Non-functional Requirements (NFR) Identification Method using FR Characters based on ISO/IEC 25023

Nurbojatmiko*, Eko K. Budiardjo, Wahyu C. Wibowo
Faculty of Computer Science, Universitas Indonesia
Kampus Baru UI Depok, Pondok Cina, Beji
Depok, West Java, Indonesia16424

***Abstract*—The researches show that software quality depends on Functional Requirements (FR) and Non-Functional Requirements (NFR). The developers identify NFR attributes by interviewing stakeholders. The difficulty in identifying NFR attributes makes quality requirements often ignored. The basic concept of software quality measurements is the quality measuring of the software product. During product-based quality measurement, the potential of software development process repetition will occur. Factors measuring software product quality are not suitable for NFR identification. These differences result in the software development process repeating itself and additional costs. This research proposes easy NFR attributes identification using FR characters. The NFR and FR tightly relations are obtained by extending the NFR measurement at ISO/IEC25023 to programming coding level, then generalizing to get the FR character. The generalization uses the Grounded Theory method. The result is the NFR attributes identification method using FR character based on ISO/IEC 25023. The analyst or programmer can identify the NFR attributes from the FR using the FR character in the requirements stage. This research produces an NFR Identification Method that has been validated by experimenting with several programmers and experts. Tests on programmers identify NFR using the FR character method. The test is to see the level of similarity of the resulting NFR. The result of the test shows level similarity upper 75%.**

***Keywords*—Non functional requirements; FR character; ISO/IEC 25023; NFR identification**

I. Introduction

Software development success requires quality measurement results of the software product. The Software quality measurements base ISO/IEC 25023 is a complete software product quality measurements [1]. The quality classification of the software in ISO/IEC 9126 [2] updated ISO/IEC 25023 [1] is the NFR attributes classification. NFR attributes identification awareness determines the quality of software products. NFR Identification affects the resulting software product [3], [4]. Failure of NFR attributes identification may be repeatable in the software development process [5], [6]. The results of the NFR attributes identification determine the success of the software product [7], [8]. This paper shows that awareness for the NFR attributes identification is essential. The problem of improper NFR identification causes dissatisfaction with product quality, resulting in repetition of the software development process and increasing costs.

The broad meaning of quality towards software products has prompted several studies to classify NFR. Several researchers have solved the problem of NFR identification using the NFR Classification [5] [9]. Problems arise again in determining an unambiguous NFR classification [10]. Developers based on the Agile method need NFR identification quickly and precisely. The problem software developments are NFR attributes identification suitable with software quality measurements. This research uses ISO/IEC 25023 as a basis so that the process of quality identification and measurement has the same reference. How to NFR attributes identification use ISO/IEC 25023?.

This research aims to develop ISO/IEC 25023 for NFR attributes identification. This Research in ISO/IEC 25023 extended the measurement function to the programming coding level. The generalization programming code to get FR character use grounded theory. The research is NFR attributes and FR character tightly relations. The result of this research is the NFR attributes identification method using FR character. This research on the NFR identification process uses FR characters based on ISO/IEC 25023. NFR attributes identification testing uses the FR character on several programmers. The result of the test is to determine the level of similarity of the NFR attributes obtained above 75%. This research method is open coding stage, axial data stage, selective coding stage, forming theory stage, and memoing. Writing systematic of this paper is abstract, introduction, related works, research method (the generalization use grounded theory), discuss, conclusion, acknowledgments, and reference. The grounded theory method consists of stages open coding, axial coding, selective coding, and forming theory. Forming of theory determines NFR attributes identification formulation method and NFR attributes method testing.

II. Related Work

Yusop identifies NFR attributes used resulting qualitative research with an interview from 5 developers. So, the qualitative research result is NFR attribute classification [11]. Sharma, problem-solving the NFR attributes identification uses automatization detection. The Algoritma automation

*Corresponding Author

detection uses NFR attributes categorization and classification based on semantic patterns [6]. Singh use ISO/IEC 9126 standard for NFR attributes categorization dan classification [12]. Li uses the quality specification of domain and subdomain for NFR attributes classification [13]. Chung classifies NFR attributes from several studies literature such as RADC (Rome Air Development Center), Sommerville, Mac Calls, Matsumoto, Grady, and others. Some studies classify NFR based on software products [5]. Kaur develops integration of the NFR attributes and formal reference model for the NFR classification [14]. NFR attributes classification uses quantitative NFR with questioners for minimizing ambiguity [15].

