How to cite this paper:

Emmanuel O.C. Mkpojiogu & Nor Laily Hashim. (2017). Improving the quality of requirements work products using scoring rubrics-assisted reading in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference on Computing & Informatics (pp 656-662). Sintok: School of Computing.

IMPROVING THE QUALITY OF REQUIREMENTS WORK PRODUCTS USING SCORING RUBRICS-ASSISTED READING

Emmanuel O.C. Mkpojiogu[1], and Nor Laily Hashim[2]

[1]*Universiti Utara Malaysia, Malaysia, emelnuel*@hotmail.com
[2]*Universiti Utara Malaysia, Malaysia, laily*@uum.edu.my

ABSTRACT. The quality of requirements documents and of software relies on how satisfactory the documents are read and reviewed. A number of reading techniques have been proposed. These techniques help in the reviewing and improvement of the quality of software work products. However, in literature, the quality improvement efficacies of the various techniques are at varying levels. The most common reading technique is checklist reading technique. This technique, even though is an enhancement over the ad hoc reading technique, is nonetheless, encumbered with some drawbacks that affect its usability and efficacy as it offers too broad questions to readers devoid of any precise guidance on how to undertake the reading process. This paper therefore, proposes the use of scoring rubric assisted reading (SRAR), a checklist-like reading technique for the evaluation of the quality of software requirements documents. The objective of the paper is to assess the quality improvement efficacy of SRAR. Scoring Rubrics were used in reading and reviewing requirements artifacts. Initial outcomes reveal the efficacy of this technique. The study will benefit both professionals and the research community as this technique is conceptually a better reading mechanism.

Keywords: software quality, requirements work products, scoring rubrics

INTRODUCTION

Software reading or review is the textual analysis of software work products with the aim of detecting defects that are underlying therein (Basili et al., 1996). Reading involves evaluating the form, structure, content and documentation of software work documents and products. The individual analysis of textual software work documents is a core activity in many software engineering tasks such as verification and validation, maintenance, evolution and reuse etc. The purpose of software reading is to achieve an improvement in the quality of work products and ultimately of the software as a whole. It is a key technical activity in the software quality assurance process and is required all through the development life cycle. Work documents (such as requirements, requirement specification, design, code, test plan, etc) related to the various phases of software development, are often subjected to regular reviews, modification and continual understanding. In reviewing work products, the documents are read and analyzed to assess the various qualities and characteristics so as to improve software product quality (Basili et al., 1996). Effectively detecting defects in requirements work products at the early phases of the software development life cycle helps in lowering the chances

of defects slipping into the later phases of the development life cycle. Defects in the later stages of software development are more costly to correct (Basili et al., 1996) in terms of rework, effort, finance, manpower and end product delivery delays, etc. Hence, it pays off to begin the software quality assurance efforts early in the software development process. Software quality assurance of work products is usually in the form of formal design review, inspection or walkthrough with varying degrees of formality, structure, and team participation. Inspection is a frequently used method in verification (Sommerville, 2007).

Reading techniques help in the reviewing and improvement of the quality of software work products. These techniques among others include: ad hoc reading, checklist-based reading, scenario-based reading, defect-based reading, and perspective-based reading. Defect-based reading and perspective-based reading are both types of scenario-based reading (Basili et al., 1996; Shull et al., 2000; Halling et al., 2001; Kollanus & Koskinen, 2007). However, the quality improving efficacies of the various techniques are at varying degrees. The most common reading technique is checklist reading technique (Basili et al., 1996; Shull et al., 2000). This technique, although is an improvement over the ad hoc reading technique, is however, flagged with some limitations that affects its usability and efficacy (Shull et al., 2000; Basili et al., 1996). It provides too general questions to readers without any specific guidance on how they can go about the reading process. This paper therefore, proposes a conceptual reading technique, scoring rubric assisted reading (SRAR), a checklist-like reading technique for the evaluation of the quality of software requirements documents. The objective of the paper is to evaluate the quality improvement capacity and efficacy of SRAR. The paper is structured as follows: Section 2 presents the background; section 3 presents the proposed approach; section 4 is the methodology; section 5 is the results and discussion section; and lastly, section 6 concludes with summary, conclusion and future works.

