

Activity Based Quality Assessment Technique for Software Requirement Specification

M.R Raja Ramesh, Ch.Satyananda Reddy

Abstract : Most software problems will arise based on the deficiencies that occur in software requirements specification development. It is essential to develop a quality requirement specification development, to achieve success for developing any software product because this software requirement specification information will be used entirely in all project development stages. The current work is focused on assessment of software requirement specification based on the activities of software product. The proposed work is evaluated with an example and is compared with other group namely case based reasoning and machine learning algorithms of quality assessment techniques. The results had shown the effectiveness of the proposed technique.

Key words: Quality Models, Software Requirements Specification Development, Quality Requirement Specification Development, Activity Based Assessment.

I. INTRODUCTION

In software requirement specification, quality requirement models are part of the functional requirement and non-functional requirements of a system, in non-functional requirements which are important for making a product more attractive and it is usable and it also specifies properties of system which is not primary functionality. Although quality requirements are an important factor in the success of a system, but it can be neglected in the software requirements engineering process. The problem is focused on quality assessment, it is a complex and flexible idea. The first and primary requirement of software specification often struggle with specifying software quality requirement specification on a level of abstraction that is suitable for the later phases. The high level abstraction of software requirement specification as “The software system shall be maintainable”.

The second requirement of software specification is elicitation phase, it process the demands for implicit domain knowledge. One often needs to consider a variety of domain-specific standards and guidelines, whereas it often remains unclear whether, to which extent, and how the contents have to be transformed into useful and appropriate software requirements [1]. In this paper, we evaluate activity based assessment to get quality software requirement specification document that can be inferred from domain specific standards and guidelines. We examine a comparative case study at Siemens the existing specification of a Marketing system which can be compare it to a new

software requirements specification produced using the proposed approach. We measure our approach supports the software requirements specification of traceable and measurable requirements in direct comparison to the approach.

Requirements Engineering (RE) is a major part of software engineering which discovers the purpose by identifying stakeholders and their needs, and documenting them for their future analysis and implementation [2]. In general, software quality attributes for a software requirement specification are part of a software quality model which can be used to assess the quality of a requirements document contained in software requirement specification [3]. Thus, each attribute might be related to the entire software requirement specification or to each requirement defined in it which is due to many attributes attained by the software requirement specification that can be achieved for each requirement defined in it. Related to this, many of the authors proposing the same property coincide in their considerations, some issues regarding particularities in the analysis deserve to be pointed out. The various quality properties that used to achieve the quality of the SRS is and its influence in shown in the Fig 1.

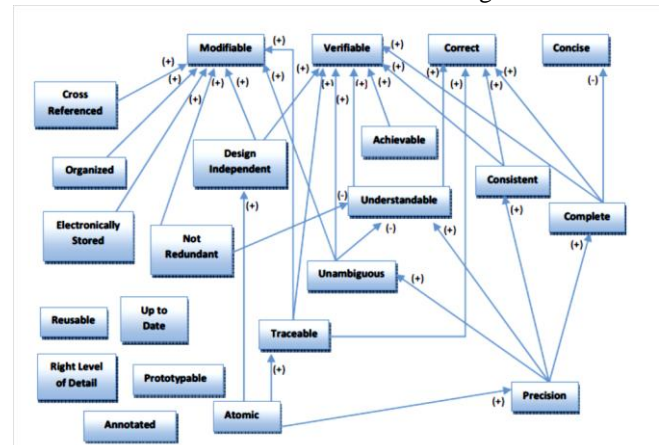


Fig 1: Influence between quality properties [4].

In this paper focused on the software quality models of software requirement specification development based on the various activities that carried out for development of the software product. The paper is structured as follows, section 2 organised related works and section 3 demonstrate the proposed research work. Section 4 evaluates the results and discussion and finally the paper ends with section 5 conclusion and presented future work of this paper.

Revised Manuscript Received on December 08, 2018.