Chung presents a goal-object pattern framework. That framework uses a model-driven by way of UML metamodel extension. The framework is capturing and reusing FR and NFR knowledge of the small-scale application [16]. Singh identifies NFR attributes using NFR attributes classification with different thematic roles based on ISO 9126 quality factors [12]. Kassab makes metamodel by tracing NFR attributes and FR relation based on strong interdependencies [17]. Farid identifies NFR attributes, its using risk-driven algorithms to the prioritization scheme [18]. Liu develops automatization to detect conflict of the NFR attributes evaluation. NFR classification approach uses ontology realizing with metamodel based on cause-effect and inferences knowledge [19].

The determination of the NFR classification so as not to be confused needs to be standardized. ISO/IEC issued standards for the quality classification and measurement of product quality in ISO/IEC9126 [2]. ISO/IEC9126 updated by ISO/IEC 25023 [1]. The NFR classification according to ISO / IEC 25023 is functionality suitability, performance efficiency, compatibility, usability, reliability, security, maintenance, and portability [1].

Several researchers made improvements to the method stages of the Agile Method to be able to identify NFR. Lawrence Chung uses approaches of modeling and techniques to explain software requirements.[20]. Farid, Agile processes use a risk priority approach with the Non-functional Requirements Planning (NORPLAN) method [18]. Domah uses the NERV methodology to obtain NFR artifacts on the User Story Card, the NAI (NERV Agility Index) score, and so on [21].

III. Research Method

This research generalization process of NFR attributes classification uses the Delon-style Grounded Theory method (Fig. 1) [22]. The first stage is open coding. The open coding stage is collecting and identifying data. The data is from the results of the SLR [23], updating papers, and ISO/IEC 25023 files. The results in this stage are NFR attributes measurement identification and classification (Table I) based on ISO / IEC 25023 standards. The Second stage is axial data. This stage is marking or tagging the measurement functions with programming codes. The result axial coding stage is the classification of NFR attribute measurements and the extension of the measurement function to code programming characteristics (Table II). The third stage is selective coding. The stage is the selecting and comparing of data programming code characteristics, it is the generalization process. The process of generalization results NFR attributes and FR characters relationship (Table III). The fourth is forming theory. This stage is the formation theory to NFR attributes and FR characters relation. The forming theory stage is formulation to NFR identification method (Fig. 2) and testing result (Fig. 3). The Fifth stage is memoing. The memoing stage record of this research to a paper journal.

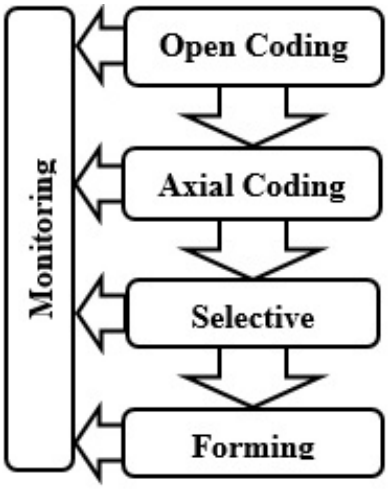

Fig. 1. Delon-style Grounded Theory [22].

IV. Results

A. Open Coding

Referring to SLA takes the condition of the established attributes NFR. That refers standard of software quality. This research uses ISO/IEC 25023 as an international standard product quality software. The measuring factors for software product quality in ISO/IEC 25023 are NFR attributes [24].

ISO/IEC 25023 is a software product quality standard. The elements of quality measures determine measurement functions. Software quality measurement in this way can determine the quantification, characteristics, and sub-characteristics of quality. System and software product quality programs explain that quality measures follow quality characteristics in evaluating internally or externally.