BACKGROUND

There are several reading techniques identified in the literature. These include: ad hoc reading, checklist-based reading, scenario-based reading, defect-based reading, and perspective-based reading. Defect-based reading and perspective-based reading are both types of scenario-based reading (Basili et al., 1996; Shull et al., 2000; Halling et al., 2001; Kollanus & Koskinen, 2007). The ad hoc reading approach allows reviewers to use their knowledge and judgment in the detection of defect. It does not provide any guide or assistance to the inspector(s). In this approach, there are no well defined procedures and reviewers learn largely by practice. Reviewers gain expertise gradually with continuous repeated practice (Shull et al., 2000). Users of ad hoc reading solely depend on their intuition and past experience. This approach is prune to faults, and it is not very effective and efficient (Mkpojiogu & Hashim, 2017a). More so, the difficulty in offering training for a poorly defined or undefined process like the ad hoc reading process compounds the problem. Therefore, the process cannot be improved upon and it is not repeatable (Shull et al., 2000). In checklist-based reading, checklist(s) is/are used. A checklist consists of a set of general questions concerning types of defects or the possible signs (pointing to defects) to look out for in a given type of document. Checklist aids the reviewer/inspector in reading, remembering and recollecting the aspects that are to be checked and reviewed. However, it provides little guidance on what specifically the reviewer is to do. The reviewer/inspector has to map checklist questions to tasks and plan how to traverse the document to be reviewed. Checklist does not allow for the reading process to be repeatable and it is also open to faults and changes/variations (Halling et al., 2001). One checklist in most cases is used by all the reviewers in a team, but does not enable a focused coordination of the work of the various members of the team. This may lead to effort wastage in the team. Checklist can cover a wide range of areas/issues, but it still requires reviewers reading through the document sequentially and that, severally. Thus, this limits the applica-

tion of checklists to only documents with limited size. The use of checklists may make the reviewers to revert back to ad hoc reading by using their personal intuition and experience (Halling et al., 2001).

Scenario-based reading uses procedures to find and detect specific classes of defects. This technique guides readers through a given document with a particular point of view or emphasis. It provides scenarios that offers step by step guideline and stimulate readers to work actively with the document through note taking, annotating of the document and building a mental framework or model that can lead to more consistency in the given view of a reader (Halling et al., 2001). Scenarios provides guidance on different levels of details, beginning from organizing entities, and instructing reviewers on how to recognize them and to abstract information that are relevant, and add them to the analysis. These steps are repeated on several levels of detail (Halling et al., 2001). This technique offers some advantages: the process can be repeated, audited and also provides the planners of inspection, a means to assign inspectors/reviewers to where to focus, in a prescribed form for particular defects (Halling et al., 2001). Scenario-based reading consists of the following reading types: defect-based and perspective-based reading. The above descriptions reveal the inadequacies in ad hoc and checklist reading techniques. However, this study seeks to propose a checklist-type reading technique that overcomes the shortcomings of checklist and ad hoc reading techniques.

PROPOSED APPROACH

The study proposes the use of scoring rubrics-assisted reading (SRAR) technique to assist in requirements work products/ artifact reading, defect detection and in improving the quality of work products. Checklists are simplified rubrics. Moreover, rubrics are better off than checklists because they provide detailed clues based on the attributes of the given work product along all dimensions of the work product, rather than just offering general questions that do not provide specific guidance as in the case of checklists. It also scores the quality of the attributes of the document as well as the entire work product. Thus, it provides a platform to measure and monitor progress in the quality improvement of a given work document or artifact. Rubrics are reliable, valid and consistent. Their assessment process can be repeated with a good level of consistency unlike checklists or the ad hoc approach. Rubrics are two dimensional, with rows and columns. The rows are the attributes of the particular work product while the column is a Likert-type rating scale that clearly describes the various expected quality level of the row-wise work product attributes. The qualities graduate in ordinal form from no quality to the most expected quality. Rubrics are commonly used in the field of education to assess the performance of students' work and projects. It is an instrument used for assessment and evaluation. It offers uniform and consistent evaluation platform (Jonsson & Svingby, 2007; Dietrich, 2008; Sherman & Martin, 2015; McKenzie & Wood-Bradley, 2014; Dessai et al., 2014). In the field of education, the emphasis is on the performance of the student. However, in reality, it is the work of the student that is actually being assessed for quality and not the student per se. This educational tool can be adapted and tailored to software reading, with the focus being on using the mechanism to assess and analyze software work products and artifacts, with the aim of detecting defect in them. In software reading, the emphasis is on the work product and not necessarily on the reviewer because the purpose of software reading is to review software documents and find errors in them. This leads to the subsequent removal of such defects and the eventual improvement of the quality of the given work product (Mkpojiogu & Hashim, 2017a; 2017b). However, this study does not compare methods but as a preliminary study, it shows how requirements work products' quality can be improved using the scores of rubrics tool.

