International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 6
ISSN: 2088-5334

A Conceptual Framework for Metrics Selection: SMeS

Zubaidah Bukhari#, Jamaiah Yahaya#, Aziz Deraman*

#*Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia*
E-mail: zubaidabukhari@gmail.com, jhy@ukm.edu.my

**School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, Kuala Terengganu, Malaysia*
E-mail: a.d@umt.edu.my

***Abstract*— Measurement is an emerging field of software engineering. A systematic and efficient measurement process can assist in the production of quality government web applications since it can support planning, monitoring, and improving the software development process. The quality of software products is often measured using the metrics-based method. Various types of software metrics have been suggested in the last three decades to measure software quality, track software progress, estimate effort and certify software products. However, the process of identifying and determining a suitable software metrics is still lacking in term of guidance and structure. Without an effective method to evaluate and select suitable metrics, the time spent for selecting the correct and relevant metrics may offset the advantages of using them. In this paper, using the Evaluation Theory as the base theory and building upon the integration of Goal Question Metric (GQM) approach, a conceptual framework of software metrics selection (SMeS) was proposed. The main elements of the proposed conceptual framework include goal-based metrics selection process integrating with GQM, evaluation criteria, reference standard, and synthesis technic in decision making mechanisms. The integrated framework produces a software metrics selection process which is more structured, transparent, and directive in choosing a set of relevant metrics that may help the organizations' measurement program. The result from this study improves the existing metrics software selection model and serves as a guideline in implementing a clear and practical software metrics selection.**

***Keywords*— software metrics; software quality; evaluation theory; goal question metrics.**

I. INTRODUCTION

Systematic measurement is important in software organizations to track and maintain the quality attributes of project deliverables. Making the measurement processes work in organizations requires collecting correct and relevant metrics. In order to obtain metrics and measurements that address the needs of organizations, the measurement process must be structured and guided. Software metrics is a measure of software characteristics, which are measurable or countable. Software metrics is "an objective, mathematical measure of software that is sensitive to differences in software characteristics. It provides a quantitative measure of an attribute which the body of software exhibits" [1]. Metrics are used to improve software quality, to track software progress, and to certify software products [2]. Many studies have proposed different types of metrics such as security metrics[3], crawlability metrics [4], and web application metrics [5]. Software metrics will influence a measurement program and eliminating inaccurate metrics will improve software performance and reduce software wastage [5].

However, there is no consensus on which metrics are relevant and worthy of selection [2], [7], [5], [8]. Studies have verified that after the second year of implementing measurement metrics, 50%-80% of these measurements are not maintained [9], [6]. The studies also found a very high failure rate in metric implementation—66.7%. This paper addresses these issues by integrating the Evaluation Theory and Goal-Question-Metric (GQM) as a basis for a proposed conceptual framework for software metrics evaluation and selection model.

This framework informs practitioners of the most valuable metrics and most important goals of organizations, and thus help them make informed decisions when collecting the right metrics. This study proposed two research questions. Firstly, what are the components needed in a software metrics evaluation and selection model?. Secondly, how can the components be integrated to develop a model?This paper is organized as follows: the materials and methods are presented in Section II. Results and discussion are outlined in Section III. Finally, a summary of the findings and recommendations for future work are laid out in Section IV.

II. Material and Method

A good measurement framework has the following characteristics: i) is consistent with user objectives; ii) measures what is required; iii) measures all elements that the user needs to achieve his goals, and iv) is valid in the environment it is used [3], [13], [5], [14]. Measurement plays an integral role in improving software processes. Also, it also increases the effectiveness of communication between software organizations and customers. The success of measurement programs depends on a two-level approach. The first level involves collecting metrics, coaching, metrics analysis, data collection procedures, metrics quality, automatic tools development, and response [3]. The second level includes goal setting by stakeholders, resource sufficiency, organization funding, and capability level [15]. The details of the success factors can be read in [15].

A. Research Background

The management has access to numerous models, standards, and frameworks, which have been developed to assist them in arranging measurement programs. GQM-DSFMS, a GQM-decision support framework for metrics selection, is one such example [11] as well as the software ageing measurement framework [16], GQM for efficient measurement framework [17], quality model for web-based application [18], software security measure[19] and measurement processes in SMEs [6].

Metrics selection is always challenging in a *bottom-up* method. Measurement goals should be clear in order to select the proper metrics. Metrics that are embedded into a goal-oriented framework are broadly viewed as having good implementation. One of the widely used approaches is the Goal-Question-Metric (GQM) method, which was created by Dr. Victor Basili from the University of Maryland with Dr. David Weiss contributing in 1984. GQM works on the concept of linking organizational information needs, strategies, and goals of decision makers to measurement needs.