The result of quality measurement identification has eight NFR attributes. The eight NFR attributes in ISO/IEC 25023 are functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, portability. Each attribute is classified and specified into several measurement functions. The ISO/IEC 25023 file on page 10 is an example of measuring performance on Time behavior with specifications for measuring mean response time, response time adequacy, mean turnaround time, and turnaround time. The measurement function has measures the quality of the software product and characteristics of quality. The result of quality measurement identification is the example in Table I.

TABLE I. NFR ATTRIBUTE MEASUREMENT FUNCTION BASED ON ISO/IEC25023 (EXAMPLE: FUNCTIONAL SUITABILITY AND PART OF PERFORMANCE EFFICIENCY)

Specifications	Sub-specifications	Measurement Function	
		Measurements	**Measurement Characteristics**
NFR1. Functional Suitability			
NFR1.1. F. Completeness	NFR1.1.1. F. Completeness	What proportion of the specified functions has been implemented?	1) All FR has boolean value (yes/no=1/0) 2) The Unit of Number FR 3) Number of FR codes that have been made 4) Number of FR encodings that work 5) The number of FR accordance with the function of the actor 6) The number of FR has conformity to the system (average for point (5))
NFR1.2. F. Correctness	NFR1.2.1. F. Correctness	What proportion of functions provides the correct results?	
NFR1.3. F. Appropriateness (FA)	NFR1.3.1. FA of usage objective	What proportion of the functions required by the user provides appropriate outcomes to achieve a specific usage objective?	
	NFR1.3.2. FA of system	What proportion of the functions required by the user to achieve their objectives provides appropriate outcomes?	
NFR2. Performance Efficiency			
NFR2.1. Time Behaviour	NFR2.1.1. Mean response time	How long is the mean time taken by the system to respond to a user task or system task?	1) Unit of time is ms (millisecond) 2) Response time is the time of the page to page 3) Average of response time
	NFR2.1.2. Response Time Adequate	How well does the system response time meet the specified target?	
	NFR2.1.3. Mean turnaround time	What is the meantime taken for the completion of a job or an asynchronous process?	1)Unit of Time is ms (millisecond) 2) Turnaround time is the time of process to one task or process to asynchronous until the finish 3) Average of turnaround time
	NFR2.1.4. Turnaround Time	How well does the turnaround time meet the specified targets?	
	NFR2.1.5. Mean Throughput Time	What is the mean number of jobs completed per unit time?	1) This unit is the number of data transfers per time (number of data every ms) 2) Throughput time is the time needed to start transferring some data to completion to the destination 3) Average throughput time
NFR2.2. Resource Utilization	NFR2.2.1. Mean processor utilization	How much processor time is used to execute a given set of tasks compared to the operation time?	1) The unit of time, ms (millisecond) 2) Processor time is time used to execute a given set of tasks 3) Operation time to perform the tasks 4) Time required for 1 task to perform arithmetic OR logic functions 5) Mean processor time
	NFR2.2.2. Mean memory utilization	How much memory is used to execute a given set of tasks compared to the available memory?	1) Unit size of memory (byte) 2) Size of memory used to perform series of task 3) Average for point (2) 4) Size of memory available
	NFR2.2.3. Mean I/O device utilization	How much of the I/O device's busy time is used to perform a given set of tasks compared to the I/O operation time?	1) Unit of time (ms) 2) The time of I/O device used to perform a series of task 3) Time of I/O operation
	NFR2.2.4. Bandwidth utilization	What proportion of the available bandwidth is utilized to perform a given set of tasks?	1) Unit of bandwidth (bps/bits per second) 2) Size of data (byte) carried to perform a series of tasks per time (second)
NFR2.3. Capacity	NFR2.3.1. Transaction processing capacity	How many transactions can be processed per unit time?	1) Unit number of transaction processes for per time (second) 2) Average of point (1) 3) Transactions related to record in the database (Create, Read, Update, Delete)
	NFR2.3.2. User access capacity	How many users can access the system simultaneously at a certain time?	1) Maximum number of users at the same time 2) Unit number of user
	NFR2.3.3. User access increase adequacy	How many users can be added successfully per unit time?	1) Unit number of users per time 2) Acceleration of user growth