METHODOLOGY

In this paper, the scoring rubric instrument was used to evaluate the quality of requirements work documents. A scoring rubric is a two-dimensional Likert-like instrument. The columns represent a 4-point Likert-type rating scale (for example, 1. Not acceptable, 2. Below expectations, 3. Meet expectations, 4. Exceeds expectations); A fifth column is added for "not applicable" (see Table 1). The rows consist of the attributes of the given work products. In some cases, the attributes are further defined into criteria. In addition, the cells created by the intersection of the rows and columns represent a clear description of the work products' attributes with respect to the corresponding rating scale. Each attribute is scored and the scores of all attributes are totaled to form a total score for the given work product that represents its relative quality. The scores of the rubrics are used as metrics to assess the quality of the requirements documents. Higher rubric scores indicate higher quality (that is, lower defects) and lower rubric scores reveal lower quality of software requirements documents (that is, higher defects).

Table 1. Scoring Rubric Framework

Attributes	Criteria (optional)	Scale of Score					Score
		1.(Not acceptable)	2.(Below expectations)	3.(Meets expectations)	4. (Exceeds expectations)	N/A (Not available)	
$Attribute_1$	…	$Cell_{11}$	$Cell_{12}$	$Cell_{13}$	$Cell_{14}$	…	…
…	…	$Cell_{21}$	$Cell_{22}$	$Cell_{23}$	$Cell_{24}$	…	…
…	…	$Cell_{31}$	$Cell_{32}$	$Cell_{33}$	$Cell_{34}$	…	…
Attribute	…	$Cell_{n1}$	$Cell_{n2}$	$Cell_{n3}$	$Cell_{n4}$	…	…
Total Score							(%)

This study was conducted at the School of Computing, Universiti Utara Malaysia (UUM). The Software Engineering sub-department of the School has over the years been using scoring rubrics in evaluating the quality of students' software/requirements documents and models. But the tool has not been empirically validated. This study is an attempt to provide an empirical evaluation of the efficacy of the mechanism. The study was part of the study carried out to develop an e-health awareness system (Hussain, Mkpojiogu & Kamal, 2015; Hussain & Mkpojiogu, 2016; Hussain, Mkpojiogu & Nawi, 2016; Mkpojiogu & Hashin, 2016). After the requirements documents and models were produced, they were read and reviewed in two iterations (rounds) by four (4) reviewers who detected defects in the documents and scored the artifact. Each reviewer read, reviewed and scored the requirements documents and models independently. After each round of review, the requirements work products were refined and the defects removed. The review of requirements models was done by four (4) Senior Lecturers (in Software Engineering) in the School of Computing, Universiti Utara Malaysia (UUM). The following research question guided the study: Does Scoring Rubric-assisted reading improve the quality of requirements work products? As this study is an exploratory one, descriptive statistics was used in answering the research question. The scores of the scoring rubric were used as metrics in the study. The study's data (rubric scores) were presented as averages and percentages.

RESULTS AND DISCUSSION

Rubrics Scores

Table 2. Average rubric scores for two rounds of reading: Overall Average: 70.72

Work Product	Rubric Score (Round 1)	Rubric Score (Round 2)	Difference	% Increase in Quality	Improvement?
Vision & Scope Doc	73.44	87.50	14.06	19	√