M.R Raja Ramesh, Department of Computer Science and Engineering, Sri Vasavi Engineering College, Pedatadepalli, Tadepalligudem, Andhra Pradesh, India

Prof. Ch.Satyananda Reddy, Department of Computer Science and Systems Engineering, A.U College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India.



II. RELATED WORK:

In this paper focused on the software quality models of software requirement specification development activities based on the various activities, in which software requirements specification document was often gathered and specified customer's needs by using natural language processing have been proposed by many researchers can be classified into two groups: Machine learning is applied to assess the quality of software requirement specification development such as classification, case-based reasoning and neural network in the first group [6-8]. There are some limitations of the proposed research work such as i) The quality of software requirement specification development based on machine learning's algorithm is used to indicate

efficiency and ii) Training data set effectiveness. The defined metric is applied to assess quality of software requirement specification development in the second group [9-14]. Characteristics of good software requirement specification development are extracted and measured directly [15-17]. A problem remains because some certain metrics still required human decision[18-20]. The quality of software requirement specification development document using natural language to specify requirements which were focused by many researchers are mainly not in real [5]. The various documents that occur in the software development life cycle in shown in the Fig 2.

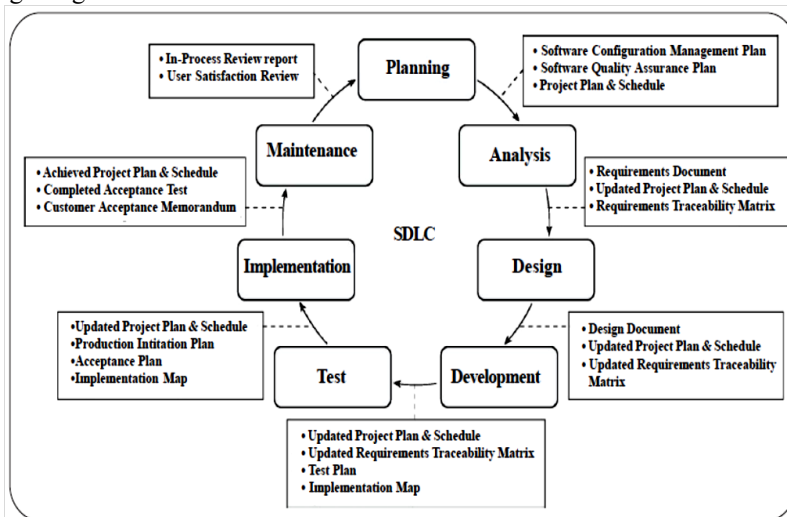


Fig 2: Deliverable Documents Occurred During SDLC Process[5]

III. PROPOSED SYSTEM:

In the proposed system the assessment of software requirements specifications is done by the activity based technique. Here the activities are formulated based upon a software product development as Software Requirement Specification, Quality model, Quality levels development measurement and assessment. Here the process starts with preparing activity in perspective of both internal and external software requirement specifications. This should specify by the system analyst who can specify the system requirements that can be clearly written in natural language based on that needs. system analyst will identify the quality requirements.

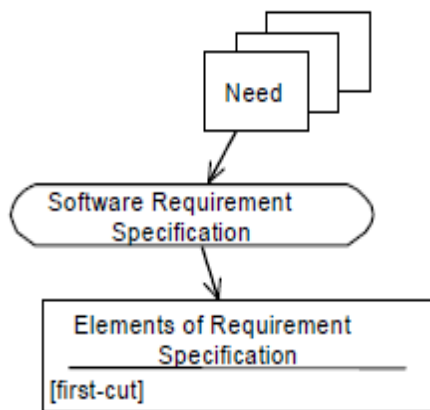


Fig 3: preparing activity

- Activity 1 produces the output as SRS
- Activity 2 produces the output as elements of quality model.
- Activity 3 here the assessment of the elements of quality model was obtained
- Activity 4 Here artifacts are measured and metric values are presented.
- Activity 5 produces the quality values for all levels.