TABLE II. EXTENDING FROM MEASUREMENT FUNCTION TO PROGRAMMING CODE (EXAMPLE: PERFORMANCE EFFICIENCY OF THE TIME BEHAVIOR SPECIFICATION)

ID_NFR	Measurement Function		
	Measurements	**Measurement Characteristics**	**Programming Code**
NFR2. Performance Efficiency			
NFR2.1.1.	How long the meantime took by the system to respond to a user task or system task is?	1) Unit of time is ms (millisecond) 2) Response time is the time of the page to page 3) Average of response time	Link, submit, download, upload, back, next
NFR2.1.2.	How well does the system response time meet the specified target?		
NFR2.1.3.	What is the meantime taken for the completion of a job or an asynchronous process?	1)Unit of Time is ms (millisecond) 2) Turnaround time is a time of process to one task or process to asynchronous until the finish 3) Average of turnaround time	Proses CRUD in the database (Create, Read, Update, Delete)
NFR2.1.4.	How well does the turnaround time meet the specified targets?		
NFR2.1.5.	What is the mean number of jobs completed per unit time?	1) This unit is the number of data transfers per time (number of data every ms) 2) Throughput time is the time needed to start transferring some data to completion to the destination 3) Average throughput time	Sum of data per unit time for the transfer process to or from the database (CRUD)
NFR2.2.1.	How much processor time is used to execute a given set of tasks compared to the operation time?	1) The unit of time, ms (millisecond) 2) Processor time is time used to execute a given set of tasks 3) Operation time to perform the tasks 4) Time required for 1 task to perform arithmetic OR logic functions 5) Mean processor time	1) Process of logic and arithmetic functions 2) Process of mathematics operation (+, -, /, *) 3) Process of logic (<, >, = =, <=, >=) 4) Process of condition functions (IF, FOR, While, Switch - Case, Do – While)
NFR2.2.2.	How much memory is used to execute a given set of tasks compared to the available memory?	1) Unit size of memory (byte) 2) Size of memory used to perform series of task 3) Average for point (2) 4) Size of memory available	Functions of variable OR declaration
NFR2.2.3.	How much of the I/O device's busy time is used to perform a given set of tasks compared to the I/O operation time?	1) Unit of time (ms) 2) The time of I/O device used to perform a series of task 3) Time of I/O operation	Time used to I/O operation (example, process to print, download(curl_setopt), upload(fungsi; input type="file") and scan functional view to report, tranfering data network)
NFR2.2.4.	What proportion of the available bandwidth is utilized to perform a given set of tasks?	1) Unit of bandwidth (bps/bits per second) 2) Size of data (byte) carried to perform a series of tasks per time (second)	Transmission (variable array)
NFR2.3.1.	How many transactions can be processed per unit time?	1) Unit number of transaction processes for per time (second) 2) Average of point (1) 3) Transactions related to record in the database (Create, Read, Update, Delete)	The Number of CRUD function process in the database
NFR2.3.2.	How many users can access the system simultaneously at a certain time?	1) Maximum number of users at the same time 2) Unit number of user	The active user name function
NFR2.3.3.	How many users can be added successfully per unit time?	1) Unit number of users per time 2) Acceleration of user growth	

B. Axial Coding

This stage performs the preparation of NFR measurements and decreases from measurement function to programming coding. The composition of the software quality measurement is the arrangement of the NFR attributes in Table I. This research extends from measurement functions to programming code. This measurement results in a more precise measurement. That NFR measurement determines FR and NFR relations. The fragment of an ISO / IEC 25023 file of Time Behavior Measures in the Measurement Function column contains the measurement algorithm for that quality. For example, the measurement algorithm for response time adequacy on the performance efficiency attribute says Ai = the time it takes the system to respond. The time needed by the system to respond has the meaning of time when the user clicks on a function until the function appears. Simplifies, it is the span of the time from one page to the next page. The time has a unit of time in ms (millisecond). The time-span from one page to another on the measurement translates in the code programming, namely link, submit, download, upload, back, next in the code programming column with the id number NFR2.1.1 in Table II. The result of this stage extends from the measurement function to the programming code, as shown in Table II.

