ISSN 2278-3091

Volume 8, No.1.3, 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse4081.32019.pdf

https://doi.org/10.30534/ijatcse/2019/4081.32019

Non-Functional Requirement Template for Usability Aspect based on NIMSAD Evaluation Engineering

Noor Atikah Amira Fauzi[1], Rohayanti Hassan[1*], Muhammad Barja Sanjaya[2], Zuraini Ali Shah[1], Asraful Syifaa' Ahmad[1], Shahreen Kasim[3]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[2]School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia

[3]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Batu Pahat, Johor, Malaysia, rohayanti@utm.my

ABSTRACT

In requirement engineering, non-functional requirement is a requirement that specifies characteristics of system behavior and sys-tem quality attributes. Furthermore, a non-functional requirement template facilitates system stakeholders in better system documentation, system elicitation and system traceability. However, some non-functional requirements may come from many type of requirements document and no stringent standard has been applied that lead to various pattern of non-functional requirements tem-plate. Specifically, in usability requirement, the quality attribute is being ignored and less expressive in majority requirements document. Therefore, this study was motivated to propose the most feasible non-functional requirement template for usability aspect. NIMSAD evaluation is used to obtain the most feasible non-functional requirement template by comparing existing non-functional requirement templates based on the following criteria which are i) general concepts, ii) modeling concepts and iii) analysis concepts. From the NIMSAD evaluation results, it is found that Boilerplates template is the most feasible non-functional requirement tem-plate for usability aspect.

Key words : Non-Functional Requirement Template, NIMSAD Evaluation, Usability, System Quality Attribute.

1. INTRODUCTION

Non- functional requirement (NFR) template is a set of simple structure that can be used to extract requirement statement. NFR template is used to facilitate the traceability and the elicitation process. Thus, some of researches introduced template which aim to help formulating NFR [1]. Generally, during the system development, NFR need to be tested to know how well the system executes its functions. Usability in NFR is hard to be satisfied because the only way to know the system is usable is by having real users try it out. Furthermore, usability requirement is among NFR that is hard to interpret completely in the template [2]. Therefore, this study has pro-posed an investigation to determine the suitable nonfunctional requirement template for usability aspect. The investigation was conducted based on the Normative Information Model-based Systems Analysis and Design Evaluation (NIMSAD) framework.

This paper is organized as follows. In section 2, the related works are given. In section 3, the normative information model-based method based system analysis and design evaluation is presented. The experimental results of comparative evaluation proposed in this paper is also presented in section 4. Finally, our work of this paper is summarized in the last section.

2. RELATED WORKS

A successful system depends upon adherence to NFR. When NFR are ignored issues can arise. The challenges in requirement documentation impress NFR being express. The big companies include government sector contain critical NFR for projects. To meet the goal, the idea of extracting NFR easily with use the best and suitable template. The use of a template for non-functional during requirement phase deals benefits to stakeholder [2]. There are several types of templates that have been discussed and proposed by previous researcher. Example of the templates are Easy Approach to Requirements Syntax (EARS) [3], Rupp's3, CESAR, Parameterized Safe-ty Requirements Templates [4] and Boilerplate [5]. From the previous researcher these five templates have been discussed. These template has their criteria based on the suitability. Some templates focus on functional and some focus on specific quality. The templates have been designed to improve the proficient and quality of the requirements statement in variety quality aspect. In turn, appropriate written template for requirements should facilitate readable specification document.

In order to choose a feasible and suitable NFR template for usability, we implemented NIMSAD evaluation framework. NIMSAD is a framework that can be used to evaluate some methodologies which are important to be implemented in a system [6]. Those methodologies are compared then by using different multi criteria. After making a comparison, one of them will be chosen and applied on a desired system. NIMSAD concerns about problem and problem solving process in work. There are several researchers that used NIMSAD in their research work such as [7].