3.1 Example

The Example in this paper deals with situation at the market field. The managers preparing the root map for sales person. Generally root map is prepared manually from the data given by the region managers. Here the problem is to find the feasible root map for the given data. The root map shows the different places for the particular region. The corresponding salesman specifies the place in the region. This enables easy to track the sales person and sales orders by the sales managers. Here the manager needs to develop software that supports the root map that expected.

The following reasons may leads to get incorrect root map that can be specified by manager.



Activity Based Quality Assessment Technique for Software Requirement Specification

Input data is incomplete,

It is not access the database directly,

Different constraints received from teachers,

Congestion the sales person

To resolve problem the following corrections should be made:

Data that should be accessible from anywhere at any time

Constraints can be recorded based on the dead line.

Preview of root map can be taken.

Model 1: SRS Development

Input: Customer requirements

Output: Software requirements

Some of the requirements are as follows:

R1) Needs the sales man details and place details.

R2) needs the vacancy places that are required to assign

the sales man.

R3) needs the specific details to assign required numbers of sales man and sales managers.

R4) Developed software will be implemented in Microsoft Platform.

R5) this system should be accessible from anywhere at any time.

R6) the presentation, reports, tables should be accessible from anywhere at any time.

Software Product (SP) will consist of code and Documentation with conjunction of MS Access and Microsoft excel.

Model 2: Software Excellence

Input: Software requirement specification development

Productivity: Attributes of software quality model described table 1.

