

# Quality improvement measures for Software Requirement Specification

Dr.R.Bulli Babu<sup>1</sup>, Mr.S.Jalaiah<sup>2</sup>, Mr.P.Naga Bhushanam<sup>3</sup>

*\*Professor, Department of CSE, St.Marys Group of Institutions Guntur.*

*\*\*Assistant Professor, Department of CSE, St.Marys Group of Institutions Guntur.*

*\*\*\*Assistant Professor, Department of CSE, St.Marys Group of Institutions Guntur.*

drbullibabur@gmail.com, jalaiahcse@gmail.com, bhushanam1985@gmail.com

**Abstract:** one of the crucial steps in the software development process is collecting user requirements and represent them in the form of Software Requirements Specifications document(SRS).This SRS can be supplied as input for the next phase Requirement analysis which involves all the tasks that are conducted to identify the needs of the stakeholders. The Software Requirements Specifications document (SRS) is used to verifying the correctness of the software product throughout the software development process. It has been observed that many software organizations cannot deliver the finalized product according to customer needs and their satisfaction levels. This is because of the miscommunication between software developers and users while collecting the requirements and representing them in an ambiguous natural language in the requirement specification which negatively impacts the quality of the software. This paper aims to avoid the above inconsistencies in the software requirement specification phase and provide quality improvement measures. The main objectives of this paper are to identify the steps to measure the SRS quality and help the software requirement engineer to improve the quality of the SRS.

**Keywords:** SRS, Quality Model, RSQ, RDQ, Quality Indicators

## I Introduction

Software requirement specification means the agreements between users and software developers, what is expected and what is not expected to do from the software, which is willing to produce. This permits rigorous requirements, assessment before design, so this can reduce later redesign matters of the software. This supposed to be giving a practicable estimated cost of the product, and for the schedules. To make productively organizations validations and verification plans, they

will produce their own software requirements. Regarding the software requirement specifications, while transferring new software to a new user, the developer has to give an idea about the software. Normally software requirements are written in natural language. Apparently, for that as to the software requirement specification, we have to note down information informal or semi-formal methods. Describe more sharply by giving clear information about the appropriately selected permit of notations in exact requirements and aspects of the software architecture. General rules in individuals notations should allow to unfolding exact requirements. This is concerning certain types of reliable software and meticulous crucial safety-critical [1]. Anyhow choice of notation is a constraint for the authors and reader of preferences and skill and for training.

An SRS is a document that describes all the externally observable behaviors and characteristics expected of a software system [2]. It is important to the developers as it allows them to save time on communication, minimize development efforts, give the customer feedback, eliminates task duplication, facilitates the transfer to new users or to new machines, and break problems into parts. Furthermore, it also serves as the main document to verify validation and testing processes. There is no standard way to write an SRS document. Nonetheless, a good SRS should contain all the information as suggested in the IEEE Recommended Practice for SRS [3].

## II Literature Review

A poor requirement cannot lead to excellent software because the quality of any product depends on the quality of SRS itself [4]. Moreover, not all software developers are being trained to properly document and verify the quality of requirements in the SRS. Quality

Properties of SRS is one that contributes to successful, cost-effective creation of software that solves real user needs [5] and some of the qualities are:

**i. Conciseness** An SRS is concise if, and only if, every requirement stated therein has only one interpretation [6]. This criterion possibly is the most difficult attribute to achieve using natural language. Though, the third-party inspector can determine the existence of conciseness by answering the predefined closed questions.

**ii. Understandability** The capability of each requirement to be fully understood by the stakeholders when they used it for developing software.

**iii. Completeness** An SRS is complete if, and only if, it includes the following elements [6]

a) All significant requirements, whether involving functionality, performance, design constraints, attributes or external interfaces.

b) Complete labels and references to all figures, tables, and diagrams in the SRS and definition of all terms and units of measure.

This paper mainly organized into five sections where Section I deals with the introduction part of the paper which presents the bag round details, Section II discusses Literature Review, Section III presents Research Methodology about the topic, Section IV discusses our observations regarding the study. Section V concludes this paper.

In order to write effective software requirement specification, we should incorporate the following characteristics which are useful for reflecting quality in SRS Document. The IEEE 830 standard [7] 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 a table of contents, index and explicit cross-referencing. In representing the software requirements specification, since requirements are gathered in natural language.

Ashfa Amber in [8] presented a methodology to overcome these NLPs issues and introduce a vocabulary named as SBVR (Software business vocabulary rules). In [9] umairah anuar et al. introduced an approach to overcome NLPs issue by defining boilerplates, that are helpful to covert natural language to formal language and other benefits of these boilerplates are their re-usability.

Few researchers also try to automate the SRS so it will generate automatically as Azeddine chikh proposed a dynamic model for SRS [10], this model replaces the manual process of adding requirement in SRS template, they proposed a software in which requirements are added, at the end it generates the SRS on the basis of given inputs, but it has few limitations.