3. NORMATIVE INFORMATION MODEL-BASED SYSTEMS ANALYSIS AND DESIGN EVALUATION STYLE

Normative Information Model-based Systems Analysis and Design (NIMSAD) is a framework which is mainly used to analyse and evaluate the existing non-functional requirement templates that best fit for usability type requirement. Figure 1 illustrates the Software Process Engineering Meta-Model (SPEM) that demonstrates how NIMSAD is used in this study which comprises three major phase: i) Initialization, ii) Evaluation and iii) Analysis. The process begins by defining the criteria, sub-criteria and question for existing non-functional requirements template in Initialization phase. Next followed by the Evaluation phase that assess the existing non-functional requirements template based on criteria defined earlier. In the final phase, the existing non-functional requirements template is analysed for the usability fitness based on the prior evaluation. The process of NIMSAD is illustrated in Figure 1.

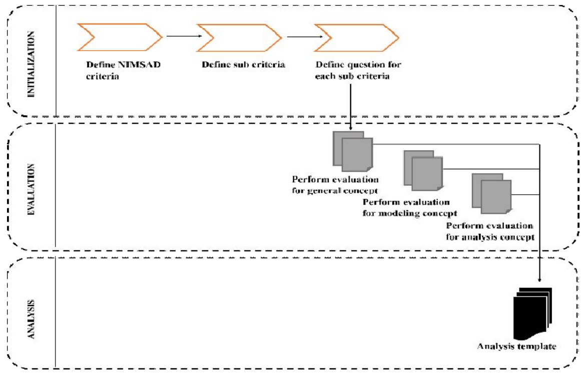
INITIALIZATION
Define NIMSAD criteria
Define sub criteria
Define question for each sub criteria
EVALUATION
Perform evaluation for general concept
Perform evaluation for modeling concept
Perform evaluation for analysis concept
ANALYSIS
Analysis template

Figure 1: Process of NIMSAD

3.1. NIMSAD Criteria Description

Table 1 tabulates the details criteria to evaluate the existing non-functional requirement template with the scope of interest is to find the best fit for usability aspect. The evaluation is limited to: i) general used and focus quality attributes, ii) modelling structure and iii) measuring metrics used.

Table 1: NIMSAD Criteria Description

Criteria	Sub-criteria	Questions
General Concepts	Goal	What is the primary goal of the NFR template?
	Use For	What are the main problems that the NFR template is used for?
	Quality attributes	What type of quality attributes that focus to be addressed?
	ISO Quality Model	What ISO quality standard is followed?
Analysis Concepts	Measuring Metrics	What are the measuring metrics can be used to access the NFR template?

4. RESULTS AND DISCUSSION

4.1. Comparison on General Used and Focus Quality Attributes Detailed

Five NFR templates have been compared and evaluated. There are EARS [3], Rupp's [3], CESAR, Parameterized Safety Requirements Templates [7] and Boilerplate [8]. Most of the template was designed that aim to represent the performance-based and functional-based type of requirements. None of them was designed for usability-based requirements. It was shown that the template was proposed in order to overcome the traceability issues in the requirements as well as to improve the proficient and the quality of the requirement statements. Most researchers focused in designing the non-functional requirements template that comply the completeness, consistency and unambiguity. Furthermore, majority of the non-functional requirements template was designed based on ISO 9126 quality model, except for Parameterized Safety Requirements [4] that followed ISO 26262 and ISO 29148.

4.2. Comparison on General Used and Focus Quality Attributes Detailed

The five NFR templates using structured natural language to represent the needs and statements elicited from the stakeholders. All requirements document used in this study used natural language-based approach with various document templates and notations. Since the structure of the five referred template are too varies, thus this study generalized the template structure by using the notations in Table 2. There are eight notations were used in the template structure, which the NFR statements can be broken into several segments, for instance as follows:

i. Original non-functional requirement statements;

ii. The system shall allow the users to access the system from the Internet using HTML or its derivative technologies.

iii. The existing structured of non-functional requirements statement; the system #shall [allow the users to access the system from the] (Internet using HTML or its derivative technologies).

iv. The non-functional requirements statement is broken into segment using our notation; S1: The system S2: #shall S3: [allow the users to access the system from the] S4 :(Internet using HTML or its derivative technologies).

v. The proposed structure and notation of non-functional requirements; S1: The system S2: #shall/ should/will S3: [process] S4 :(object).

