REQUIREMENTS MANAGEMENT MEASURES VALIDATION

MAHMOUD KHRAIWESH
Faculty of Science & Information Technology
Zarqa Private University
Zarqa-Jordan

ASIM EL SHEIKH
Faculty of Information System & Technology
The Arab Academy for Banking & financial Sciences
Amman-Jordan

ABSTRACT

Requirements management measures help organizations to understand, control and assess requirements management process. The goal of this paper is to validate a set of requirements management measures. The measures were defined for the five specific practices of Requirements Management Key Process Area (KPA) in Capability Maturity Model Integration (CMMI) by applying the Goal Question Metrics (GQM) paradigm to the five specific practices. We have made a questionnaire to prove the validity and reliability of the defined measures and confirm that they really measure the five specific practices. The questionnaire was filled by practitioners in six institutions specialized in software development. The collected data were analyzed by cronbach alpha in SPSS.

Keywords: Software Requirements**,** Requirements Management, Measures, Measures Validation

1. INTRODUCTION

Measurement is a mechanism for characterizing, evaluating, and predicting for various software processes and products [2]. The only way to improve any process is to measure specific attributes of the process, develop a set of meaningful metrics based on these attributes, and then use the metrics to provide indicators that will lead to strategy for improvement. Software measurement plays important role in understanding and controlling software development practices and products [10].

Since requirements often change, even during development, it is important to control the continuing definition of requirements as they change throughout the software life cycle to be able to anticipate and respond to requests of change [14]. Requirements are the foundation of the software development process. Carefully developed software requirements are a key issue for project success [9]. The reason for concentrating on this early phase of the software process was that problems in this area have a profound effect on system development costs and functionality [18].

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to characterize the attributes by clearly defined rules [6]. Measurement is important for three basic activities: understanding, control and improvement [5]. Reasons for measuring are: to assess achievement of quality goals, to determine status with respect to plans, to gain understanding of processes, products, resources, and environments, to establish baselines for comparisons with future assessments and track improvement efforts [12] .

Software engineering is not grounded in the basic quantitative laws of physics. Direct measure such as voltage, mass, velocity, or temperature, are uncommon in the software world. Because software measures and metrics are often indirect, they are open to debate [13]. Software measurement is currently in a phase in which terminology, principles and methods are still being defined and consolidated. We should not expect to find quantitative laws that are generally valid and applicable, and have the same precisions and accuracy as the laws of Physics, for instance. As a consequence, the identification of universally valid and applicable measures may be an ideal, long term research goal, which cannot be achieved in the near future [4].

Hall et al. [8] carried out a case study of 12 companies at different levels of capability as measured by the CMM. They discovered that, out of a total of 268 development problems cited, almost 50% (128) were requirements problems.

Organizations from industry, government, and the Software Engineering Institute (SEI) joined together to develop the CMMI Framework, a set of integrated CMMI models.

Two kinds of materials are contained in the CMMI model [1]:

1. Materials to evaluate the contents of the processes-information that is essential to

your technical, support and managerial activities.

2. Materials to improve process performance-information that is used to increase the capability of the organization's activities.

A survey of over 8000 projects from 350 US companies and revealed that one third of the projects were never completed and one half succeeded only partially, that is, with partial functionalities, major cost overruns, and significant delays. When asked about the causes of such failures, executive managers identified poor requirements as the major source of problems (about half of the responses) - more specifically, the lack of user involvement (13%), requirements incompleteness (12%), changing requirements (11%), unrealistic expectations (6%), and unclear objectives (5%). On the European side, a survey of over 3800 organizations in 17 countries similarly concluded that most of the perceived software problems are in the area of requirements specification (greater than 50%) and requirements management (50%) Lamsweerde [11].

The Goal/Question/Metric (GQM) paradigm to process and metrics was developed by Basili and Weiss [3] as a technique for identifying meaningful measures for any part of the software process. It has proven to be a particularly effective approach to selecting and implementing measures.