List of Reqs	25.00	50.00	25.00	100	√
Use Case Description	81.25	89.29	8.04	10	√
Software Reqs Spec	71.88	87.50	15.62	22	√
Test Plan	62.09	86.25	24.16	39	√
Test Cases	64.29	80.36	16.07	25	√
Ave. Work Product (Docs)	56.22	72.71	16.49	29	√
Use Case Diagram	53.57	78.57	25.00	47	√
Activity Diagram	50.00	62.50	12.50	25	√
Sequence Diagram	89.29	78.57	-10.77	-12	**X**
Collaboration Diagram	64.29	78.57	14.28	22	√
Class Diagram	59.38	82.14	22.76	38	√
Ave. Work Products (Models)	63.30	76.07	12.77	20	√
Ave. Work Products (Docs & Models)	45.06	56.95	11.89	26	√

This exploratory reading study was assisted by the use of Scoring Rubrics instrument. The results as shown in the table 2 above indicates a good improvement in the quality of the requirements work products. Rubric scores (percentage scores) are performance metrics. Among the documents read and reviewed with their corresponding percentage increase in quality include: vision and scope document (19%), list of requirements (100%), use case description (10%), software requirements specification (22%), test plan (39%), test cases (25%), use case diagrams (47%), activity diagrams (25%), sequence diagrams (-12%), collaboration diagrams (22%), and class diagrams (38%). From table 2 above, the average quality improvement for all textual documents is 29%. Also, the average percentage increase in quality for all requirements models put together is 20%. In addition, the average percentage increase in quality for all work documents read and reviewed (that is requirements documents and models) is 26%. As can be seen, the percentage increase in quality ranged from 10 to 100 percent increase. Almost all the work products reviewed showed observable improvement in quality. Only one work product had -12%. The overall percentage quality of the entire work documents reviewed is 70.72%. These results indicate a good quality improvement rate. It provides some noticeable evidences to show that the use of SRAR enhanced the quality of the inspected products within the two rounds of reviews, thus attesting to the technique's efficacy.

Table 3. Reliability analysis of work products rubrics: Overall Cronbach Alpha coefficient: 0.987

Work Product	No of Attributes	Mean	Variance	Std Deviation	Cronbach Alpha
Vision & Scope Doc	8	80.469	171.712	13.104	.986
Use Case Desc	8	42.634	2434.307	49.339	.978
SRS	8	79.688	289.714	17.021	.970
Test Plan	10	74.166	334.741	18.296	.940
Test Cases	7	72.321	258.325	16.073	.931
Use Case Diagram	7	33.036	1559.323	39.488	.964
Activity Diagram	2	28.125	1080.729	32.875	.867
Sequence Diagram	7	41.965	2367.213	48.654	.981
Collaboration Diagram	7	35.715	1734.716	41.650	.974
Class Diagram	8	35.380	1755.342	41.897	.966
Overall	72	523.497	88609.739	297.674	.987

The result of the reliability analysis as shown in Table 3 reveals that the Cronbach alpha coefficient (α) of the work products rubrics scores ranged from 0.867 to 0.986. Vision and scope document with 8 attributes has α = .986. The use case description with 8 attributes has α = 9.78. The software requirements specification (SRS) with 8 attributes has a Cronbach alpha (α) of .970. Also, the test plan with 10 attributes has a Cronbach alpha (α) of .040. Other work products with their corresponding number of attributes and α coefficient are as follows: test cases (7, .931), use case diagram (7, .964), activity diagram (2, .867), sequence diagram (7, .981), collaboration diagram (7, .974), class diagram (8, .966). Furthermore, the

overall reliability coefficient is 0.987. The entire instrument has a total of 72 attributes. These Cronbach alpha scores are indicative of very good internal consistency. Reliability scores that are 0.70 and above, are normally regarded as good reliability scores (Nunnaly, 1978); therefore, scoring rubric reading instrument is highly reliable and its use can produce consistent results. The tool has also been face validated. Therefore, scoring rubric instrument is not only reliable, but is also valid for the reading and review of software requirements work documents, for the detection of defects and the improvement of the quality of the documents.