This notation template is made up of 8 parts. (1) An input variable. This notation is used for system name or any component or item name; (2) the action variable used for capturing functions that the system performs; (3) condition, the optional addition details about the action; (4) object is used for which functionality is needed. (5) A modal shall/should/will be specifying how important the requirement is. (6) Unit metric, this alternative is used for measurement unit to access action. (7) Segment, the segment is the sequence of structure template. The last is (8) underlined text, which is used for fixed text. The sentence that remain unchanged in the requirements. With using the proposed notations, the non-functional requirement statements may consist of combination of input variable, action variable, conditional, object, modal of importance, unit metric and/or fixed text. Table 3 presents the structure model for five non-functional requirements template using our proposed notations.

The minimum number of segment being used is four, which is EARS template [12] and Boilerplate template [9] for type 1, while for demonstrates the maximum number of segment being used is CESAR template which is 10. Requirement template without unit metric are EARS and Rupp's [3]. However, the rest of template are requirement template with metric.

4.3. Comparison on Modelling Structure

There are five metrics have been used by the existing studies in order to measure the effectiveness of NFR templates which are precision, recall, f-measure, stability, containment and feasible. EARS and Rupp's template [3] analyzing of effectiveness based on precision and recall. Precision is used for low number of false positives. Recall is used for low number of false negative. Besides, to be able to compare precision and recall f-measure is used to computes the mean. Boilerplates [9] covers feasible as measuring metrics.

4.4. Analysis on Usability Quality Attributes

Since none from the five templates above concentrate on the usability quality attribute, thus we extended our study on investigating the characteristics that frequently used by usability related ISO quality standard. Four types of ISO quality standard that support the usability aspect are known as ISO 9126, ISO 9241, ISO 25010 and ISO 12119. The ISO 9126 is a standard for determining the quality of software product for software evaluation.

It consists of four parts: quality model, external metrics, internal metrics and quality in use metrics. This standard can be used in many sectors. In addition, ISO 9126 standard has been upgraded to ISO 25010. The ISO 25010 standard is designed to strengthen the security and compatibility aspect of the system. However, ISO 25010 is very minimum focus on usability aspect which not cover understandability, effectiveness, efficiency, usability compliance and attractiveness.

Furthermore, ISO 9241 standard has been implemented for system usability and ergonomics. This standard focus to strengthen quality of interaction between a user and an interactive system. On the other hand, ISO/IEC 12119 is applicable to software package. Example are word processing, spreadsheets, database and presentation programs and utility programs. This standard deals only with soft-ware package as offered and delivered. In this study, we choose to use ISO 9126 as a guideline to construct on non-functional requirement template for usability aspect due to the following reasons:

i. This standard covers the most usability aspect such as learnability, understandability, operability, usability compliance and attractiveness compared to the other standard.

ii. More adaptable and can be used across many sector

Table 2: Notation used in Non-Functional Requirements Template

Notation	Name	Description	Example
< >	Input variable	The system, component or item name.	The ***<system>*** shall be available at all times.
[ ]	Action variable	The process of functionality which interacts with system.	The system shall be able to ***[send notification]*** by SMS.
{ }	Conditional	The optional about the action.	***{As soon as a power outage is detected},*** the Surveillance and Tracking module shall record (a warning in the system alert log file.
( )	Object	The object which the functionality is needed.	The system shall be able to send notification ***(by SMS)***.
#	Shall/ should/ will	Modal specifying how important the requirement.	The system ***#shall*** be available at all times.
| |	Unit metric	Measuring unit to assess the action.	The system shall respond not more than |***3seconds***|.
S1: … S*n*:	Segment	Sequence of structure.	***S1:*** The system ***S2:*** shall be able to ***S3:*** send notification ***S4:*** by short message services (SMS).
<u>Underlined text</u>	Fixed text	Sentence that fixed, remain unchanged in the requirements.	The system shall ***<u>be able to</u>*** send notification by SMS.