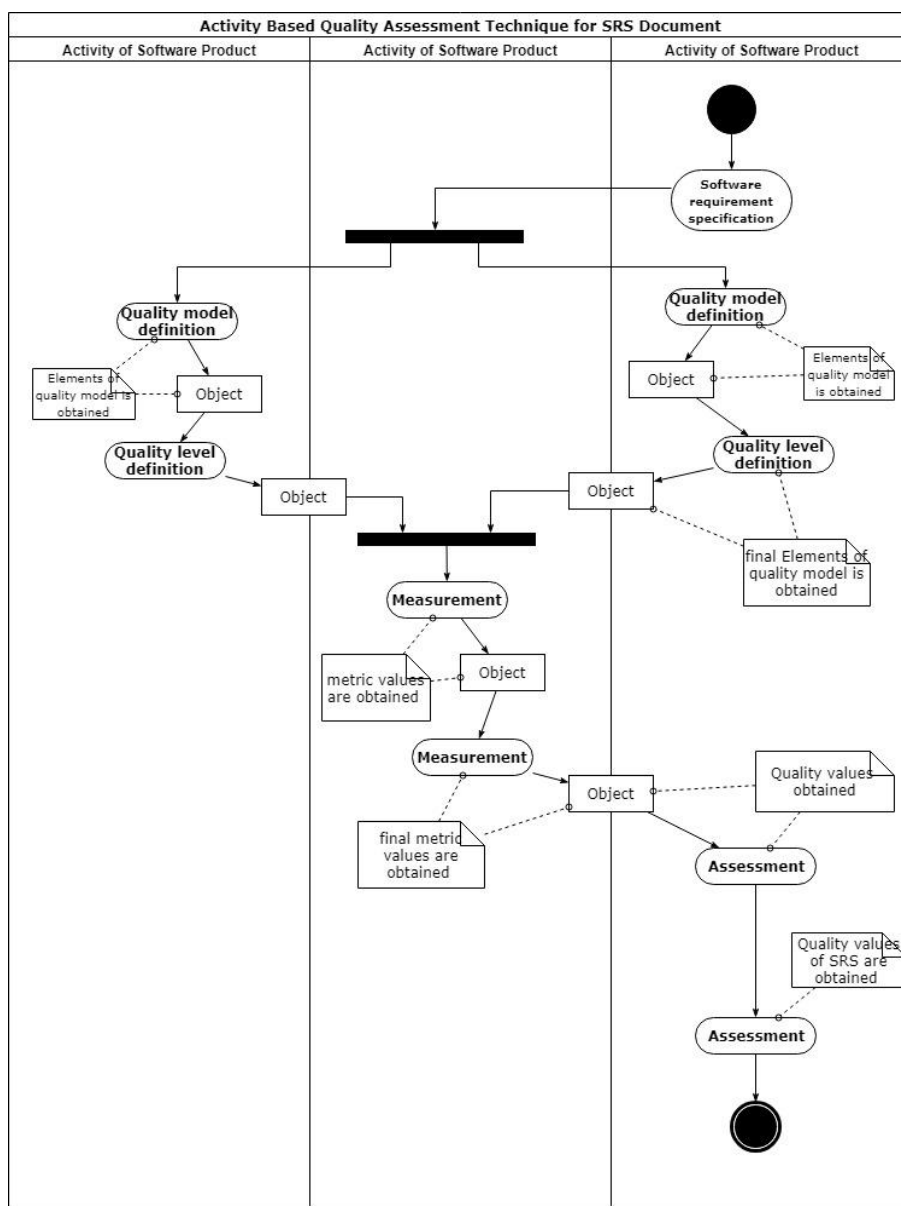


Fig 4: Proposed Activity based SRS Quality Model

Table 1. Software excellence model for software requirement specification

Metric ID	Quality Attribute	Metric
1	Coverage	$M_1 = \frac{n_{ui}}{n_r}$
2	Usability Compliance	$M_2 = \frac{n_u}{n_i \times n_s}$
3	Data Exchangeability	$M_3 = \frac{n_c}{n_c * n_{NV}}$
4	Availability	$M_4 = \frac{n_r}{n_r + \sum_i c(ri) + \sum_i t(ri)}$
5	MS-Software Compliance	$M_5 = \frac{n_f}{n_u}$
6	Internally consistent	$M_6 = \frac{n_u - n_n}{n_u}$
7	Design Independent	$M_7 = \frac{D(R_E \cup R_I)}{D(R_E)}$
8	Concise	$M_8 = \frac{1}{size + 1}$

Model 3: Software Quality Level Definition

Input: Software quality model defined by various metrics.

Productivity: Assessment of those elements of software quality model which is shown in Table 2.

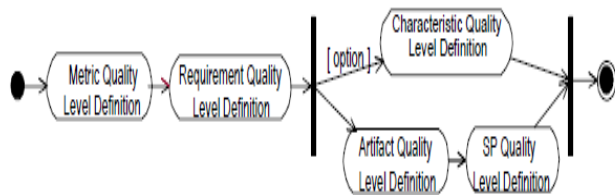


Fig 5: Quality Level Definition Activity

Table 2. Definition of Software Quality Requirement Specification Development

Requirements	ID_Metric	Software assessment function definition
RQ1	1	Min>0.70; Goal=1.0
RQ2	2	Min>0.4; Goal>=0.8
RQ3	3	Min=0.6; Goal=1.0
RQ4	2	Non-satisfactory<=0.4;Min<0.3; Goal=0.99; Exceeding>0.99
RQ5	4	Non-satisfactory<=0.7;Min<1.0; Goal=1.00

Let us supposed the appraisal functions of software quality levels are defined according to the software models, the software metrics has been defined in Table 1. The software metrics also defined as follows.

Non-satisfactory if xCode = Non-satisfactory

Software product-level (xCode, xDoc) = Min if xCode = Min (4.1)

Goal if xCode = Goal and xDocMin

More than if xCode = Exceeding and xDocMin

The software assessment functions can be described by customer as follows

Model 4: Software Measurement

Input: Software documentation and software coding of system.

Productivity: Artifacts are measured the resulting software metric, the various software metric values obtained in table 3.