In our previous work [9] we analyzed the five specific practices defined in the Requirements Management Key Process Area (KPA) of the CMMI [15]. By means of the Goal Question Metrics (GQM) paradigm [2] we defined nearly 70 measures.

This paper validates the defined measures in [9] for the five specific practices of Requirements Management KPA in CMMI-SW (Staged Representation) model and confirm that they are really measure the five specific practices. The five specific practices are: Obtain an understanding of requirements, obtain commitment to requirements, manage requirements changes, maintain bidirectional traceability of requirements, and identify inconsistencies between project work and requirements.

The remainder of the paper is organized as follows: section 2 describes measurement theory, section, section 3 describes the validity and reliability of the defined measures, and section 4 presents conclusions and future research.

2. MEASUREMENT THEORY

Measurement is not solely the domain of professional. We use it in every day life. Price acts as a measure of values of an item in a shop. When making a journey, we calculate distance, choose our route, measure our speed, and predict when we will arrive at our destination. So measurement helps us to understand our world, interact with our surroundings and improve our lives.

Software measurement is concerned with deriving a numeric value for an attribute of a software product or process. By comparing these values to each other and to standards that apply across an organization, you may be able to draw conclusions about the quality of software or software process.

Software measurement is concerned with deriving a numeric value for some attributes of product or process. By comparing these values to each other and to standards that apply in the organization we can conclude the quality of the product or process [17]. Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to defined rules [5] [16].

Measurement captures information about attributes of entities. An entity is an object (such as person or a room) or an event (such as a journey) in real world. We want to describe the entity by identifying characteristics that are important to distinguishing one entity from another. An attribute is a feature or property of an entity, the area or color of a room and the cost of a journey. When we describe entities by using attributes, we often define the attributes using numbers or symbols. Some software engineers claim that important attributes like dependability, quality, usability and maintainability are simply not quantifiable, we prefer to try to use measurement to advance our understanding of them [5].

Formally we define measurement as mapping from empirical world to formal, relational world. Consequently, a measure is the number or symbol assigned to an entity in order to characterize an attribute by this mapping. We begin in the real world, studying the entity. Thus the real world is the domain of the mapping and the mathematical world is the range.

The purpose of performing the mapping is to be able to manipulate data in the numerical system and use the result to draw conclusions about the attribute in the empirical system. But not all measurement

mapping are the same. And differences among the mappings can restrict the kind of analysis we can do. We refer to our measuring mapping as a measurement scale [5].

Although some companies have introduced measurement programs, most organizations still don't make systematic use of software measurement. Because the software processes are poorly defined and controlled, and are not sufficiently mature to make use of measurements. Another reason is that there are few established standards in this area.

Software metrics may be either predictor metrics used to predict product attributes or control metrics used to control the software process [17].

In software there are three classes of entities and attributes we wish to measure:

1. Processes: Are collections of software-related activities.
2. Products: Are any artifacts, deliverables or documents that result from a process activity.
3. Resources: Are entities required by a process activity (example: documentation from previous phase).

Within each class of entity, we can distinguish between internal and external attributes:

- Internal attributes of a product, process or resources: Are those that can be measured purely in terms of the product, process or resources itself .
- External attributes of a product, process or resources: Are those that can be measured only with respect to how the product, process or resources relates to its environment.

The relationship between the internal and the external attributes should be clear and validated. (example: stability of requirements is external attribute, while number of requirements changes is internal attributes). It is impossible to measure software quality attributes directly. Quality attributes such as maintainability, understandability and usability are external attributes that relate to how developers and users see the software. They are affected by many factors and there is no simple way to measure them [17].

Direct measurement of an attribute of an entity involves no other attributes or entity (length of source code measured by lines of code, duration of testing process measured by elapsed time in hours). Indirect measurement of an attribute of an entity involves other attributes or entity.