TABLE III. FUNCTION CHARACTER OF NFR ATTRIBUTE BASED ON ISO/IEC 25023

No	NFR Attributes	Function Characters
1	**Functional Suitability**	The function runs according to the actor
2	**Performance Efficiency**	CRUD (Create, Read, Update, Delete)/ database Process
3	**Compatibility**	Files Transfer
4	**Usability**	Make it easier for users and reduce human error
5	**Reliability Measures**	Failure/error detection in the system
6	**Security**	Security at the time of transfer, data, access rights, and control
7	**Maintainability**	Module-related functions (reusability, log, analysis, modification, testability)
8	**Portability**	Configuring apps; Functions to adapt to other environments/applications/software (installations, products)

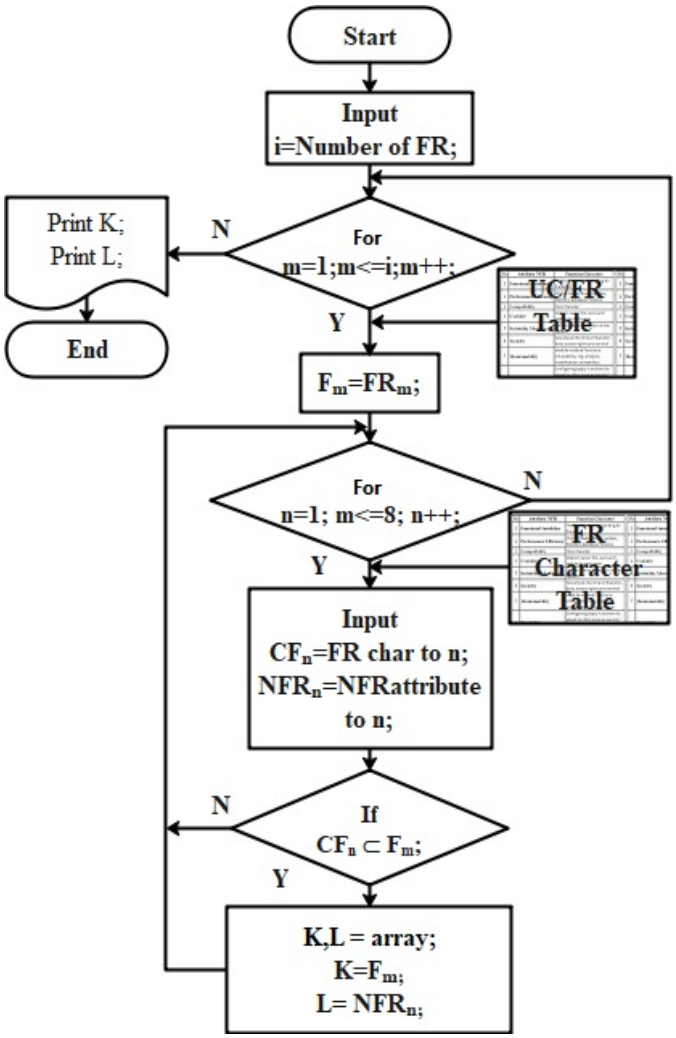

Fig. 2. Pseudo Code NFR Identification use FR Character Method.

C. Selecting Coding

The research takes as an example the NFR attributes of performance efficiency (Table II) with the behavior of time, resource utilization, and capacity specifications. The table contains the ID_NFR, measurements, measurement characteristics, and programming code columns.

Comparison of 2 is the process of CRUD with the process of arithmetic and logic in line 4 on ID_NFR 2.1.4. The CRUD process follows some arithmetic and logical process more than others. The arithmetic and logic processes for measuring processor speed followed by the CRUD process significantly affect processes that are not followed by the CRUD process because there is a process for bringing extensive data from the database. The arithmetic and logic processes that need to be measured are those that the CRUD process follows.

Comparison of 3 is the process of CRUD with rows 5,6,7 and 9, on ID_NFR 2.2.1, ID_NFR 2.2.2, ID_NFR 2.2.3, ID_NFR 2.2.5. These rows have a profound effect on the process the CRUD process follows. That it can ignore another process because the process represents that the CRUD process follows. Performance efficiency measurement for FR concludes that all FRs have a function for the CRUD process. The data comparison results show in Table III, namely that FR's characters have NFR attributes.

