Framework for Measuring the Quality of Software Specification

E.Stephen, E.Mit
Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak (UNIMAS), 94300 Kota Samarahan, Sarawak.
ellystephen@yahoo.com

***Abstract*—This paper proposes a platform for measuring the quality of structure and functional requirement in software requirement specification (SRS). The SRS contains information needed to ensure the quality of the software. Measurement will be proposed based on four quality properties namely preciseness, consistency, completeness and correctness. The completeness properties will be used to measure the SRS which is based on IEEE 830 as a minimal standard. Meanwhile, the consistency, correctness and preciseness properties are proposed to be used for measuring the functional requirement in the document. The measurement of the overall quality of the SRS will be calculated based on all quality properties. The rules and formula for computing the SRS quality are embedded in proposed framework., which is a basis for platform for assessing the software quality.**

***Index Terms*—Formal Specification; Qualitative Measurement; Software Quality; Software Requirement Specification.**

I. INTRODUCTION

The requirement is the first stage of software development project and known that the successfulness of software development depending on the quality of the SRS. Survey on Requirement Engineering practice and its critical problem shown that hidden or incomplete requirement is the main cause of project failure [1].

The study on problem solution for software specification assessment has contributed toward the successfulness of the development. According to D.M. Fernandez *et. al.* (2015) [1], 45% of respondent agreed toward implementation of the standard guideline; whereas 44% of respondent agreed on the clear role and responsibilities that needs to be carried out through the development.

SRS consists of properties needed to develop the system. The collected requirement in natural language may cause ambiguity due to the difference interpretation by developer. There are various studies had been done to overcome the ambiguity of natural language [16, 19]. Due to the focus on the natural language analysis, quality of the requirement can be assessed by formalizing the requirement.

This paper proposed the quantitative measurement of the quality of heterogeneous SRS. The study is focused on four quality properties that can be assessed as early as requirements documentation stage, which were preciseness, correctness, consistency and completeness. The study is divided into two categories namely the structure of the document and the functional requirement. The completeness properties will be assessed based on document's structure and correctness, consistency and preciseness will be assessed based on the functional requirement.

II. RELATED WORK

Variety of domain in software development is one of the factors affect the software quality [24]. Each domain may have their own focus quality properties. Even if the focused quality properties are difference between each domain, a standard had been implemented to standardize the SRS [11, 21]. Research had been done in automotive industry show that this standard is not enough to show a complete structure for this domain [17]. Additional quality properties may have to be implemented to accommodate required domain. But according to A. Takoshima *et. al.* (2015) [17], the implemented standard may become a minimal requirement that every SRS should follow.

Commonly software quality is grouped as a non-functional requirement for a software project. A comprehensive study had been done between software quality model namely McCall model, Boehm model, Dromey model and ISO 9126 [6]. Improvements in the model increase the understanding of the quality to be assessed. A lot of difference approaches had been done to overcome the crisis regarding the software quality [7, 8, 9]. Conversion between the requirement phase to the design phase is crucial because the functionality must be precise, consistent and correct. The capability to trace the function in the design and validate it with the requirement specification must be done to ensure the consistency. That validation shows the degree of correctness of developed design. To ensure the level of satisfaction by the client, those functional requirements must be stated in precise without any vague details [3].

A summary of techniques had been done using a qualitative approach [2, 22]. Since the requirement specification is written in natural language, it had caused a blooming in research to overcome its ambiguity [2, 5, 15, 16, 17, 22]. The concern of the research is due to the conflict interpretation of the functional requirement between different levels of stakeholder. The processes of validation and verification are time consumption and need commitment from the client. Most of the studies focus on the consistency of the term used in the requirement phase and compared it with the later phase of development. By assessing the term, traceability between the requirement phases with another phase can be easily done. In this proposed research, the formalization of quantitative measurement in SRS helps in term of measuring the document by concentrating on the structural and functional requirement in the SRS.