3. VALIDITY AND RELIABILITY OF THE DEFINED MEASURES

We have made a questionnaire to prove the reliability and validity of the defined measures and confirm that they are actually measure the five specific practices. The collected data will be analyzed by cronbach alpha reliability in SPSS.

The questionnaire was reviewed and confirmed by academics and practitioners in software engineering and software development. The questionnaire was filled by system analysts and software engineers. The questionnaire consists of five parts, each part is related to one specific practice of the requirement management process, each part consists of a group of statements (measures) related to the specific practice, beside each statement there is five options: strongly agree, agree, neither agree nor disagree, disagree, strongly disagree. The questioner will read the statement and write his opinion of the statement relation with the specific practice by choosing one of the five options, a sample shown in Appendix A.

Cronbach alpha is designed as a measure of internal consistency, that is, do all items measure the same thing? (measure a single unidimentional structure). Cronbach alpha varies between 0 and 1, the closer the alpha is to 1, the greater the internal consistency of items being assessed [7]. If alpha is less than 0.5 then internal consistency is unacceptable.

We distributed the questionnaire on six institutions in Jordan: Zarqa Municipality, Islamic Hospital, Adaptive Techsoft (ATS) software house, Computer and Engineering Bureau (CEB), Hashemite University, and Zarqa Private University. We have collected fifty questionnaires: nine questionnaires from Zarqa Municipality, five questionnaires from Islamic Hospital, ten questionnaires from Adaptive Techsoft (ATS) software house, ten questionnaires from computer and engineering bureau (CEB), six questionnaires from Hashemite University, and ten questionnaires from Zarqa Private University.

After applying the collected data on Cronbach Alphain in SPSS we got the following standardized items alpha results:

- 0.769 for the first group of statements which are related to specific practice1: obtain understanding of requirements, which means

that the statements (measures) are consistent and have a good reliability and validity to measure the understanding of requirements.

Statement serial	statement
1	The requirements providers direct relation to work has a positive effect on understanding of requirements.
2	The requirements providers good understanding of the work has a positive effect on understanding of requirements.
3	The requirements providers desire to improve the work has a positive effect on understanding of requirements.
4	More than one requirement provider involved has appositive effect on understanding of requirements.
5	Variety of many management levels the requirements providers (users) are from has a positive effect on understanding of requirements. **(management levels**: employee, manager, high manager,…)
6	The uniquely identified requirements has a positive effect on understanding of requirements. (**uniquely identified**: independently and clearly)
7	Shared understanding and agreement when analyzing requirements is important for understanding of requirements. (**Shared understanding** between users and technicians)
8	The increase number of identified misunderstand Requirements has negative indication on understanding of requirements.
9	The increase number of identified missing requirements has negative indication on understanding of requirements.
10	The increase number of identified rejected requirements after implementation has negative indication on understanding of requirements.

- 0.524 for the second group of statements which are related to specific practice2: obtain commitment to requirements, which means that the statements (measures) are consistent and have a good reliability and validity to measure the commitment to requirements.

Statement serial	Statement
11	The number of requirements that are implemented and delivered as planned is important for commitment to requirements.
12	The number of requests to change that are implemented and delivered as planned is important for commitment to requirements.
13	Identifying the cost of each request to change is important for commitment to requirements. (cost: estimated time and actual time needed to finish the change)

- 0.769 for the third group of statements which are related to specific practice3: manage requirements changes, which means that the statements (measures) are consistent and have a good reliability and validity to measure the management of requirements changes.