This approach provides a method for an organization or a project to define metrics in a *top-down* approach. First, specific goals are defined, questions are asked, and goals are enhanced into specifications of the data to be collected. Then, the resulting data is analyzed and interpreted in light of the original goals [20][5]. Studies on access security in cloud storage have proven the GQM method as a good quality approach [21]. Besides that, GQM has also proven both adaptable and rigorous, as indicated by various empirical studies conducted in different contexts [22], [5], [23], [21].

Such methods help in the development and establishment of measurement programs at the early stage. However, as time passes, dealing with and supporting the measurement programs become challenging. Some of the reasons for this are (i) *Lack of commitment.* A measurement program's objectives and goals should be precise and easily understandable across organizations so that the support from top management for the measurement programs is ensured and data providers are encouraged to collect the most up-to-date and specific data [24]; (ii) *Lack of guidance.* It is important that the metrics be collected be determined as early as the initial planning in order to fulfill the organization's needs [12]. There is no standard way of defining the structure, quantity, and syntax of goals and questions [25]; (iii) *Insufficient cost-saving methods.* Collection of data has been a critical activity due to cost concerns.

Furthermore, organizations have restricted resources to fund measurement programs. To achieve cost reduction when managing measurement programs, questions and goals could be reused, and the number of metrics limited [25]; (iv) *Lack of measurement expertise.* The lack of informed judgment for selecting the relevant number of metrics is another reason for unsuccessful software measurement programs. Due to universities only teaching measurement topics to post-graduates and not making it compulsory for undergraduates, software engineering graduates have insufficient measurement skills [26]; (v) *Complexity*. As time passes, the complexity of measurement programs increases, mainly due to adjustments in metrics, vertical and horizontal dependencies between goals, information needs, and change of stakeholders in dynamic environments [21].

The Evaluation Theory [27] is very closely related to this study. Generally, an evaluation includes identification of appropriate standards, assessment, scrutiny of the achievement of targets, and analysis of the results. In conducting the evaluation, six evaluation components need to be considered, which are listed below [27]–[31]:

- Target: the object under evaluation.
- Evaluation criteria: the characteristics of the evaluated object.
- Reference standard/Yardstick: the ideal target that is compared with the real target.
- Data gathering technique: a technique required to collect data for each criterion under analysis.
- Synthesis technique: a technique used to synthesize the information gained with the data gathering technique.
- Evaluation process: activities to be performed for the evaluation process.

Several researchers in the software engineering field have used the Evaluation Theory as the benchmark for their studies. Al-Tarawneh [28] used the theory to develop an evaluation method for Commercial Off-The-Shelf (COTS). Mohamad [31] applied this theory to develop a certification and evaluation software process model based on an agile and secure software process. Zarour [32] utilized the theory to develop a user needs experience evaluation method. This paper proposes a software metrics evaluation and selection model built upon Goal-Question-Metric (GQM) and Evaluation Theory. The proposed model considers the important attributes of software metrics, and emphasizes on the most important goals, which translate to the most valuable metrics. Therefore, the practitioners are continuously supported in making informed decisions on which metrics to collect.

B. Research Approach

The following sections discuss the four main phases implemented in this research work.