Several studies were done on the software requirement structure [5, 12, 13, 14, 15] whereby the standard requirement for the structure should be met [11, 17, 21]. An ontology approach had been proposed by researchers to ensure the completeness properties of the structure [12, 13, 17]. All the implemented structures are based on the standard in IEEE 830 [22] as it inherits almost similar structure with IEEE 29148 with supported common good characteristic of SRS. Besides, IEEE 830 still used by present researcher [14, 16, 17]. Automated SRS generation also implemented the IEEE 830 structure. The completeness properties can be measured by using the minimal number of topic in IEEE 830 table of content.

In Requirement Boilerplate (RB) model, [19] additional information is required for functional requirement and the model include the non-functional requirement such as data type or even the invariant value [23]. Aside from that, the model can be used by a function to refer to another functional requirement. The idea of the RB is to minimize the ambiguity in natural language at level of defining the system functionality [2]. Any vague details or ambiguity may lead toward impreciseness of the functions. By enforcing a restriction on the usage of natural language in specifying the functional requirement, it may help to minimize the ambiguity problem [2, 19]. Hence, to ensure preciseness properties of the functional requirements.

According to [21], three of the quality properties namely consistency, correctness and completeness are common quality characteristic. Here, the ambiguity in the sentences will be related to preciseness quality. The proposed idea on measuring the structure and functional requirement is based on the targeted quality properties which will be discussed in Section III.

III. Propose Framework

As discussed in Section II, Section III will be a platform to propose framework based on required properties. Rules and formula are defined based on the definition of the quality and work in [16, 17].

The new propose framework for measuring quality of software requirement is as shown in Figure 1. Each of quality properties, rules and measurement will be proposed by using formal logic. The quality properties are defined by using formal specification to enable precise quality measurement for each component and is discussed in the following sub-section.

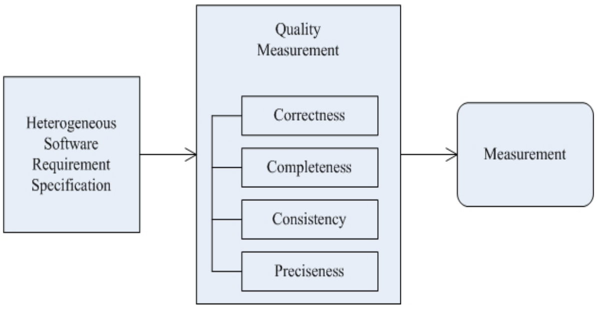


Figure 1: Propose Framework

A. *Heterogeneous SRS*

Heterogeneous is defined as various characters or content. The SRS will record in .doc or .docx format. As stated in Section I, the measurement is divided into two categories namely structure and functional requirement. The first step is to assess the structural document and then proceed to the functional requirement

B. *Quality Measurement*

The rules of quality measurement are as follows:

a. *Completeness*

In this research, this property is used to measure the structure of SRS.

The idea of measuring the structure of the SRS will ensure the completeness of the topic that will represent the whole document of the SRS.

IEEE 830 standard will become a guideline to ensure the completeness of the document. This is to make sure that all topics in different domain is counted. There are 23 numbers of topics that will become a constant for the measurement [21]. As there may have different number of topics in SRS, any new topic will be identified as an additional topic.

Once the topics were identified, it will be marked as found if it matches or synonym to topics. If not, it will be considered as additional topic.

A rule is created to formalize the assessment of the structure. In this research, a new completeness rules is proposed as:

$$(((T_n \wedge L_t) \equiv \text{Same}) \vee A_t) \equiv (T_n \equiv \text{Complete}) \quad (1)$$

T_n represent the topic being assessed, L_t represent a list of topics gathered from IEEE 830 table of content and A_t represent the added list of the topic from the tested table of content. From the rule, it clearly states that to ensure the structure is complete, the topic that currently assessed and the list of topics from the IEEE 830 table of content must be the same or the topic is from the added list then the assessed topic is considered complete.

b. *Correctness*

Correctness is defined as the degree of which software, documentation or other items meet specified requirements [10] or capability to meet the satisfactory needs [15]. The degree of satisfactory of the user can be measured by adopting the Likert Scale Analysis method. This method assigned each of the points with certain quantitative measurement and commonly used to measure the level of human satisfaction toward a subject.

First part of functional requirement measurement is to measure the degree of user satisfaction with the functional requirement. Three point Likert Scale Analysis is implemented as a point of measurement for each of the functional requirement. Table 2 shows that each level of satisfactory will be assigned with a certain degree of measurement.