Statement serial	statements
14	The initial and current number of requirements in the project is important to manage requirements changes.
15	The initial and current number of requirements to be determined (TBD) initially is important to manage requirements changes. (**TBD**: postpone requirements to be processed later**)**
16	The historical status of each requirement is important to manage requirements changes. (**requirement status**: proposed, analyzed, TBD, approved, tested, implemented, delivered,…)
17	The historical status of each requests to change is important to manage requirements changes. (**request to change status**: proposed, analyzed, TBD, approved, tested, implemented, delivered,…)

18	Identifying the other requirements that are affected by the request to change is important to manage requirements changes.
19	Identifying the source of each request to change is important to manage requirements changes. (**source:** department and user)
20	Identifying the reason for each request to change is important to manage requirements changes. (**reason**: missing, new regulation, improvement, misunderstanding, inconsistencies)
21	Identifying the type of change on requirement for each request to change is important to manage requirements changes. (**type of change** on requirement: change, delete, new).
22	Identifying the cost of each request to change is important to manage requirements changes. (**cost:** estimated time and actual time needed to finish the change)
23	Identifying the project items that are affected by the request to change is important to manage requirements changes. (**project item**: form, report, entity, menu,….)
24	Following the number of request to change for each requirement (decrease or increase) is important to manage requirements changes.

- 0.780 for the fourth group of statements which are related to specific practice4: maintain traceability of requirements, which means that the statements (measures) are consistent and have a good reliability and validity to measure traceability of requirements.

Statement serial	Statement
25	Identifying the source of each request to change is important to maintain traceability. (**source**: department and user)
26	Identifying the relationship between requirements is important to maintain traceability. (**relationship** between requirements: identify the requirements which are related in the system)
27	Identifying the project items related to each requirement is important to maintain traceability. **(project item**: form, report, entity, menu,….)
28	Identifying other related requirements that are affected by the request to change is important to maintain traceability. (**project item**: form, report, entity, menu,….)
29	Identifying the project items that are affected by the request to change is important to maintain traceability. (**to estimate** the time needed to do the changing on the project item before give commitment for the request to
30	Identifying the cost of each request to change is important to maintain traceability. (**cost:** estimated time and actual time needed to finish the change)

- 0.728 for the fifth group of statements which are related to specific practice5: identify inconsistencies of requirements, which means that the statements (measures) are consistent and have a good reliability and validity to measure identifying inconsistencies of requirements.

Statement serial	Statement
31	Identifying the source of inconsistencies cases in requirements is important to identify inconsistencies. (**source of inconsistency**: report, form, database, function,…) (**inconsistencies** between requirements and products)
32	Identifying the reason of inconsistencies cases is important to identify inconsistencies. **(reasons of inconsistency**: requirements provider, practitioner, documentation,…)
33	The number of inconsistencies cases is important to identify inconsistencies. (**inconsistency** between requirements and products)
34	Identifying other requirements that are affected by the inconsistency is important to identify inconsistencies.
35	Identifying the project items that are affected by the inconsistency case is important to identify inconsistencies. (**project item**: form, report, entity, menu,….).

4. CONCLUSION AND FUTURE RESEARCH

This paper has proved the validity and reliability of the defined measures in [9] and confirm that they are really measure the five specific practices, by using a questionnaire and analyzing the collected data by cronbach alpha in SPSS. The questionnaires were filled by system analysts and software engineers in six institutions of software development.

An important area of future research is to prove the validity of the defined measures empirically, by applying the defined measures on some information systems. Another important area of future research is the definition of measures for other key process areas in CMMI.

REFERENCES

[1] Ahern, D. M., Clouse, A. and Turner, R., *CMMI Distilled: A Practical introduction to Integrated Process Improvement*. Second edition, Addison – Wesley, 2003.

[2] Basili, V. R. and Rombach, H. D., "The TAME Project: Towards Improvement-Oriented Software Environments", in *IEEE Transactions on Software Engineering*, vol. 14, no. 6, pp.758-773, 1988.

[3] Basili, V. R., and Weiss, D. M., "A Methodology for collecting Valid Software Engineering Data", in *IEEE Transactions on Software Engineering*, volume. SE-10, pp.728-738, 1984.

