A Generic Method for Identifying Maintainability Requirements Using ISO Standards

Khalid T. Al-Sarayreh
Hashemite University
Software Engineering Department
Zarqa, 13133, Jordan
P.O.Box 33127,00962-798471991
khalidt@hu.edu.jo

Asma Labadi
Hashemite University
Software Engineering Department
Zarqa, 13133, Jordan
P.O.Box 33127,00962-777179808
asma.labadi@itc.hu.edu.jo

Kenza Meridji
University of Petra
Software Engineering Department
Amman, 11196, Jordan
P.O.Box 33127,00962-791996084
kmeridji@uop.edu.jo

ABSTRACT
ISO 9126 standards for the software product quality; includes maintainability requirements for the software Quality. The software maintainability requirements can be measured internally and externally; According to these standards maintainability requirements shall be apportioned to set of maintainability requirements for lower level products to conform to the maintenance concept, maintainability requirements of the system and the maintainability analysis shall identify maintainability critical items. This paper presents a method for identifying the maintainability requirements. The proposed method suggesting a generic model to locate and measure all maintainability requirements as defined in ISO.

Categories and Subject Descriptors
Knowledge Engineering, Information and Communication for developing, Technologies as Teaching Strategy and Learning Styles.

General Terms
Measurement, Design, Standardization.

Keywords
Maintainability Requirements, Design of Measurement Method, ISO International Standards.

1. INTRODUCTION
According to European Standards the maintenance process contains the activities and tasks of the maintainer. The objective is to modify an existing software product while preserving its integrity. This process includes the migration and retirement of the software product. The process ends with the retirement of the software product.

The maintainer manages the maintenance process at the project level following the management process, which is instantiated for software in this process. This process consists of the following activities: (process implementation, problem and modification analysis, modification implementation, conducting maintenance reviews and software migration) [1], [2] and [3].

Maintainability requirements shall be apportioned to set maintainability requirements for lower level products to conform to the maintenance concept and maintainability requirements of the system and the maintainability analysis shall identify maintainability critical items [2].

ISO 24765 [4] define the maintainability as the ease with which a software system or component can be modified to change or add capabilities, correct faults or defects, improve performance or other attributes, or adapt to a changed environment

While, IEEE 14764 [5] defines maintainability as the capability of the software product to be modified and maintainability is the speed and ease with which a program can be corrected or changed.

ISO 9126 [6] defines the maintainability as a capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

The paper will report a design measurement method to identify the functional size of the software maintainability which avoids the weaknesses observed in the maintenance measures currently available.

The measurement scope in this paper is to identify separately the all functionality allocated to software maintainability as a piece of the application in the Requirements for embedded and real time software.

Furthermore, the main contribution of this paper is the proposed a reference procedural method for maintainability requirements. The proposed generic method is considered as kind of a reference model in the sense of an 'etalon' standard that is being used for the measurement of maintainability.

This paper is organized as follows. Section 2 presents the related works. Section 3 presents the steps for designing a measurement method for maintainability as defined in ISO. Finally discussion and a conclusion are presented in section 4.

2. RELATED WORK
There are many early works on maintainability in the previous work without attention for such maintainability requirements in details, for instance, Antonellis et al.[7] proposed a model for mapping the object oriented code metrics on the maintainability characteristics based on ISO 9126.

Moreover, Chidamber and Kemerer [8] proposed metrics suite to propose weights for each metrics sub characteristics that can be used in maintainability measuring, in Broy et al. [9] developed a

IPAC '15, November 23-25, 2015, Batna, Algeria
 ISBN 978-1-4503-3458-7/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2816839.2816929

model for measuring maintainability activities based on IEEE 1219 maintenance process standards.

For instance, Oman et al. [10] proposed a hierarchal model structure for measuring maintainability attributes based on 35 publications reviews including not just the source code but also metrics for defects, changes.

Furthermore, Halstead [11] proposed a metrics to measure the software attributes that can have an effect on the maintenance while McCabe [12] combine the aforementioned metrics in [11] into one metric called maintainability index to determine weighs in statistics way.

In Aggarwal [13] introduced a model for measuring software maintainability using the rule base for the fuzzy models. The main attributes that are taking into consideration are: RSC (Software Readability of source code), DOQ (Documentation Quality), and UOS (understandability of software).

Finally Deissenboeck et.al [14] proposed a quality model taking into account the maintenance activities using two dimensional metrics for the maintainability activities and the other dimension for the system properties that are related to them. They performed a case study for evaluating the proposed model in the model-based development of embedded systems.