Table 2
Degree of satisfactory

Satisfaction Level	Metric Assign
Agree	1
Undecided	0.5
Disagree	0

The metric assigned for each of satisfaction level is between 0 to 1. Main reason to assign the metric as in Table 2 is to ease the measurement of functional requirement. As there are only three level of satisfaction chosen, the minimal assign metric is 0 which for disagree, 0.5 for undecided and 1 for agree. Each of the function in the functional requirement is assigned with degree of 0, 0.5 and 1.

c. *Preciseness*

Preciseness is defined as software specification that provides the basis for analyzing the requirements, validating that they are the stakeholder's intentions, defining what the designers must build and verifying that they have done are correct [18]. It also said that the requirement documentation should not contain vague details [3].

A vast collection of function in functional in a single SRS may cause data imprecise. Inconsistent way and usage of high level of abstract to specify the function result from the use of natural language. It is difficult if not impossible to identify the data type from the requirement specification. Therefore, in this study, suggest the data type will be formulated based on intrinsic nature of the term, in conjunction with the word repository (e.g., WordNet). The design stage will be much easier if the whole data type of the functional requirement is fully stated. Possible and accurate sketch design for user interface can be minimize as the possible data type stated.

Furthermore, the usage of the natural language as a medium to specify the functional requirement may cause problem. A list of the vague words is gather for this research to identify the possibility of it being used in written the functional requirement [20]. There are 138 words had been identified as vague word and may promote toward ambiguity of sentences. The identification of vague word and data type are expected to increase the preciseness of functional requirement.

A rule had been proposed to measure the preciseness of each function in functional requirement. As for the proposed measurement for each function in functional requirement, the constant of 2 had been implemented. The constant of 2 are types of detail collected and vague word. Each detail type is denoted as constant of 1 if found and constant of 0 if not found. Calculation of preciseness for each function depends on the present of the data type and vague word. In this research, a new preciseness rules is proposed as:

$$(D_t \wedge \neg V_w) \equiv (F_n \equiv \text{Precise}) \qquad (2)$$

Based on the rule (2), D_t represent data type, V_w represent the vague word and F_n represent the function in the functional requirement that is assessed. From the rule (2), it is stated that the assessed functional requirement is precise if and only if the data type is presented and none of the vague word detected.

d. Consistency

Consistency is defined as the degree of uniformity, standardization and freedom from contradiction among the documents or parts of a system or component [10]. In a simple definition, a specification is considered consistent if it does not conflict to each other. Since the functional requirement is written in natural language, a tool such as Stanford Parser can be used to analyst the sentences.

Third part of functional requirement measurement is by assessing consistency properties. Stanford Parser tool can be used to analyze the sentences by breaking down into a chunk of noun and verb phrase respectively. This process called text chunking which allows a better understanding on the sentences meaning. A better understanding of the sentences can be done by using the natural language processing tool.

Aside from analyzing the meaning of the sentences in functional requirement, the stakeholder of correspond function is also identified. Some of the issues had been identified such as conflict requirement caused by the redundancy. The issue of redundancy is the use of similar or synonym word such as the user may be called as client, customer or consumer. In order to solve this issue,a knowledge repository to store the possible synonym for the stakeholder is built. By combining technique of stakeholder and proper role function identification, the number of possible conflict requirement may be reduced which lead toward consistency.

In this research, a new consistency rules are proposed as:
Actor: Stakeholder

$$((H_n \wedge R_n) \wedge (H_{nl} \wedge R_{n+1})) \equiv (F_n \equiv \text{Consistent}) \qquad (3)$$

Actor: System

$$((Q_1 \wedge R_n) \wedge (Q_1 \wedge R_{n+1})) \equiv (F_n \equiv \text{Consistent}) \qquad (4)$$

Based on the rules (3) and (4), H_n represent the stakeholder in the function that is being assessed, R_n represent the role of the function being assess, H_{nl} represent other function with the similar stakeholder or not, R_{n+1} represent other roles of the function, Q_1 represent the system and F_n represent the function being assessed now.