Inspection is a very important step in the requirement phase to analyze the quality of requirements [11]. Frank salger in [12] implemented qualitative research in order to examine the effects which can be implemented by aspects like completeness, consistency, comprehensibility, testability, changeability, traceability. It less effectively addresses the following quality attributes, correctness, and feasibility. It insufficiently addresses the following quality attributes external completeness and conciseness. This research focuses on quality attributes effectiveness regarding the QG inspection technique since every technique has its own pros and cons, so in this research, their inspection methodology was QG-spec inspection technique.

Goal quality metrics (GQM) another method [13] presented by Eric Knauss. A survey that focuses on the quality of SRS by defining a GQM plan, Its goal depends upon two perspectives one is quality of requirements, which are formal requirement quality and content related requirement quality of SRS, on the basis of these goals author analyzed 40 different projects and drawn results.

### III Methodology

According to the Quality Evaluation of Software Requirements Specification [14] and Writing Effective Requirements Specification [9] the following two aspects reflect the quality measurement in software requirement specification document.

**i) Requirement Sentence Quality (RSQ):** the syntactical quality of single sentences considered separately;

**ii. Requirement Document Quality (RDQ):** the quality of the sentences considered in the context of the whole requirements documents.

From the papers, we finalized the quality attributes that are feasible to be automated by using the rules to determine the RSQ and RDQ that will further be discussed in the subsequent subsection.

**Quality Attributes** The definition of each of the Quality Properties has been discussed in the Introduction, while this section will briefly explain the definition of each of the Quality Attributes. The Quality Attributes are divided into attributes related to RSQ and RDQ. The following are Quality Attributes with its brief definition of each attribute.

### 1) RSQ related attributes

#### a) Implicit Sentences

A sentence is an implicit subject sentence if:

- its subject contains a demonstrative adjective;
- is expressed by means of pronouns;
- is specified by prepositions and is specifies by an adjective [14].

#### b) Optional Sentences

If a sentence contains an option then it is called an optional sentence [14].

#### c) Vague Sentences

A sentence is vague if it includes words holding inherent vagueness [14].

#### d) Weak Sentences

Category of clauses that will cause uncertainty and leave room for multiple interpretations [15].

#### e) Multiple Sentences

A sentence is said to be multiple if it contains more than one subject or more than one main verb [14].

#### f) Directives

Category of words and phrases that point to illustrative information within the requirement document. The data and information pointed to by directives will strengthen the quality of SRS [15].

### 2) RDQ related attributes

#### a) Readability Index

A category of attributes that measure how easily an adult can read and understand the requirements document [15].

Each of the above attributes had been associated with the Quality Properties under the Goal Properties as listed in Table 1.

Table 1  
List of Quality Attributes

Aspects	Quality Properties	Quality Attributes
RSQ	Conciseness	▪ Implicit Sentences
		▪ Optional Sentences
		▪ Vague Sentences
RDQ	Understandability	▪ Weak Sentences
		▪ Multiple Sentences
		▪ Directives Sentences
	Completeness	▪ Readability Index

In Table 1, the SRS Quality Properties that had been identified were listed and mapped based on the previously defined Quality Attributes. The Quality Attributes that are feasible to be automatically calculated are mapped to the respective Quality Properties [14]. For example, conciseness quality property can be measured from the perspectives of implicit, optional, vague and weak sentences.

### 3) Quality indicators

These quality indicators mainly focus on the requirement specification document and can be easily computed the information related to particular quality property. Several Indicators have been included in the Quality Model, each associated with a Property. Below several indicators are given and an association between them is also represented followed by the quality model and its properties are shown.

The Quality model mainly includes two aspects Requirement Sentences Quality (RSQ) and Requirement Document Quality (RDQ) under which different indicators are organized.

The Requirement Sentences Quality (RSQ) related Indicators are

- Implicit Subjects Sentences
- Multiple Sentences
- Optional Sentences
- Subjective Sentences
- Underspecified Sentences
- Vague Sentences
- Weak Sentences

The Requirements Document Quality (RDQ) related Indicators are

- Comment frequency,
- Readability Index
- Under referenced Sentences
- Unexplained Sentences

The description of each Quality Indicator with examples is presented below.

**Implicit Subject Sentences:**

An implicit subject sentence includes the following:

- consists of demonstrative adjectives like this, these, that, those
- represented by pronouns: it, they.
- represented a preposition like above, below, ...
- represented by an adjective like previous, next, following, last, first, ...

**Multiple Sentences:**

A Sentence is multiple if it:

- contain more than one subject or more than one main verb
- contain more than one direct or indirect compliment that specifies its subject

**Optional Sentences:**

If a sentence contains an option then it is said to be optional.

Note: If a sentence contains words like possibly, eventually, if appropriate or if needed then it is referred to as an optional sentence.

**Subjective Sentences:**

A sentence is subjective if it refers to personal opinion or feeling.

Note: sentences containing the following kinds of wording are to be pointed out as potentially subjective:

- having in mind, take (into) account, take into consideration, ...
- similar, better, similarly, worse, ...
- as [adjective] as possible.