D. Forming Theory

The generalization results in Table III show that each FR character has an NFR attribute. NFR is highly dependent on the FR. The determination of the NFR attribute is from the character content in the FR. NFR has a relationship with FR. Research shows that NFR has tightly coupled with FR. FR has more than one character means it has more than one NFR attribute. Each NFR attribute can be on multiple FRs. The relationship between FR and NFR has cardinality, many to many, meaning that FR has more than one NFR, and conversely, NFR has more than one FR. Identification of quality can be known early based on the FR obtained.

The FR character is the result of the generalization from the measurement function, as shown in Table III. NFR attributes can be detected quickly and accurately against FR by an analyst at the requirements stage. Early identification of NFR attributes can monitor the quality of functions during the development process.

1) NFR attributes identification method formulation: The requirements stage is the identification process of FR and NFR. The determination of the NFR attribute at that stage is after the FR determination. Determination of NFR attributes using the FR character (Table III) against the FR. Each FR will derive NFR attributes based on the characters it contains. Further research formulates a method for the identification of NFR attributes using the FR character. The FR from Use Case Specification (UC Spec) or the FR table is the input. Each FR determines the content of FR characters in Table III to obtain NFR attributes. The result is that FR is related to NFR attributes. Fig. 2 is the NFR identification method algorithm using the FR character.

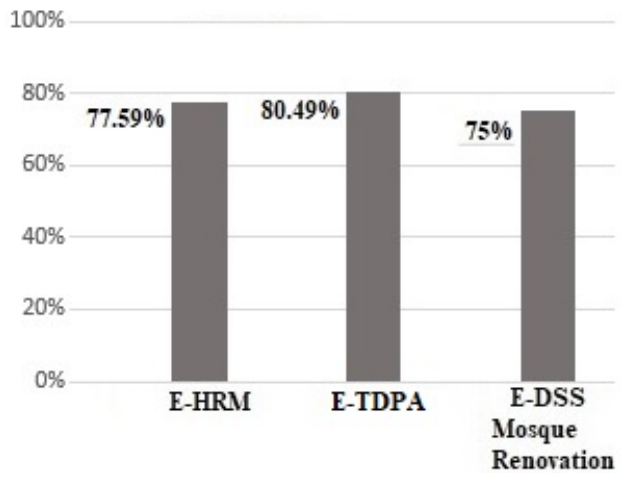

Fig. 3. The Result of the Similarity Level of NFR Attributes Identification.

2) NFR attributes identification method testing: Testing the NFR attribute identification method to determine the level of similarity of the programmers to determine the NFR attributes to their FR. The testing used the samples of 3 web applications based on PHP and MySQL, namely e-HRM, electronic Tax Dispute Power of Attorney (e_TDPA), Electronic DSS Renovation of Indonesian Mosques (e-DSS Mosques Renovation). The study identified FR in all three applications according to its SRS. The research identified the FR of the SRS documents in each application, then arranged them in the FR table. The programmer uses the FR character table guide to fill in the NFR attributes contained in each FR. The research uses the samples of 5 programmers. The test results show the similarity level of NFR attributes above 75% (Fig. 3). These results indicate that the programmers have a high common perception of NFR attributes.

V. Discussion

Pratama determined software quality using ISO/IEC 25010. He assessed software quality of 8 attributes using Black-Box testing, stress testing, and questioner of the 100 respondents. And then leveling of the 8 attributes used AHP [25]. This research for software quality using ISO/IEC 25023 standard of software quality product. That standard measure to suitable between software product quality with requirements. ISO/IEC 25010 is the family of 250n standards. ISO/IEC 25010 is measurements to software quality model and ISO/IEC 25023 is measurements to software product quality. Yusop classified NFR attributes based on qualitative research results [11]. Sharma classified NFR attributes only performance, availability, and security is not using an international standard, Li used quality specifications into domain and subdomain [13], and Singh used ISO/IEC 9126 [6], [26]. Kaur classified NFR attributes with NFR attributes and formal reference relations. The Formal References are domain knowledge, customer requirements, specification, programming platform, and machine [14]. This research NFR attributes categorization and classification use ISO/IEC 25023 that is updating from ISO/IEC 9126. Then, the research develops NFR attributes categorization and classification to get NFR attributes and FR characters integration.