From the logical rule, there are two rules which defined for the stakeholder and the system. For the stakeholder (3), even though the role in other function of same stakeholder is not the same during assessment, it considers consistent. It also applied the same rule for the system but it only represents itself comparable to the various type of stakeholders (4).

C. Measurement

Measurement for each quality properties is based on the proposed rules as well as the overall quality of SRS as follow:

a. Completeness

The proposed measurement of SRS structure (5), where S represent degree of completeness of structure, M_t represent the total number of matched topic, A_t represent added topic by tested SRS table of content and C_1 represent constant of 23. The constant of 23 represents 23 numbers of topics in the structure in IEEE 830. So, it can be said that minimal number for each structure in SRS is 23 and the standard shall be followed by any SRS.

$$S = \left(\frac{\sum M_t + A_t}{C_1 + \sum A_t}\right) * 100\% \qquad (5)$$

b. Correctness

The proposed measurement of correctness of functional requirement (6), U represents the degree of correctness of all function in functional requirement, P_t represent the total of mean point and F_t represent a total number of functions in functional requirement. The sum of the assigned satisfactory of each function in functional requirement is collected and divided by total number of function in functional requirement to gain a mean value.

$$U = \left(\frac{\sum P_t}{\sum F_t}\right) * 100\% \qquad (6)$$

c. Preciseness

The proposed measurement of preciseness of functional requirement (7), P represent the preciseness of all functional requirement, D_t represent the existent of data type, V_w represent the existent of vague value, C_2 represent constant of 2 and F_t represent total number of function in functional requirement. For each of the assessed function, the value for each of the function is based on the presentable of the data type and vague word. The sum of all function will be divided with the sum of total function plus the sum of detected vague word to gather the mean value.

$$P = \left(\frac{\sum \frac{D_t + V_w}{C_2}}{\sum F_t} \right) * 100\% \tag{7}$$

d. Consistency

The proposed measurement of consistency of functional requirement (8), *T* represent consistency, B_t represent the total number of consistent function and F_t represent the total number of functional requirement available. The mean is calculated to measure the degree of consistency of the functional requirement. To measure the mean, the total number of functions that are consistent divided with the total number of functions in functional requirement.

$$T = \left(\frac{\sum B_t}{\sum F_t} \right) * 100\% \tag{8}$$

e. Overall Quality

Overall quality of SRS is measured based on the result from structural and functional requirement. The proposed measurement to measure the SRS is presented in (9). For the overall quality, *S* represent the overall consistency, *U* represent the overall correctness, *P* represent the overall preciseness and *T* represent the overall consistency.

$$Overall\ Quality = \frac{S+U+P+T}{4} \tag{9}$$

IV. Framework Evaluation

In the previous section, the rule and equation to measure the structure and functional requirement are proposed and will be used to justified the proposed rule and equation based on the proposed framework.

For the proposed rule of each quality properties, condition statements are specified.

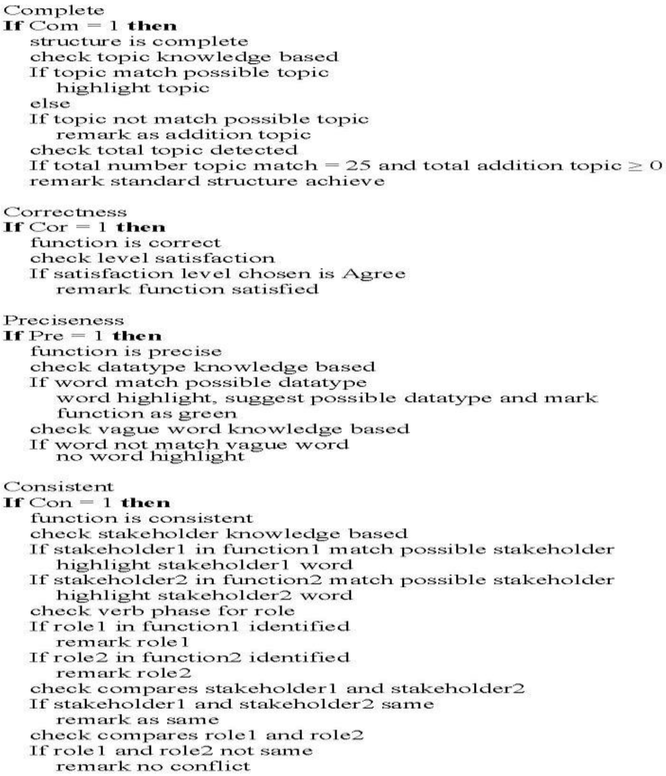
Figure 2 show the conditional statements for each of the quality properties. The condition for each of the statements is when all the rules are followed.