The motivation of this research paper is to contribute to better define, describe and measure the maintainability requirements as input required for adequate estimation of software projects. The measurement scope in this paper is to identify separately all functionality allocated to software maintainability requirements for software product quality as defined in ISO.

The focus of this paper is on a single type of NFR that is, maintainability requirements. This paper reports on the work carried out to define an integrated view of software functional user requirements for maintainability requirements for the software product on the basis of ISO international standards for more details see [15], [16], [17] and [18].

3. DESIGN MAINTAINABILITY METHOD

Based on the maintainability requirements definitions stated by ISO standards the design measures steps will be as follows:

3.1 Measurement Objectives

This section illustrates the objectives of maintenance, followed by the measurement point of view and the intended uses of the measurement results.

- The objective: is to measure the size of the maintainability requirements as defined in ISO.
- Measurement point of view: Software perspective.
- Intended uses of the measurement results: throughout the software life cycle: the size of the maintainability for a software product, whether it has yet to be built or it has already been delivered.

3.2 Main Measurements Concepts

This section illustrates the the measurable concept to be defined into a measurable construct and from it the measurement method to be built, the concept to be measured must be clearly defined.

- The functional size of maintainability internally or externally.
- External maintainability Measures**:** should be able to measure such attributes as the behaviour of the maintainer, user, or system including the software, when the software i**s** maintained or modified during testing or maintenance**.**
- Internal maintainability Measures: are used for predicting the level of effort required for modifying the software product.

The external maintainability entities to be measured

- Analysability: should be able to measure such attributes as the maintainer's or user's effort or spent of resources when trying to diagnose, deficiencies or causes of failures, or for identifying parts to be modified, audit trial capability, failure analysis capability and status monitoring capability
- Changeability: should be able to measure such attributes as the maintainer's or user's effort by measuring the behaviour of the maintainer, user or system including the software when trying to implement a specified modification, change efficiency and software change control capability
- Stability metrics: should be able to measure attributes related to unexpected behaviour of the system including the software when software is tested or operated after modification, Change Success Ratio
- Testability metrics: should be able to measure such attributes as the maintainer's or user's effort by measuring the behaviour of the maintainer, user or system including software when trying to test the modified or non-modified software, availability of built in test function and re-test efficiency

The internal maintainability entities to be measured

- Analysability : indicate a set of attributes for predicting the maintainer's or user's spent effort or spent resources in trying to diagnose for deficiencies or causes of failure, or for identification of parts to be modified in the software product (Diagnostic Function Support).
- Changeability: indicate a set of attributes for predicting the maintainer's or user's spent effort when trying to implement a specified modification in the software product (Modifiability).
- Stability: indicates a set of attributes for predicting how stable the software product would be after any modification (Modification Impact**)**
- Testability: indicates a set of attributes for predicting the amount of designed and implemented autonomous test aid functions present in the software product (Test Restart ability**).**

3.3 A procedural Method for Maintainability

This section presents the model of the software maintainability on the basis of the previous section.

3.3.1 Analyzability

In the following design of the Meta models- see Figure 1:

- Entity type 1 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to resolve failure for one functional process.
- Entity type 2 can be used to measure the functional size of the internal software maintainability throughout observe behaviour of user-developer-maintainer to resolve failure using diagnostic functions.
- Entity type 3 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to resolve failure using diagnostic functions.

- Entity type 4 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to get monitored data recording status during operations for one functional process.

3.3.2 Changeability

In the following design of the Meta models- see Figure 2:

- Entity type 5 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to change the software for one functional process.
- Entity type 6 can be used to measure the functional size of the internal software maintainability throughout observe

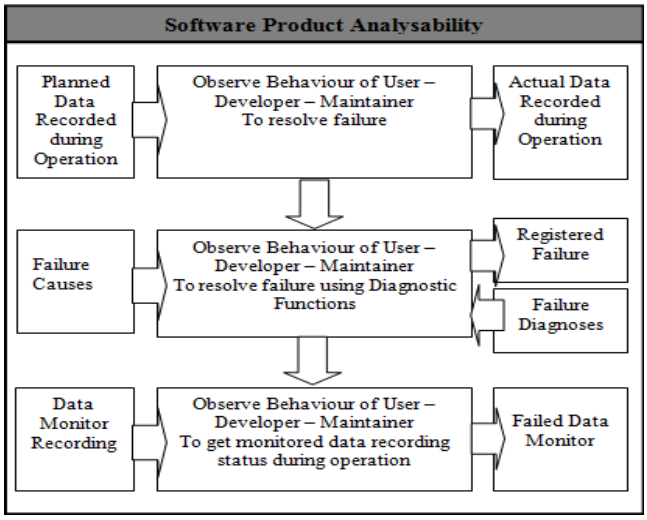

Fig. 1: Generic of FSM Model of Software Product Analyzability

behaviour of user-developer-maintainer to investigate the software problems for one functional process.