Chung identified the requirements to use the objective-object pattern of FR and NFR relations. The knowledge patterns have format from the experience of several applications samples [16]. Farid used a risk-driven algorithm for NFR attributes identification [18]. Liu detected NFR attribute conflict using ontology realization. This research NFR attributes identification used the FR characters approach. Kassab used an understanding of NFR attributes and FR relationships for the detection of NFR attributes [17]. This research uses NFR attributes and FR relation with FR characters approach. Research identifies NFR attributes based on the character from FR. Each FR has characters. FR character comes from NFR attributes categorization and classification based on ISO/IEC 25023 extend. NFR attribute identification and software quality measurements have the same based on ISO/IEC 25023. The result of software quality measurements suite requirements.

VI. Conclusion

ISO/IEC 25023 is a standard for measuring software product quality. This research succeeded in constructing the FR character from ISO/IEC25023 for the NFR Identification Method. The FR character to identify the NFR attribute of each FR that has the characters. FR character and measurement of quality software products have the same basis, namely ISO/IEC 25023. Identification of NFR attributes using FR character will produce NFR attributes following the desired quality software product.

FR character is a bridge that connects FR with NFR quality or attributes. NFR and FR are relations that both have tightly coupled. Stakeholders are very helpful in determining NFR attributes without having to interpret the type of software quality. Programmers can control the quality of built-in functions while coding these functions. Product software quality is in line with the same NFR attributes based on ISO/IEC 25023. Quality control of software before it becomes a product avoids repeating the software development process and adding costs. Identifying the right NFR attributes determines a quality software product. The future research monitors software quality using FR characters in the Scrum software development method.

Acknowledgment

The funder for this research was from the PUTI Grant the Universitas Indonesia with the number NKB-559/UN2.RST/HKP.05.00/2020.

References

[1] ISO/IEC25023, “Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality,” Int. Stand. ISO / IEC 25023, vol. 2016, no. First edition, pp. 1–54, 2016.

[2] ISO/IEC9126, “International Standard ISO / IEC 9126 Software Engineering — Product quality ISO 9126 - Content,” Int. Stand. ISO/IEC 9126, Softw. Eng. Qual., pp. 1–8, 2001.

[3] H. Kaur and A. Sharma, “A Measure for Modelling Non-Functional Requirements Using Extended Use Case,” Proc. 10th INDIACom; 2016 3rd Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2016, pp. 1101–1105, 2016.

[4] A. Alwadi, A. Nahhas, S. Bosse, N. Jamous, and K. Turowski, "Toward a performance requirements model for the early design phase of IT systems," Proc. - 2018 6th Int. Conf. Enterp. Syst. ES 2018, pp. 9–16, 2018.

[5] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, NON-FUNCTIONAL REQUIREMENTS IN SOFTWARE ENGINEERING, 1st ed. New York: SPRINGER SCIENCE+BUSINESS MEDIA, LLC, New York, 2000.

[6] V. S. Sharma, R. R. Ramnani, and S. Sengupta, "A Framework for Identifying and Analyzing Non-functional Requirements from Text Categories and Subject Descriptors," TwinPeaks, Hyderabad, India, June 1, 2014, vol. 14, pp. 1–8, 2014.

[7] M. Iqbal, "NFR Modeling Approaches," Proc. - 1st ACIS Int. Symp. Softw. Netw. Eng. SSNE 2011, pp. 109–114, 2011.

[8] M. Ahmad, N. Belloir, and J. Bruel, "Modeling and verification of Functional and Non-Functional Requirements of ambient Self-Adaptive Systems," J. Syst. Softw., vol. 107, pp. 50–70, 2015.

[9] R. Gomes and N. Bencomo, "On Modeling and Satisfaction of Non-Functional Requirements using Cloud Computing," IEEE, pp. 1–6, 2013.