SUMMARY, CONCLUSSION AND FUTURE WORKS

In sum, it is observed that after round two of the reading and review, there were observable improvements in the work documents quality levels as seen in the metrics used. This indicates the efficacy of the reading mechanism. The rubric score for all textual documents put together, improved in quality by 29%. Furthermore, the rubric score of all requirements models combined improved by 20% and the one of requirements models combined with textual documents improved by 26%. The overall rubric score of 70.72% shows that on the overall, the average quality of the work product was good. Thus, SRAR promises to be a reading technique with good efficacy, which overcomes the inadequacies of the checklist reading technique. It is simple, guiding, defined, consistent and scoring. It has the capacity of assisting reviewers/readers detect defects in software/requirements work products. In addition, it can be used to monitor the progress of quality improvements of work products. However, this study was an exploratory one and was limited in the sense that SRAR was not empirically compared with checklist or any other reading technique, but the efficacy was judged based on the defect detection capability and quality improvement potentials as observed from the study. Future works will concentrate on evaluating and comparing SRAR with other reading techniques using both professional and non-professional users/readers.

REFERENCES

Basili, V., Caldiera. G., Lanubile, F., & Shull F. (1996). Studies on reading techniques. Institute of Advanced Computer Studies, University of Maryland, USA.

Dessai, K.G.G., Kamat, V.V., & Wagh, R.S.. (2014). Effective use of rubrics in computer assissted subjective answer script evaluation. 2014 IEEE 6th International Conference on Technology for Education (T4E), Clappana, 18-21 Dec. 2014, p. 95-98.

Dietrich, S. (2008). Quality-based assessment of papers and projects in computer science. *J Comp Sci Colleg*. 23(3), 16-22.

Halling, M., Biffl, S., Grechenig, T., & Kohle, M. (2001). Using reading techniques to focus inspection performance. Proceedings of the 27th 2001 IEEE Euromicro Conference, Warsaw, 4-6 September, 2001, pp. 248-257.

Jonsson, A., & Svingby, G. (2007). The use of scoring rubrics: reliability, validity and educational consequences. *Education Research Review.* 2, 130-144.

Hussain, A., Mkpojiogu, E.O.C., (2016). An application of Kano method in the elicitation of stakeholder satisfying requirements for an e-Ebola awareness system. *International Journal of Systems Applications, Engineering and Development*. 10, 169-178; ISSN: 2074-1308;

Hussain, A., Mkpojiogu, E.O.C., & Kamal, F.M. (2015). Eliciting user satisfying requirements for an e-health awareness system using kano model. In Xiaodong Zhuang (Ed.) Recent Advances in Computer Science, pp.156-165. WSEAS press. ISBN: 978-1-61804-297-2.

Hussain, A., Mkpojiogu, E.O.C. & Nawi, M.N.M. (2016), Requirements model for an e-health aware ness portal. Proceedings of the International Conference on Applied Science and Technology (ICAST'16), Kedah, Malaysia, AIP Conf. Proc., 1761 (1), 020048

Kollanus, S., Koskinen, J. (2007). Survey of sftware inspection research: 1991-2005. Computer Science and Information Systems Reports Working Papers WP-40. Department of Computer Science and Information Systems, University of Jyvaskyla, Finland.

McKenzie, S., Wood-Bradley, G. (2014). Using rubrics in IT: Experiences of assessment and feedback at Deakin University. 2014 International Conference on Teaching, Assessment and Learning for Engineering (TALE), Welligton, 8-10 Dec., 2014, pp. 474-479.

Mkpojiogu, E.O.C., & Hashim, N.L. (2016), Understanding the relationship between Kano model's customer satisfaction scores and self-stated requirements importance, *SpringerPlus*. 5 (1), 1-22; doi: 10.1186/s40064-016-1860-y

Mkpojiogu, E.O.C. & Hashim, N.L. (2017a), Assessing the efficiency of scoring rubric-assisted reading (SRAR) approach in the review of software requirements work products, *Journal of Engineering and Applied Sciences (JEAS)*

Mkpojiogu, E.O.C. & Hashim, N.L. (2017b), Evaluating the effectiveness of scoring rubric-assisted reading (SRAR) technique in the reading of software requirements work documents, *Journal of Engineering and Applied Sciences (JEAS)*

Nunnaly, J., (1978). *Psychometric Theory*, New York: McGraw-Hill.

Sherman, M., Martin, F. (2015). The assessment of mobile computational thinking. *J Comp Sci Colleg*. 30(6), 53-59.

Shull, F., Rus, I., & Basili, V. (2000). How perspective.based reading can improve requirements inspections, *Computer*, 33(7), 73-79.

Sommerville, I. (2007). *Software Engineering*. Eigth Edition, Addison-Wesley.