- Entity type 7 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to investigate the software problems for one functional process.

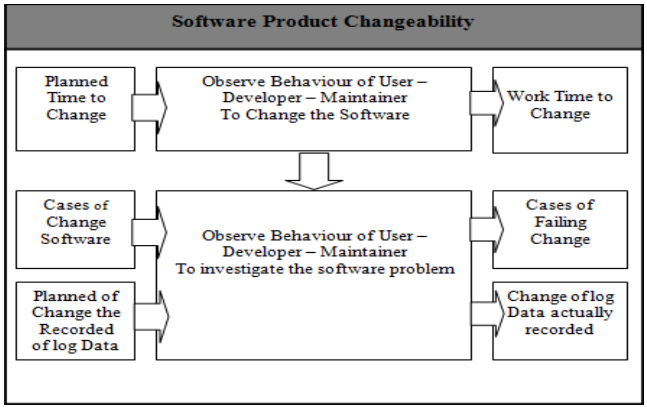

Fig. 2: Generic of FSM Model of Software Product Changeability

3.3.3 Stability

In the following design of the Meta models- see Figure 3:

- Entity type 8 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to the software after maintenance for one functional process.
- Entity type 9 can be used to measure the functional size of the internal software maintainability throughout observe behaviour of user-developer-maintainer to investigate the software failures occurred after change for one functional process.

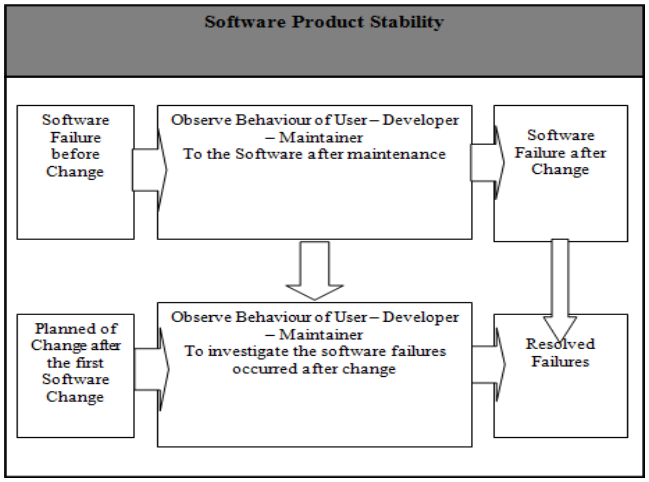

Fig. 3: Generic of FSM Model of Software Stability

3.3.4 Testability

In the following design of the Meta models- see Figure 4:

- Entity type 10 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to testing software after maintenance for one functional process.
- Entity type 11 can be used to measure the functional size of the external software maintainability throughout observe behaviour of user-developer-maintainer to testing software after maintenance for one functional process
- Entity type 12 can be used to measure the functional size of the internal software maintainability throughout observe behaviour of user-developer-maintainer to testing software after maintenance for one functional process.

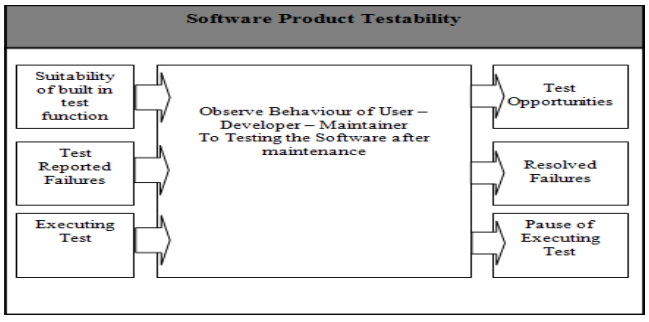

Fig. 4: Generic of FSM Model of Software Product Testability

3.4 Identification of data Maintainability Groups

This section illustrates data Maintainability groups as defined in ISO (i.e. Input and output) form sources and/or to data destinations for maintainability Requirements for details see-table 1 and table 2. Table 1 illustrates the maintainability requirements data sources, while table 2 illustrates the maintainability data destination. Furthermore, Data sources can be defined by software engineering developers and data destination can be test by software engineer testers based on data sources.

