100.00	100.00	50.00	91.67	86.11
SM_{20}	33.33	100.00	100.00	100.00	85.18

SM abbreviated from Structure Metrics

C. Analysis

From Table IX, when considering the Pearson's correlation coefficient for each domain (rightmost column), it is observed that all the values are positive, indicating that the results from the SRS quality assessment using the presented method and the results expected by the experts are correlated in the same direction. The 6th metric, the appearance of auxiliary verbs, received the highest Pearson's correlation coefficient, 0.96, as the appearance of auxiliary verbs in software requirements sentences make it unclear whether those software requirements will be implemented. The 1st metric, ambiguous words, received the lowest Pearson's correlation coefficient, 0.43, as the appearance of ambiguous words results in software requirements being able to be interpreted in more than one meaning. Also, each expert may have different opinions on the ambiguity of the words that appear. Therefore, allowing the involved persons to specify a group of ambiguous words relating to the domain of the SRS, in order to more inclusively detect the appearance of ambiguous words, is one method to alleviate this problem. It has also been observed that the Pearson's correlation coefficient for the 2nd metric, that is, the appearance of abbreviations, of the resource management domain, has a value of -0.01, meaning that the results from assessing the quality of documents using the presented method, and the results expected by experts correlate in opposite directions. This may be because the experts already know the meaning of those abbreviation, even if these abbreviations were not defined in the glossary, such as "i.e." However, this experiment cannot consider the correlation of the 4th metric, the specification of software requirements in the form of personal opinions, as no such attribute appears in any of the SRS, but if this attribute appears in a SRS, it can be detected using the presented document quality assessment method.

From Table X, the results obtained from SRS quality assessment is consistent with the results expected by most experts, as when considering the consistency percentage value for each domain (rightmost column) it is observed that all the values are higher than 50%. The 18th metric, the appearance of the Design constraints topics received the lowest consistency percentage value, 50.93%. Also, it has been observed that the consistency percentage value for the 20th metric, the appearance of the Index for the security domain, the 4th metric, the appearance of Definitions, acronyms, and abbreviations topic for the information retrieval domain, the 6th metric, the appearance of the Overview topic, and the 18th metric, the appearance of the Design constraints topic for other domains, have the values of 33.33, 33.34, 8.33, and 8.33 respectively, which are less than 50%. This may be because the synonyms of the specified topics may not be sufficient for detecting these topics. Therefore, specifying the synonyms of these topics and allowing the system administrators to manage the set of synonyms is one method that the problems can be solved. Also, another possible cause is human error from the experts, as there are many SRS, and much time is required for quality assessment: On average, it takes approximately 78 minutes per 1 SRS.

V. Threats to Validity

Important issues to be considered for SRS quality assessment are as follows:

1) *Completeness, correctness, and validity of the defined measurement:* As these metric will be used for assessing the quality of SRS, it directly affects the results indicating the quality of the assssed SRS. In this research, the defined metrics validity is mentioned in III.B.5.

2) *The correctness of the assessment of SRS quality in usage of natural language in specifying requirements:* When the frequency of the appearance of attributes is counted to be equal, but the software requirements sentences are not equal, when the results are immediately compared, the results obtained do not reflect real SRS quality. In this research, a method for solving this problem was proposed, by normalizing the assessed results to the same baseline.

3) *Accuracy in detecting the appearance of topics for important contents in the SRS:* As specifying the topics in this document may use other words with the same meaning, this research presents an approach to solving this problem by specifying a set of synonyms for those topics. However, the accuracy in detecting the appearance of the topics still depends on the specified set of synonyms. Therefore, the research allows the system administrator to manage the group of specified synonyms.

4) *Validity of the results obtained from SRS quality assessment using the presented method:* As the results of assessment must reflect the quality of the SRS, this research verifies the validity by comparing with the results expected by experts, as explained in section IV.