[4] Briand, L.C., Morasca, S., and Basili, V. R., "An Operational Process for Goal Driven Definition of Measures", in *IEEE Transactions on Software Engineering*, vol. 28, no. 12, 2002.

[5] Fenton, N.E., and Pfleeger, S.L., *Software Metrics - A Rigorous & Practical Approach*, 2nd Edition, International Thomson Publishing, Boston, MA, 1996.

[6] Fenton, N. E. & Whitty, R. and Yoshinori, I. *Software Quality Assurance and Measurement, A Worldwide Perspective*. London: International Thomson Computer Press, 1995.

[7] George, D. and Mallery, Paul., SPSS for windows step by step A Simple Guide and Reference, Fourth Edition, 2003.

[8] Hall, T., Beecham, S., and Rainer, A. "Requirements problems in twelve software companies: An empirical analysis", *IEE Proceedings: Software* vol. 149, issue. 5, 153-160, 2002.

[9] Khraiwesh, M. and El Sheikh, A., "Requirements Management Measures", *in proceedings of the International Arab Conference on Information Technology ACIT'2005*, Dec 6-8, Jordan, 2005.

[10] Kitchenham, B., Pfleeger, S.L., and Fenton, N., Towards a Framework for Software Measurement Validation, in IEEE Transactions on Software Engineering, 21(12), December 1995.

[11] Lamsweerde, A. , "Requirements engineering in the year 00: A research perspective". In *proceedings of the 22nd International Conference on Software Engineering (ICSE'2000)*, Limerick, Ireland, 5-19, ACM Press, 2000.

[12] Park, R. E., Goethert, W. B. and Florac, W. A., *Goal-Driven Software Measurement-A Guidebook*, Software Engineering Institute Handbook, CMU/SEI-96- HB-002, August, 1996.

[13] Pressman, R. S., *Software engineering: A practitioner's Approach*, Sixth edition, 2005.

[14] Reifer D.J., "Requirements Management: The search for Nirvana", in *IEEE Software,* May/June, pp. 45-47, 2000.

[15] SEI (Software Engineering Institute), *Capability Maturity Model - Integrated - Software Engineering Stage Representation*. Version 1.1. Carnegie Mellon University Software Engineering Institute, August, 2002.

[16] Solingen, V. R., and Berghout, E., *The Goal/Question/Metric Method - A practical Guide for Quality Improvement of Software development*, McGRAW-Hill Companies, London, 1999.

[17] Sommerville, I., Software engineering, Addison – Wesley, Seventh Edition 2004.

[18] Sommerville, I., and Ranson, J., "An Empirical study of Industrial Requirements Engineering Proces Assessment and Improvement", in *ACM Transactions on Software Engineering and Methodology* vol. 14, no. 1, pp. 85-117, 2005.

APPENDIX A

Questionnaire and Analysis

Questionnaire:

This questionnaire is related to the requirement management process. Requirement management is the process of understanding and controlling changes to requirements.

The requirement management process has five goals:

1- Understanding of Requirements.
2- Commitment to Requirements.
3- Manage Requirements Changes.
4- Maintain Bidirectional Traceability.
5- Identify Inconsistencies.

We would like to measure the achievement of the above goals, so, we define some statements related to each goal. We suppose that the information in these statements help us in achievement of the above five goals.

Please, fill the enclosed form by writing √ in the suitable place. Responding to this question: do you think that the statements have an effect on the achievement of the goals?

Goal1: Obtain an Understanding of Requirements: Develop an understanding with the requirements providers (users) on the meaning of the requirements.

(do you think that these statements have an effect on the achievement of goal1: obtain an Understanding of Requirements ?)

Statement serial	Statements	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
1	The requirements providers direct relation to work has a positive effect on understanding of requirements.					
2	The requirements providers good understanding of the work has a positive effect on understanding of requirements					
3	The requirements providers desire to improve the work has a positive effect on understanding of requirements.					