Table 1 : Maintainability Requirements Data Sources

Categories	Data Sources	Objects of Interest
Analysability	• planned data recorded during operation	Data
	• actual data recorded during operation	Data
	• Failure causes	Access
	• actual registered failure	Failure
	• Failure diagnoses	Failure
	• data monitor recording	Data
	• actual failed data monitor	Data
Changeability	• Designed required planned time to change	Time
	• actual work time to change	Time
	• number cases of change the software	Cases of change
	• number cases of failing change	Cases of change
	• planned of change the recorded of log data	Data
	• change of log data actually recorded	Data
Stability	• software failure after change	Failure
	• change success ratio	Time ratio
	• software failure before change	Failure
	• Software failure after change	Failure
Testability	• Suitability of built in test function.	Function
	• Test opportunities.	Test
	• The pause of the executing test.	Time
	• test re-startability	Time
	• The test reported failures.	Failure
	• The resolved failures.	Failure

Table 2 : Maintainability Data Destinations

Categories	Data Destinations
Analysability	• audit trial capability • diagnostic function support • failure analysis capability • status monitoring capability
Changeability	• change efficiency • modifiability • software change control capability
Stability	• Change success ratio. • modification impact
Testability	• availability of built in test function • Re-test efficiency • test re-startability

3.5 Numerical Rules for Maintainability

Numerical Assignment Rules: consists in defining the measurable concept and an empirical relational set, to complete the design of a measurement method, a numerical relational set and a homomorphism between these two relational sets must be defined, it can be achieved by answer the following questions.

- Defining the numerical assignment rules.
- Selection of a measurement unit + The justification for this selection
- How each sub-concept contributes to the numerical assignment rules: (A descriptive text and Mathematical expression).

In this step, the basis for these numerical assignment rules is the proposed meta-models and the characterization of the concept. A numerical assignment rule can be described through a descriptive text (a practitioner's description) or through mathematical expressions (a formal theoretical viewpoint).

3.5.1 Total measurement Size of Analyzability:

This section will illustrate the mathematical equations for analyzability as follows:

The functional size of the analyzability (externally and internally) for one process for the following:

- Audit trial capability
- Diagnostic function support
- Failure analysis capability
- Status monitoring capability

The total functional size of the analyzability [for the all functional processes]

= N x ∑ functional size of the analyzability (internally) + N x ∑ the functional size of the analyzability (externally)

N: number of functional processes for the analyzability.

3.5.2 Total measurement Size of Changeability:

This section will illustrate the mathematical equations for changeability as follows:

The functional size of the changeability (externally and internally) for one process for the following:

- Change efficiency
- Modifiability
- Software change control capability

The Total Functional **S**ize of the changeability [for the all functional processes]

= N x ∑ functional size of the changeability (internally) + N x ∑ The Functional size of the changeability (externally)

N: number of functional processes for the changeability.

3.5.3 Total measurement Size of Stability:

This section will illustrate the mathematical equations for stability as follows:

The functional size of the stability (externally and internally) for one process for the following:

- Change success ratio.
- Modification impact

The Total Functional **S**ize of the stability [for the all functional processes]

= N x ∑ functional size of the stability (internally) + N x ∑ The Functional size of the stability (externally)

N: number of functional processes for the stability.

3.5.4 Total measurement Size of Testability:

This section will illustrate the mathematical equations for stability as follows:

The functional size of the testability (externally and internally) for one process for the following:

- Availability of built in test function
- Re-test efficiency
- Test re-startability

The Total Functional Size of the testability [for the all functional processes]

= N x ∑ functional size of the testability (internally) + N x ∑ The Functional size of the testability (externally)

N: number of functional processes for the testability.

3.5.5 Total measurement Size of Maintainability:

The functional size of the maintainability (externally and internally) for one process

= ∑ data movement (Data group) (analyzability + changeability + stability + testability).

The Total Functional Size of the maintainability [for the all functional processes]

+ N x ∑ The Functional size of the maintainability (externally)

N: number of functional processes for the maintainability.

4. CONCLUSION

This paper introduced a new design measure internally and externally for the maintainability for the software product quality; as well as proposed three meta models for maintainability requirement using ISO standards independently of the software type or languages.