**Underspecified Sentences:**

A sentence is under indicated if its subject contains a word distinguishing a class of items without a modifier determining an occurrence of such class.

**Vague Sentences:**

A sentence is vague if it includes words holding inherent vagueness. Note: Vague adjectives are:

- Intrinsic characteristics possessed adjectives like clear, well, easy, strong, weak, good, bad, efficient, low .....
- Environmental characteristics possessed adjectives like: useful, significant, adequate, fast, slow...
- Time characteristics possessed adjectives like old, new, future, recent, past, today’s, ...
- Location characteristics possessed adjectives like: near, far, close, back, in front,...

**Comment Frequency:**

The value of the CFI (Comment Frequency Index) is obtained from The Comment Frequency of a Requirements Document which can be specified as

$$CFI = NC / NR$$

where NC is the total number of Requirements having one or more comments, NR is the total number of Requirements of the document.

**Under referenced Sentences:**

A sentence of a Requirements Document (RD) is under referenced if it contains explicit references to:

- The sentences which are not numbered in the RD itself,
- The documents which are not referenced into the RD itself,

Note: The accompanying sorts of wording are to be considered as explicit references:

- according to
- on the basis of
- relatively to
- compliant with
- conformant to

**Readability Index:**

The unreadability index is the value of ARI (Automated Readability Index) [6] ( $ARI = WS + 9SW$  where WS is the average words per sentence, SW is the average letters per word)

**Unexplained Sentences:**

A sentence of a Requirements Specification Document (RSD) is unexplained on the off chance that it contains abbreviations not unequivocally and totally clarified inside the RSD itself.

The Quality indicators which are explained above have been related to a property and the quality model which can be shown in the following table.

**Table 2 Quality Model based on indicators**

Goal Properties	Properties	Quality Indicator
Requirement Sentences Quality	Non-Ambiguity	Implicit Subjects Sentences, Optional Phrases, Subjective Sentences, Vague Sentences, Weak Sentences
	Completeness	Underspecified Sentences
	Understandability	Multiple Sentences
Requirement Document Quality	Consistency	Underreferenced Sentences
	Understandability	Comment Frequency, Readability Index, Unexplained Sentences

## IV Discussion

From the above observations, we know that this examination is incomprehensive as there are Quality Attributes that are not achievable to be computerized. In such cases, human analysts are as yet required to make decisions. Furthermore, the practicality of the methodology is exposed to advance assessment by specialists. In any case, our exertion in directing this examination is, at any rate, can be valuable for necessities specialists to quantify their SRS quality. The usage can profit by estimating SRS quality for the applied ascribes or even to be utilized in pre-audit sessions.

## V Conclusion

This paper mainly focused on the necessity of using quality indicators in software requirements specification documents and also the use of a quality model to clearly understand user requirements collected by the developers before writing the software requirement specification document. Using quality indicators will indirectly help to improve quality in SRS Document. In conclusion, the representation of SRS without ambiguity will increase the success rate of finalized products which can possible by using quality indicators and Quality models.

## References :

- [1] Naik, S. and Tripathy, P. (2008) Software Testing and Quality Assurance: Theory and Practice. Wiley-Spektrum. <https://doi.org/10.1002/9780470382844>
- [2] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledebor, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and measuring quality in a software requirements specification," in Proceedings of the First International Software Metrics Symposium, 1993, pp. 141–152.
- [3] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std., 830-1998, pp. 1–40, Oct. 1998.
- [4] D. Rubinstein, "Standish Group report: There's less development chaos today," Software Development Times, vol. 1, 2007.
- [5] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledebor, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and measuring quality in a software requirements specification," in Proceedings of the First International Software Metrics Symposium, 1993, pp. 141–152.
- [6] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std., 830-1998, pp. 1–40, Oct. 1998.
- [7] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998, pp. 1-40, 1998.
- [8] J. Siddiqi, "Requirement engineering: The emerging wisdom," IEEE Softw., vol. 13, no. 2, pp. 15-19, Mar. 1996.
- [9] D. Rubinstein, "Standish Group report: There's less development chaos today," Software Development Times, vol. 1, 2007.
- [10] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledebor, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and measuring quality in a software requirements specification," in Proceedings of the First International Software Metrics Symposium, 1993, pp. 141–152.
- [11] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std., 830-1998, pp. 1–40, Oct. 1998.
- [12] A. van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specification. 1st edition. NJ: Wiley, 2009.
- [13] S. O. Mokhtar, R. Nordin, Z. A. Aziz and R. M. Rawi, "Issues and challenges of requirement review in the industry." Indian Journal of Science and Technology, vol. 10, no. 3, pp. 1-5, Jan. 2017.
- [14] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "Quality evaluation of software requirement specifications," in Proceedings of the Software and Internet Quality Week 2000 Conference, 2000, pp. 1–18.
- [15] Wilson, W. M., "Writing effective requirements specifications," CrossTalk: The Journal of Defense Software Engineering, pp. 16–19, 1999.