Table 3. Different metrics performed by software measurement

Id_Metric	Software Metric	Performed Software Metric values	
		Software Lines of Code	Software Development
1	Treatment	SC=0.88	SD=0.41
2	Usability fulfilment	SC=0.90	SD=0.81
3	Data Exchangeability	SC=1	NA
4	Ease of use	SC=0.6	SD=1
5	MS-Software fulfilment	SC=1	NA

The measurement of software performance performed in user environment. Where NA is not applicable

Model 5: Software Evaluation

Input: Grades of software activity models 3 and 4 are considered

Productivity: Measuring the software quality values for different levels

By means of software assessment function defined in software activity model 4, as per the level of each assessment the software metrics as performed in table 4-7. The software assessments has been performed separately on each level for every artifact. The software requirement has one linked with the another software metric, the results of software quality values for software requirement quality level are the same as per software metric quality level.

Table 4. Performed Software quality as per software quality levels

Quality Requirements	ID_Metric	Performed Software Quality Values	
		Code	Development
RQ1	1	Min	Non-satisfactory

Activity Based Quality Assessment Technique for Software Requirement Specification

RQ2	2	Goal	Min
RQ3	3	Goal	NA
RQ4	4	Min	Goal
RQ5	5	Goal	NA

Table 5. Performed software quality principles for pre-characteristics

Pre-characteristic	ID_Metric	Performed Software Quality Values	
		Software Code	Development
Appropriateness	1	Min	Non-satisfactory
Interoperability	1, 3	Goal	NA
Usability fulfilment	1, 2	Goal	Min
Recoverability	4	Min	Exceeding

Table 6. Performed software quality values for

Table 8: Contrast between proposed work and related work

	First group [6-8]	Second group [9-14]	Proposed work
Purpose	Software requirement specification	To assess quality of software requirement specification	To assess quality of software requirement specification
Methodology	Machine Learning	Direct Metrics	Activity based and Metrics
Advantages	Automated	Obtained good Characteristics	<ul style="list-style-type: none"> • Obtained good Characteristics • Based on metrics good document is obtained • Based on human need the quality threshold can be apply.
Limitations	Efficiency is based on the Machine Learning and the training Data set	Human intervention is needed to define metrics	Clearly specified in the natural language with the predefined document structure

IV. CONCLUSION:

The assessment of the software requirement specification is very important and difficult. In this paper, we developed the activity-based assessment model for software requirement specification. This model allows keeping up a complete and neat separation of all activities in developing for the target systems. Each Activity in this process is treated as a separate module in the model. Step by step in all the

following quality values

Quality Level	Performed Quality Value	
	Software Code	Development
Functionality	Min	Non-satisfactory
Usability	Goal	Min
Dependability	Min	Exceeding

Table 7. Performed software quality values for following artifacts level

Level of Artifacts	Performed Software Quality Value
Software Code	Min
Development	Non-satisfactory

The final evaluation of the software requirement specification system, the software quality is minimal according to the aforesaid software models, the definition of software assessment function for the software product is compare between the related researches and proposed research work as shown in the table 8.

activities leads to obtain the good characteristics of the software requirement specification. Activity by activity is evaluated in this example to show the effectiveness of the proposed model. Finally the Comparison between existing and proposed work is tabulated.

REFERENCES:

1. K. Lochmann, D. Méndez Fernández and S. Wagner "A Case Study on Specifying Quality Requirements Using a Quality Model" APSEC December 4-7, 2012, Hong Kong Copyright 2012 ACM.
2. Ron S. Kenett, "SOFTWARE SPECIFICATIONS METRICS: A quantitative approach to assess the quality of documents" 0-7803-3330-6/96/\$5.00 1996-IEEE.
3. Gnesi, S.: Analysis of Software Requirements. IEI-CNR Pisa <http://www.iei.pi.cnr.it/ERI/iei/qmslideseri.ppt> (2000).
4. Roxana Saavedra1, Luciana Ballejos2, Mariel Ale3, "Quality Properties Evaluation for Software Requirements Specifications: An Exploratory Analysis"
5. Patra Thitisathienkul, Nakornthip Prompoon, "Quality Assessment Method for Software Development Process Document based on Software Document Characteristics Metric", Journal of Intelligent Computing Volume 5 Number 4 December 2014.
6. Hussain, I., Ormandjieva, O., Kosseim, L. (2007). Automatic Quality Assessment of SRS Text by Means of a Decision-Tree- Based Text Classifier. Paper presented at the Quality Software, 2007. QSIC '07. Seventh International Conference on.
7. Jani, H. M. (2010). Applying Case-Based Reasoning to software requirements specifications quality analysis system. Paper presented at the Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on.
8. Mat Jani, H., Tariqul Islam, A. B. M. (2012). A framework of software requirements quality analysis system using case-based reasoning and Neural Network. Paper presented at the Information Science and Service Science and Data Mining (ISSDM), 6th International Conference on New Trends in.
9. Davis A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, et al. (1993). Identifying and measuring quality in a software requirements specification. Paper presented at the Software Metrics Symposium. In: Proceedings., First International.
10. Kenett, R. S. (1996). Software specifications metrics: a quantitative approach to assess the quality of documents. Paper presented at the Electrical and Electronics Engineers in Israel, Nineteenth Convention of.
11. Lami, G., Gnesi, S., Fabbri, F., Fusani, M., Trentanni, G. (2004). An automatic tool for the analysis of natural language requirements. Informe técnico, CNR Information Science and Technology Institute, Pisa, Italia, Setiembre.
12. Jani, H. M. (2007). Online Quality Analysis of the Requirements Specifications Phase of the Software Development Cycle. Paper presented at the South East Asian Association for Institutional Research (SEAAIR), Bangkok, Thailand.
13. Génova, G., Fuentes, J., Llorens, J., Hurtado, O., Moreno, V. (2013). A framework to measure and improve the quality of textual requirements. Requirements Engineering, 18 (1), 25-41.
14. Carlson, N., Laplante, P. (2014). The NASA automated requirements measurement tool: a reconstruction. Innovations in Systems and Software Engineering, 10 (2), 77-91.
15. Iwona Dubielewicz1, Bogumia Hnatkowska1, Zbigniew Huzar1, Lech Tuzinkiewicz1, "An Approach to Software Quality Specification and Evaluation (SPoQE)
16. B.Mallikarjuna, P.Venkata Krishna "VCZRP: Enhancing Performance of Mobile Ad hoc Networks Using Virtual Coordinates" Indian Journal of Computer Science and Engineering(IJCSE), Volume.3 No.2, Apr-May 2012 ISSN: 0976-5166 page 298-303.
17. B.Mallikarjuna, N. Kusuma "Versatile Time cost Genetic Algorithm (VTCGA) for Non Preemptiv Critical Workflow tasks in Cloud Computing Environment" International Journal on computing communications and Systems(IJCCS), Vol-4, Issue-1 Feb-2015.
18. B.Mallikarjuna, K.V.Ramanaiah, P.Mohanaiah, and V.Vijay Kumar Reddy "Recognizing Human Object Using Genetic Algorithm for Face Detection in Natural Driving Environment" i-manager's Journal on Software Engineering, ISSN: 2230-7168 Vol. 7 No.2, and October-December 2012 page 10-18.
19. B.Mallikarjuna, V.Saritha, Ch.Puspalatha "Fault tolerant Resource Management Mutual Exclusion Algorithm for Mobile Ad hoc networks" International Journal of Scientific and Research Publications, Volume 2, Issue 4, April 2012 Edition, ISSN 2250-3153.
20. VeeraRaghavulu, B Mallikarjuna "Methods for seeking of information on the WWW" International Journal of Scientific and Research Publications, Volume 2, Issue 3, March 2012 Edition" Page 332-337.