Moreover, the design of the measurement method clearly defining to specify the strategy of the measurement rules to perform the mapping between the concepts of ISO and the concepts of the suggesting design of the generic maintainability meta models and rules to identify the data movements and perform the measurement.

It is important to remark that the design measurement procedure for maintainability requirements have been developed to apply the ISO 19761 measurement method to the maintainability requirements in order to obtain the functional size of the maintainability as a separate piece of software in early stages of the software development process.

The advantages, limitations and case study of the meta models also stated during the sections of the paper as future work to enhance the proposed generic models and to applicable to use it in the industry.

Future work includes verification and validation of this proposed model to make sure full exposure of maintainability requirements. A proof with groups of experts will be performed to expand a consensual standard based model that may be proposed as a candidate for standardization.

5. Acknowledgment

Our thanks to IPAC 2015 conference team for their support and help our software engineering research groups at Hashemite University in Jordan.

6. REFERENCES

[1] ECSS-E-40-Part-1B, "Space Engineering: Software - Part 1 Principles and Requirements", European Cooperation for Space Standardization,The Netherlands, 2003.

[2] ECSS-E-ST-10C, "Space engineering: System engineering general requirements", Requirements & Standards Division, Noordwijk, The Netherlands, 2009.

[3] ECSS-E-40-Part-2B, "Space Engineeing:Software-part 2 Document Requirements Definitions", European Cooperation for Space Standardization, The Netherlands, 2005.

[4] ISO/IEC-24570 (2005). "Software engineering - NESMA Functional Size Measurement Method v.2.2 - Definitions and counting guidelines for the application of Function Point Analysis."

[5] ISO/IEC-14764, "Standard for Software Engineering: Software Life Cycle Processes—Maintenance", ISO, Geneva (Switzerland), 2006.

[6] ISO/IEC-9126, "Software Engineering - Product Quality - Part 1: Quality Model ", International Organization for Standardization, Geneva (Switzerland), 2004.

[7] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, and N.Tsirakis, "A data mining methodology for evaluating maintainability according to ISO/IEC-9126 software engineering –product quality standard," in Special Session on System Quality and Maintainability - SQM2007, 2007.

[8] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design." IEEE Trans. Software Eng., vol. 20, no. 6, pp. 476–493, 1994

[9] M. Broy, F. Deissenboeck, and M. Pizka, "Demystifying maintainability, "in Fourth International Workshop on Software Quality Assurance (SOQUA 2007). ACM, 2007.

[10] P. Oman and J. Hagemeister, "Metrics for assessing a software system's maintainability," in Proceedings of Conference on Software Maintenance, 1992., Nov. 1992, pp. 337–344

[11] M. Halstead. Elements of Software Science. Elsevier Science Inc., New York, NY, USA, 1977.

[12] T. J. McCabe. A complexity measure. ICSE 1976. IEEE CS Press, 1976.

[13] Aggarwal, K.K.; Singh, Y.; Chhabra, J.K., "An integrated measure of software maintainability," Reliability and Maintainability Symposium, 2002. Proceedings. vol., no., pp.235,241, 2002

[14] Deissenboeck, F.; Wagner, S.; Pizka, M.; Teuchert, S.; Girard, J.F., "An Activity-Based Quality Model for Maintainability," Software Maintenance, 2007. ICSM 2007. IEEE International Conference on , vol., no., pp.184,193, 2-5 Oct. 2007

[15] Al-Sarayreh, K. T., A. Abran and and J. J. Cuadrado-Gallego," A Standards-based model of system maintainability requirements", Journal of Software: Evolution and Process, John Wiley & Sons, Ltd, 2013. http://dx.doi.org/10.1002/smr.1553

[16] Meridji, Kenza and, Khalid T. Al-Sarayreh, "A generic model for the specification of software reliability requirements and measurement of their functional size." *International Journal of Information Quality* 3, no.2 (2013): 139-163.

[17] Al-Sarayreh, Khalid T., Ibrahim Al-Oqily, and Kenza Meridji. "A standard-based reference framework for system operations requirements." *International Journal of Computer Applications in Technology* 47, no. 4 (2013): 351-363.

[18] Abran, A., K. T. Al-Sarayreh, and J. J. Cuadrado-Gallego, "A Standards-based Reference Framework for System Portability Requirements", Computer Standards and Interface, Elsevier, 2013.