Table 3: Comparative Evaluation for Modelling Concepts

Types of template	**Criteria: Modelling Concepts**				
	Construction Structure				
	Construction template	**Input Variable**	**Action Variable**	**Conditional**	**Unit Metric**
EARS [3,5]	Structure 1: Ubiquitous Requirement S1: The S2: <system name> S3:#shall S4: [system response]	System name	System response	NA	NA
	Structure 2: Event Driven Requirement S1:When S2:{optional precondition} S3:<system name> S4:#shall S5:[system response]	System name	System response	Optional pre-condition	NA
	Structure 3: Unwanted Behavior Requirement S1: If S2:{optional preconditions} S3:Then S4: <system name> S5:#shall S6: [system response]	System name	System response	Optional pre-condition	NA
	Structure 4: State Driven Requirement S1: While S2:{in a specific state} S3:<system name> S4:#shall S5:[system response]	System name	System response	In a specific state	NA
	Structure 5: Optional Feature Requirement S1: Where S2:{feature is included} S3:<system name> S4:#shall S5:[system response]	System name	System response	Feature is included	NA
Rupp's [3]	Structure: S1: {When? Under what condition?} S2:<system name> S3:#shall/should/will S4:[process] S5:(object) S6:|additional details|	System name	Process	Condition	NA
CESAR Requirement Specification Languages	Structure: S1:The S2:<system name> S3:#shall S4:be able to S5:[action] S6:|entity| S7:at least S8:|number| S9:times per S10:|unit|	System name	Action	NA	-Unit -Number -Entity
Parameterized Safety Requirements [4]	Structure: S1:The S2:<system/Component/Item> S3:#shall S4:[avoid/not allow/ not causes] S5:[harm]	-System -Component -Item	Harm	NA	NA
Boilerplates [9,10]	Structure 1: S1:The S2: <system name> S3:#shall/ should/will S4:[process] S5:(object) Or S1:The S2: <system name> S3:#shall/ should/will S4:[process]	System name	Process	Conditions	NA
	Structure 2: S1:The S2:<entity> S3:#shall be able to S4:[action] S5: at least / not less than / within S6: |quantity| |units|	Entity	Action	NA	-Quantity -Units
	Structure 3: S1:{conditions} S2:the S3:<system name> S4:#shall/should/will S5:[process] S6: (object)	System name	Process	Conditions	NA

4.5. Feasibility Analysis

In this analysis, feasible score is used to determine which NFR template is the most feasible for usability aspect. For every original NFR statement, we transform them into five different tem-plates that were discussed earlier. Subsequently, the feasible score is then calculated for every NFR template using the following formula:

$$Not = \frac{\text{No.of match segmentation}}{\text{Total no.of segmentation}} \qquad (1)$$

where *Not* is a number of match segmentation (see Table 2) that being used in a selected transformed NFR statement. For example, given NFR statement as follows:

The system shall allow the users to access the system from the Internet using HTML or its derivative technologies.

1. Firstly, the NFR statement is transformed into respective templates as example below:
   a. Using Boilerplate template
   The <system> #shall [allow the users to access the system] from the (Internet using HTML or its derivative technologies)
   b. Using CESAR template

   The <system> #shall be able to [allow the users to access the system] from the Internet using HTML or its derivative technologies.

2. *Not* is calculated. See example below:
   a. **Structure by Boilerplates**; S1:The S2: <system name> S3:#shall/ should/will S4:[process] S5:(object) **is compared with transformed NFR statement,** S1: The S2: <system> S3: #shall S4: [allow the users to access the system] from the S5: (Internet using HTML or its derivative technologies)

Thus, from five segments in Boilerplate structure, all five segments found match in selected transformed NFR statement. *Not* score will be 5/5.

   b. **Structure by CESAR**; S1:The S2:<system name> S3:#shall S4:be able to S5:[action] S6:|entity| S7:at least S8:|number| S9:times per S10:|unit| **is compared with transformed NFR statement,**S1: The S2: <system> S3: #shall S5: [allow the users to access the system] from the Internet using HTML or its derivative technologies.

Thus, from seven segments in CESAR structure, only four segments found match in selected transformed NFR statement. *Not* score will be 4/7.

3. **Boilerplates** is chosen for this example of NFR statement since it has the highest score of *Not*, with 5/5:

Table 4 shows the feasible score for five different templates that being quantified on 25 NFR statements. Boilerplates is found to be the most feasible usability NFR template, since it obtained the highest chosen score which 20 out of 25 NFR statements got the highest Not. Boilerplates is mostly chosen is due to the flexibility of this template that able to match most NFR statements. In fact, in Table 2, Boilerplate provides three structures of template that flexible to the uncertain NFR statements. For Rupp's, it shows the least chosen for requirements. This is due to an optional condition is required at the beginning of the template structure which is not favor in most NFR statements.