To further justify the proposed framework, a simple case study is used by taking a SRS as a case study. The structure and functional requirement are extracted from the tested SRS.

An example of the structure to be assessed is presented in Table 3.

To determine complete topics, the proposed rules and equations will be applied to Table 3. The first topic *Introduction* is identified by applying a similarity semantic technique to each topic in the knowledge repository; contains number of similar and synonym topic which include 23 numbers of topics from IEEE 830 table of content. Then, each topic is compared with topics within IEEE 830 table of content. As for unmatched topic, a similarity semantic technique is applied to ensure possible difference sentences from the same meaning with the topics in IEEE 830 table of content. If it unmatched with any topics in knowledge repository; it reconsidered as an additional topic.

The Table 3 identification result in 18 matched topics and 3 unmatched found. The proposed technique applied to each topic and in results, being able to find similar and synonym topics. Out of 18 matched topics, 3 topics found as synonym topics with the topics in IEEE 830 table of content. Those topics namely *Data acquisition module* is equal to *Logical database requirements, User interface* is equal to *External interfaces* and *Implementation priorities* is equal to *Design constraints*.



```
Complete
If Com = 1 then
   structure is complete
   check topic knowledge based
   If topic match possible topic
      highlight topic
   else
   If topic not match possible topic
      remark as addition topic
   check total topic detected
   If total number topic match = 25 and total addition topic ≥ 0
   remark standard structure achieve

Correctness
If Cor = 1 then
   function is correct
   check level satisfaction
   If satisfaction level chosen is Agree
      remark function satisfied

Preciseness
If Pre = 1 then
   function is precise
   check datatype knowledge based
   If word match possible datatype
      word highlight, suggest possible datatype and mark
      function as green
   check vague word knowledge based
   If word not match vague word
      no word highlight

Consistent
If Con = 1 then
   function is consistent
   check stakeholder knowledge based
   If stakeholder1 in function1 match possible stakeholder
      highlight stakeholder1 word
   If stakeholder2 in function2 match possible stakeholder
      highlight stakeholder2 word
   check verb phase for role
   If role1 in function1 identified
      remark role1
   If role2 in function2 identified
      remark role2
   check compares stakeholder1 and stakeholder2
   If stakeholder1 and stakeholder2 same
      remark as same
   check compares role1 and role2
   If role1 and role2 not same
      remark no conflict
```

Figure 2: Conditional Statement

Table 3
Sample of table of content

1.	Introduction
1.1.	Purpose
1.2.	Scope
1.3.	Definition, acronym and abreactions
1.4.	References
1.5.	Overview
2.	Overall Description
2.1.	Product perspective
2.2.	Product functions
2.3.	User characteristics
2.4.	Constraints
2.5.	Assumptions and dependencies
3.	Specific Requirements
3.1.	Functional Requirements
3.1.1.	Data acquisition module
3.1.2.	Data processing module
3.1.3.	User interface
3.2.	Performance requirements
3.3.	Security requirements
3.4.	Implementation restrictions
3.5.	Implementation priorities

For the measurement, the proposed equations for the completeness properties applied.

$$\text{S} = \left(\frac{18+3}{23+3} \right) * 100\%$$
$$\text{S} = 81\% \tag{10}$$

The outcome of (10) shows unfulfilled numbers of topics from IEEE 830 table of content. Result shows 5 numbers of topics are short from 23 numbers of standard topics from

IEEE 830 table of content. It caused the structure incomplete due to unfulfilled minimal standard as mentioned in IEEE 830 numbers.

For the functional requirement, properties of correctness, preciseness and consistency applied. Rules and equations are proposed. To justify its significant to the proposed framework, sample of functional requirement is extracted from the similar case study.