[10] I. M. S. Raharja and D. O. Siahaan, "Classification of Non-Functional Requirements Using Fuzzy Similarity KNN Based on ISO / IEC 25010," Int. Conf. Inf. Commun. Technol. Syst., vol. 12, pp. 264–269, 2019.

[11] N. Yusop, D. Zowghi, and D. Lowe, "The impacts of non-functional requirements in web system projects," Proc. Eur. Mediterr. Conf. Inf. Syst. EMCIS 2006, 2006.

[12] P. Singh, D. Singh, and A. Sharma, "Rule-Based System for Automated Classification of Non-Functional Requirements from Requirement Specifications," Intl. Conf. Adv. Comput. Commun. Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India Rule-Based, pp. 620–626, 2016.

[13] F. Li et al., "Non-functional Requirements as Qualities , with a Spice of Ontology," IEEE RE 2014, Karlskrona, Sweden, pp. 293–302, 2014.

[14] H. Kaur and A. Sharma, "Use of Reference Model for Formal Specification of NFR," India Int. Conf. Inf. Process. IICIP 2016 - Proc., pp. 1–3, 2017.

[15] K. I. Gómez Sotelo, C. Baron, P. Esteban, C. Y. A. G. Estrada, and L. de J. Laredo Velázquez, "How to Find Non-Functional Requirements in System Developments," IFAC-PapersOnLine, vol. 51, no. 11, pp. 1573–1578, 2018.

[16] L. Chung and S. Supakkul, "Capturing and Reusing Functional and Non-Functional Requirements Knowledge: A Goal-Object Pattern Approach," Proc. 2006 IEEE Int. Conf. Inf. Reuse Integr. IRI-2006, pp. 539–544, 2006.

[17] M. Kassab, O. Ormandjieva, and M. Daneva, "A Metamodel for Tracing Non-Functional Requirements," World Congr. Comput. Sci. Inf. Eng., pp. 687–694, 2009.

[18] W. M. Farid and F. J. Mitropoulos, "NORPLAN : Non-functional Requirements Planning for Agile Processes," IEEE, 2013.

[19] C. Liu, "CDNFRE : Con fl ict detector in non-functional requirement evolution based on ontologies," Comput. Stand. Interfaces, vol. 47, pp. 62–76, 2016.

[20] L. Chung, M. Noguera, and M. L. Rodríguez, "An Agile Requirements Elicitation Approach based on NFRs and Business Process Models for Micro- Businesses," Profes , TORRE CANNE (BR), Italy. June 20 - 22, 2011, vol. 11, pp. 50–56, 2011.

[21] D. Domah and F. J. Mitropoulos, "The NERV Methodology : A Lightweight Process for Addressing Non-functional Requirements in Agile Software Development," Proc. IEEE SoutheastCon 2015, April 9 12, 2015 Fort Lauderdale, Florida, 2015.

[22] S. Chandrasegaran, S. K. Badam, L. Kisselburgh, K. Ramani, and N. Elmqvist, "Integrating Visual Analytics Support for Grounded Theory Practice in Qualitative Text Analysis," Comput. Graph. Forum, vol. 36, no. 3, pp. 201–212, 2017.

[23] Nurbojatmiko, E. K. Budiardjo, and W. C. Wibowo, "SLR on identification & classification of non-functional requirements attributes, and its representation in functional requirements," ACM Int. Conf. Proceeding Ser., no. December, pp. 151–157, 2018.

[24] O. Gordieiev, V. Kharchenko, and M. Fusani, "Software Quality Standards and Models Evolution: Greenness and Reliability Issues," ICTERI 2015, CCIS 594-Communications Comput. Inf. Sci., vol. 594, pp. 38–55, 2016.

[25] A. A. Pratama and A. B. Mutiara, "Software Quality Analysis for Halodoc Application using ISO 25010:2011," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 8, pp. 383–392, 2021.

[26] P. Singh, D. Singh, and A. Sharma, "Rule-Based System for Automated Classification of Non-Functional Requirements from Requirement Specifications," Intl. Conf. Adv. Comput. Commun. Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India, pp. 620–626, 2016.