Table 4: Feasible Score

Types of template	Feasible Score	
	No. of chosen requirements	Requirements no.
EARS [3,5]	9/25	[1], [2], [3], [4], [10], [11], [14], [16], [21]
Rupp's [3]	1/25	[22]
CESAR Requirement Specification Languages	8/25	[5], [6], [13], [18], [19], [23], [24], [25]
Parameterized Safety Requirements [4]	6/25	[1], [2], [3], [4], [8], [9]
Boilerplates [9,10]	20/25	[1], [2], [3], [5], [6], [7], [8], [9], [10], [11], [12], [14], [15], [16], [17], [19], [20], [21], [22], [25]

5. CONCLUSION

In this paper, NIMSAD evaluation framework has been implemented in order to determine the most feasible template for usability non-functional requirements statement. The results showed that Boilerplates is the most feasible template for usability non-functional requirements statement. However, we assume and limit that those 25 requirements that extracted from 11 specification documents are represented the usability aspect. For future work, we would like to automate the static review process by including the Boilerplates as a searching template guideline.

ACKNOWLEDGEMENT

This work is supported by MyMaster Scholarship of the Minis-try of Education Malaysia, RMC UTM, G-Heart scheme under the Gates Scholars Foundation and GUP grant, with Vot No: 16H73. We also would like to thank Universiti Tun Hussein Onn Malaysia for supporting this research under the Contract Grant (Vot number: U558), also, thanks to Gates IT Solution Sdn Bhd for the whole support

REFERENCES

1. Kopczyńska, S. & Nawrocki, J. **Using non-functional requirements templates for elicitation**: A case study. in *2014 IEEE 4th International Workshop on Requirements Patterns,*

RePa 2014 - Proceedings 47–54 (IEEE, 2014). doi:10.1109/RePa.2014.6894844

2. Lauesen, S. & Younessi, H. **Six Styles for Usability Requirement**s. in *REFSQ* 1–12 (1988).
3. Arora, C., Sabetzadeh, M., Briand, L. & Zimmer, F. **Automated checking of conformance to requirements templates using natural language processing**. *IEEE Trans. Softw. Eng.* 41, 944–968 (2015). https://doi.org/10.1109/TSE.2015.2428709
4. Antonino, P. O., Trapp, M., Barbosa, P. & Sousa, L**. The Parameterized Safety Requirements Templates**. in *Proceedings - 2015 IEEE/ACM 8th International Symposium on Software and Systems Traceability, SST 2015* 29–35 (IEEE, 2015). doi:10.1109/SST.2015.12
5. Mavin, A., Wilkinson, P., Harwood, A. & Novak, M. **Easy Approach to Requirements Syntax (EARS**). in *2009 17th IEEE International Requirements Engineering Conference* 317–322 (IEEE, 2009). doi:10.1109/RE.2009.9
6. Jayaratna, N. *Understanding and Evaluating Methodologies: NIMSAD - **A Systemic Framework**. Information systems, Management and Strategy series* 47, (Palgrave Macmillan UK, 1994).
7. Isa, M. A., Zaki, M. Z. M. & Jawawi, D. N. A. **A Survey of Design Model for Quality Analysis**: From a Performance and Reliability Perspective. *Comput. Inf. Sci.* 6, 55–70 (2013). https://doi.org/10.5539/cis.v6n2p55
8. Mavin, A. & Wilkinson, P. Big Ears **(The Return of &quot;Easy Approach to Requirements Engineering&quot;**). in *2010 18th IEEE International Requirements Engineering Conference* 277–282 (IEEE, 2010). https://doi.org/10.1109/RE.2010.39
9. Daramola, O., Sindre, G. & Stalhane, T. **Pattern-based security requirements specification using ontologies and boilerplate**s. in *2012 Second IEEE International Workshop on Requirements Patterns (RePa)* 54–59 (IEEE, 2012). https://doi.org/10.1109/RePa.2012.6359973
10. Ibrahim, N., Wan Kadir, W. M. N. & Deris, S. **Documenting requirements specifications using natural language requirements boilerplates**. in *2014 8th. Malaysian Software Engineering Conference (MySEC)* 19–24 (IEEE, 2014). https://doi.org/10.1109/MySec.2014.6985983