Table 4
List of function

No.	Function
[F1]	The user should be able to calculate the percentage of the car movement.
[F2]	The admin should be able to adjust the car movement.
[F3]	The user should be able to view almost all of the car movement data

To ensure correctness of function, rules and equations will be applied. Each sample function in Table 4 is assigned with level of satisfaction. The measurement will be based on the level of satisfaction as presented in Table 5.

Table 5
Function satisfaction level

Function	Satisfaction Level		
	Agree	Undecided	Disagree
[F1]	✓		
[F2]	✓		
[F3]		✓	

As shown in Table 5, measurement will be calculated as each satisfaction level for each function chosen and based on the assigned metric for the satisfaction level namely *[F1]* is equal to 1, *[F2]* is equal to 1 and *[F3]* is equal to 0.5. The measurement for proposed equation of correctness properties applied.

$$U = \left(\frac{1+1+0.5}{3}\right) * 100\%$$
$$U = 83\% \qquad (11)$$

The result of (11), shown the percentage of correctness based on the chosen satisfaction level . As for the preciseness property, the proposed rules and equation for it are applied to the sample in Table 4. The sentences in functions are chunk. Each word in sentences is compared to data type knowledge based. The possible word that contribute toward identification of the data type will be highlighted and remarked as found. The next stage is to identify the possible vague word in the function. If it is found it will be remarked as found and highlighted. The same example for the functional requirement, applied toward proposed rules and equation and resulting the identification of possible data type and vague word as tabulated in Table 6.

Table 6
Result identified vague word and data type

Function	Vague Word	Data type	
		Identify Word	Possible Data Type
[F1]	-	percentage	double, float, integer
[F2]	-	adjust	double, float, integer
[F3]	almost	view	varchar, string

As shown in Table 6, vague word is identified in *[F3]* and word *almost* is ambiguous to be used in sentences. The usage may impact the sentence preciseness. For the identification of data type, the possible words that are used to represent the data type is listed in Table 6 and each identified words are listed with the suggested possible data types. For the measurement of preciseness property of functional requirement, the proposed equation applied.

$$P = \left(\frac{\frac{1+1}{2}+\frac{1+1}{2}+\frac{1+0}{2}}{3}\right) * 100\%$$
$$P = 83\% \qquad (12)$$

The result of measurement (12), show *[F3]* is not precise as vague word exist in *[F3]*. Each of possible vague word found will be highlighted. The proposed rules and equation also applied for consistency properties to sample of function (Table 4). By using the same example, sentences in the function undergo chunking process and analyzed to identify the role of each function. The semantic similarity technique is used to identify the possible stakeholder will be highlighted and remark as found.

Table 7
Result identified role and stakeholder

Function	Role	Stakeholder
[F1]	calculate percentage car movement	User
[F2]	adjust the car movement	Admin
[F3]	view almost all car movement data	User

In Table 7, each possible stakeholder and role of the functions is identified. Results shown consistency in term of the role where there are no roles conflict in each function even though *[F1]* and *[F3]* share the same stakeholder. The proposed equation is used to measure the consistency property of the functional requirement the proposed equation applied.

$$T = \left(\frac{3}{3}\right) * 100\%$$
$$T = 100\% \qquad (13)$$

Result from the measurement (13) shows no conflict in term of the role of the stakeholder even though there is function sharing the same stakeholder.

Overall, SRS qualities can be calculated as the measurement to measure the structure and functional requirement. The proposed equation for the overall quality applied.

$$\text{Overall Quality} = \frac{81+83+83+100}{4}$$
$$\text{Overall Quality} = 87\% \qquad (14)$$

Result from the measurement (14), show the SRS’s overall quality. The percentage of the measurement can be increase as the standard structure of IEEE 830 table ofcontent. Aside from that, function *[F3]* should be address with the stakeholder’s member before proceeding to next development phase.

V. Conclusion

In this research, rules and measurements are proposed to assess the structural and the functional requirement in SRS. The proposed framework shows the data flow and how the structure and functional requirement are measured.

There are four quality properties been assessed:

completeness, consistency, correctness and preciseness. The SRS's structure is used to assess the completeness properties. As for the functional requirement, it is used to assess the consistency, correctness and preciseness properties.