VI. Conclusion and Future Work

The quality of the SRS is an important factor that affects the development and verification of correctness for the software product to be delivered to the customers. If any defaults appear in the document, it can negatively affect the software development process. Therefore, if the quality level and defects appearing in the SRS can be determined, it leads to those defects being corrected to make the document more complete during the earliest steps of the software development process. This research presents a method for objectively assessing the quality of the SRS in usage of natural language in specifying requirements, in document structure, and in overall quality of the document, by applying the process assessment model as a framework for assessing the quality of the SRS, and using the measurement process model and measurement information model as an approach for presenting the SRS assessment method and defining the metric, respectively. In this research, only 3 characteristics of the SRS are considered: unambiguous, verifiable, and modifiable.

The results obtained from the presented SRS quality assessment method is the defined metric. The results indicating the quality level and the defects appearing in the SRS can be stored as lessons in a database, which can be used as an approach for future development and quality assessment of subsequent SRS.

Furthermore, this research is also aware of the important issues that may affect the accuracy of the SRS quality assessment, namely, the accuracy of the SRS quality assessment in usage of natural language in specifying requirements and in document structure, as well as the validity of the defined metric and the results obtained from assessing the quality of the SRS using the presented method. Validity has been verified with experts, and the results show that the defined metric significantly affects the quality of the SRS. Also, the results from assessing the quality of SRS with the presented method can reflect the quality of the documents.

The SRS quality assessment method presented in this research can be used as an approach for assessing the quality of other documents that occur in the various steps of the software development process in a similar fashion, such as software project management plans, software test documentation etc., by studying good characteristics of those documents, as well as the factors that affect the quality of the documents. Then defining a metric may established for quality assessment by directly considering the characteristics of the document, in order to determine the level of quality and the defects that appear, leading to correction of the defects to make the document more complete, as well as allowing the software project to be successful.

The researchers have a concept of applying the presented SRS quality assessment method for tool development that can automatically assess the quality of SRS.

REFERENCES

[1] Bob Ferguson and Guiseppe Lami, "Automated Natural Language Analysis of Requirements," ed, 2005.

[2] I. Hussain, O. Ormandjieva, and L. Kosseim, "Automatic Quality Assessment of SRS Text by Means of a Decision-Tree-Based Text Classifier," in Quality Software, 2007. QSIC '07. Seventh International Conference on, 2007, pp. 209-218.

[3] H. M. Jani, "Applying Case-Based Reasoning to software requirements specifications quality analysis system," in Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on, 2010, pp. 140-144.

[4] H. Mat Jani and A. B. M. Tariqul Islam, "A framework of software requirements quality analysis system using case-based reasoning and Neural Network," in Information Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference on New Trends in, 2012, pp. 152-157.

[5] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, et al., "Identifying and measuring quality in a software requirements specification," in Software Metrics Symposium, 1993. Proceedings., First International, 1993, pp. 141-152.

[6] R. S. Kenett, "Software specifications metrics: a quantitative approach to assess the quality of documents," in Electrical and Electronics Engineers in Israel, 1996., Nineteenth Convention of, 1996, pp. 166-169.

[7] W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt, "Automated analysis of requirement specifications," in Proceedings of the 19th international conference on Software engineering, 1997, pp. 161-171.

[8] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "An automatic quality evaluation for natural language requirements," in Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality REFSQ, 2001, pp. 4-5.

[9] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool," in Software Engineering Workshop, 2001. Proceedings. 26th Annual NASA Goddard, 2001, pp. 97-105.

[10] G. Lami, S. Gnesi, F. Fabbrini, M. Fusani, and G. Trentanni, "An automatic tool for the analysis of natural language requirements," Informe técnico, CNR Information Science and Technology Institute, Pisa, Italia, Setiembre, 2004.

[11] H. M. Jani, "Online Quality Analysis of the Requirements Specifications Phase of the Software Development Cycle," in South East Asian Association for Institutional Research (SEAAIR), Bangkok, Thailand, 2007, pp. 166-179.

[12] R. Saavedra, L. Ballejos, and M. Ale, "Quality properties evaluation for software requirements specifications: An exploratory analysis," in CIbSE 2013: 16th Ibero-American Conference on Software Engineering - Memorias del 16th Workshop de Ingenieria en Requisitos, WER 2013, 2013, pp. 6-19.

[13] G. Génova, J. Fuentes, J. Llorens, O. Hurtado, and V. Moreno, "A framework to measure and improve the quality of textual requirements," Requirements Engineering, vol. 18, pp. 25-41, 2013.

[14] N. Carlson and P. Laplante, "The NASA automated requirements measurement tool: a reconstruction," Innovations in Systems and Software Engineering, vol. 10, pp. 77-91, 2014.

[15] "Information technology—Process assessment—Part2: Performing an assessment," ISO/IEC 15504-2, pp. 1-28, 2004.

[16] "Systems and software engineering - Measurement process," ISO/IEC 15939, pp. 1-46, 2007.

[17] "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," IEEE Std 610, pp. 1-217, 1991.

[18] "IEEE Standard for a Software Quality Metrics Methodology," IEEE Std 1061-1998, pp. 1-26, 1998.

[19] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998, pp. 1-40, 1998.

[20] "IEEE Guide for Developing System Requirements Specifications," IEEE Std 1233, 1998 Edition, pp. 1-36, 1998.

[21] P. Thitisathienkul and N. Prompoon, "Quality assessment method for software development process document based on software document characteristics metric," in Digital Information Management (ICDIM), 2014 Ninth International Conference on, 2014, pp. 182-188.

[22] L. C. Rick Aasen, Blake Matson, Ed Carlisle, Eric Jeffers, Josh Ritchey, Nick Carson, Travis Green, Phuong Vo, "Software Requirements Specification for SplitPay," ed, 2011.

[23] M. P. Marcus, M. A. Marcin-kiewicz, and B. Santorini, "Building a large annotated corpus of English: the penn treebank," Computational Linguistics, vol. 19, pp. 313-330, 1993.

[24] S. N. Group. The Stanford Parser: A statistical parser. Available: http://nlp.stanford.edu/software/lex-parser.shtml#Citing

[25] G. A. Miller, "WordNet: a lexical database for English," Commun. ACM, vol. 38, pp. 39-41, 1995.

[26] P. University. WordNet A lexical database for English. Available: http://wordnet.princeton.edu/

[27] O. University, "Oxford Advanced Learner's Dictionary," Patrick Phillips, Ben Francis, Suzanne Webb, and Victoria Bull, Eds., eighth ed: Oxford University Press, 2010, pp. 1-1732.

[28] Likert Rensis, New Patterns of Management: McGraw-Hill, 1961.

[29] K. Visalli, "Collaborative eNotebook Research Testbed Software Requirements Specification," ed, 2005.

[30] M. Middleton, "Open Source WARC Tools Software Requirements Specification," ed, 2007.

[31] M. Robillard, "Time Monitoring Tool Software Requirements Specifications," ed, 2001.

[32] K. Vanichbuncha, Statistical Analysis: Statistics for Management and Research, seventh ed. Thailand: SPJ, 2003.

[33] M. Vohra, "Software Requirements Specification Document Design and Development of CSC based Multi-Utility System," ed, 2007.

[34] F. Eaker, "Software Requirements Specification vyasa," ed, 2006.

[35] N. A. Jahanzeb Sherwani, Nausheen Lotia, Zahra Hayat, "Libra: An Economy-Driven Cluster Scheduler Software Requirements Specification," ed, 2001.

[36] L. Athanasios-Alexandros, "Software Requirements Specification for PeaZip," ed, 2009.

[37] N. Pavlos, "Software Requirements Specification for JHotDraw," ed, 2011.