Many rules had been proposed for each quality and each rule will be a base for proposed measurement of corresponding quality.

The research's idea is to come out with a quantitative measurement by converting the qualitative data in order to measure the degree of SRS quality.

ACKNOWLEDGMENT

The authors would like to thank Universiti Malaysia Sarawak for providing the funding to conduct this research. This work was supported by Fundamental Research Grant Scheme FRGS/2/2014/ICT07/UNIMAS/02/1

REFERENCES

[1] D. M. Fernandez, "Naming the Pain in Requirements Engineering: A Design for a Global Family of Surveys and First Results from Germany," Information and Software Engineering, 2015, pp. 616-643.

[2] C. Arora, "Automatic Checking of Conformance to Requirement Boilerplates via Text Chunking: An Industrial Case Study," ACM / IEEE International Symposium on Emperical Software Engineering and Measurement, 2013, pp. 35-44.

[3] M. J. Ali, "Metrics for Requirements Engineering," 2006.

[4] "IEEE Std 610.12-1990," IEEE Standard Glossary of Software Engineering Terminology, 1990.

[5] A. Chikh, "Towards a Dynamic Software Requirement Specification, World Congress on Computer Applications and Information System," 2014, pp. 1-7.

[6] B. Singh, "A Review on Software Quality Models, International Conference on Communication System and Network Technologies," 2013, pp. 801-806.

[7] V. Basili, "The Experience Factory and Its Relationship to Other Quality Approaches, Advances in Computers," vol. 41, 1995, pp. 65-82.

[8] T. Pyzdek, "The Six Sigma Handbook: The Complete Guide for Greenbelts, Blackbelts, and Managers at All Levels," McGraw-Hill, 2003.

[9] P. Jalote, "CMM in Practice: Processes for Executing Software Projects at Infosys," Addison Wesley Longman, 2000.

[10] ISO/IEC/IEEE 24765, "System and Software Engineering-Vocabulary," First Edition, 2010.

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

[12] T. Avdeenko, "The Ontology-Based Approach to Support the Completeness and Consistency of the Requirements Specification," Control and Communications (SIBCON), 2015, pp. 1-4.

[13] N. Mahmud, "ReSA: An Ontology-based Requirement Specification Language Tailored to Automotive Systems," 10th IEEE International Symposium on Industrial Embedded System, 2015, pp. 1-10.

[14] M. G. Georgiades, "Automatic Generation of a Software Requirements Specification (SRS) Document," 10th International Conference on Intelligent System Design and Application, 2010, pp. 1095-1100.

[15] E. Sarmiento, "Using Correctness, Consistency, and Completeness Patterns for Automated Scenarios Verification," IEEE Fifth International Workshop on Requirement Patterns (RePa), 2015, pp. 47-54.

[16] P. Thitisathienkul, "Quality Assessment Method for Software Requirements Specifications based on Document Characteristics and its Structure," Second International Conference on Trustworthy Systems and Their Applications, 2015, pp. 51-60.

[17] A. Takoshima, "Assessing the Quality of Software Requirements Specifications for Automotive Software Systems," Asia-Pasific Software Engineering Conference (APSEC), 2015, pp. 393-400.

[18] P. A. Laplante, "What Every Engineer Should Know about Software Engineering," CRC Press, 2007.

[19] U. Anuar, "A Simplified Systematic Literature Review: Improving Software Requirements Specification Quality with Boilerplates," 9th Malaysian Software Engineering Conference (MySEC), 2015, pp. 99-105.

[20] Vague Words List, access on 1th January 2017, http://www.evilauthor.com

[21] IEEE Std. 830-1998, "IEEE Recommended Practice for Software Requirements Specifications," Software Engineering Standards Committee, 1998, pp. 1-26.

[22] M. Kamalrudin, "A Review on Software Requirements Validation and Consistency Management," International Journal of Software Engineering and Its Application, 2015, pp. 39-58.

[23] I. Omoronyia, "Guided Natural Language and Requirement Boilerplate," access on 1th January 2017, www.idi.ntnu.no/emner/tdt4242/foiler/2-3-Boilerplates.ppt.

[24] Q. Zu, "Human Centered Computing," Second International Conference, Revised Selected Papers, 2016, pp. 753-754.