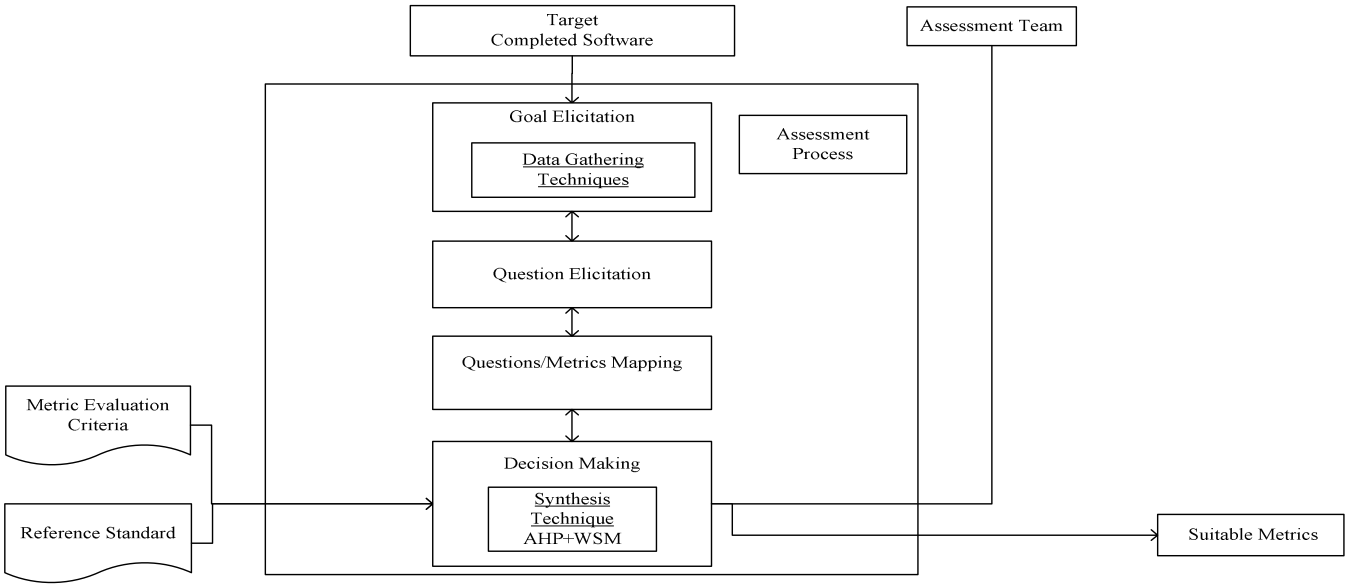

Fig. 1 Conceptual Framework for Software Metrics Selection (SMeS)

1) Phase 1 - Conceptual Framework Design: The first phase of this research is to investigate further and formulate software metrics evaluation and selection of a conceptual framework. The references consist of latest journals, books, and proceedings. This phase aims to identify the key elements that exist in software metrics evaluation and selection implementation and also to investigate issues and problems related to its implementation. The outcome of this phase is the conceptual framework of software metrics evaluation and selection, which integrates the Evaluation Theory and the Goal-Question-Metric paradigm.

2) Phase 2 - Empirical Investigation on Software Metrics Evaluation Criteria: The second phase is to design semi-structured interview questions and conduct interviews with identified experts in these areas. The interviewees are practitioners in software measurement and software testing. They were selected based on their experience and position in related works. This study aims to identify the evaluation criteria for software metrics that are important in evaluating and selecting the software metrics from the industry's perspective. The interviews were conducted face-to-face and were recorded in audio files. The outcome of this phase reveals the key elements in software metrics evaluation and selection criteria.

3) Phase 3 - Development of Software Metrics Selection Model (SMeS): The next phase is to develop a model for software metrics selection for software testing phase based on the empirical study findings as well as theoretical studies. Relationships between key elements were investigated and modeled to support the model development. The outcome of this phase is the software metrics selection model (SMeS).

4) Phase 4 - Model Validation and Verification: The last phase of this research is to validate the model. The validation adopts the case study and expert review approaches. Three experts in software metrics were identified and invited to review the model. The model was also applied in real practice, which involved collaboration with at least two government agencies for evaluation. All the review reports were analyzed and used to refine the framework.

III. Results and Discussion

A conceptual framework refers to "a theoretical structure of assumptions, principles, and rules that holds together the ideas that comprise a broad concept." GQM and Evaluation Theory were used as a basis for developing the conceptual framework in this study, which evaluates and selects software metrics. GQM consists of four components: goal, question, metric, and decision-making. Meanwhile, the Evaluation Theory consists of six main components: targets, evaluation criteria (non-functional requirements), standard references, data gathering techniques, synthesis techniques, and evaluation processes. The conceptual framework shows the main issues that should be considered (see Figure I). They are discussed further below:

A. *Defining the Target*

In any evaluation, the target is the initial crucial stage. Referring to the scope of this study, the target is the evaluation and selection of software metrics in the projects that have been completed and that the software is ready to be delivered to customers. The software metrics in this paper are adapted from [33], [34]. These software metrics are stored in a single repository to be used in the study. The software's life cycle is divided into four phases (requirement, design, implementation, and testing).

B. *Defining the Evaluation Criteria*

In this important and crucial component, the required evaluation criteria for evaluating the target are defined. The evaluation criteria describe what the model should access, which comprises the criteria that influence the evaluation of the software metrics. Several surveys and suggestions have been made by past researchers to identify metrics evaluation criteria. Among them are the evaluation of software metrics through the measurement theory [16]–[18], IEEE standards [26], Kaner framework [27], Wayuker's properties [28], new algorithm for metrics-based software defect prediction[35], [36] and constructing the membership functions of software metrics [37]. Most of the proposed criteria are suited for the internal measurement of a software product that uses procedural or object-oriented programming languages such as Wayuker's properties [29]. However, a software product that meets the internal measurement criteria at times may not be enough to guarantee the success of external measurement [30]. It is crucial to develop the measurement of software products according to an external measurement based on user satisfaction, to ensure the software product quality [10], [23]–[25]. This study takes steps to evaluate product metrics by external measurement. From the literature, the study found that there are 13 criteria for software metrics selection, which are measurement scale [6], [38], [39], measurement independence[6], [40], [39], automation [39]–[41], simplicity [6], [40]–[42], accuracy [41], [42], [39], [6], environmental [39], type of users [39], [41], programming language independence [33], [37], feedback [34], cost [41], [42], [39], [6], comparability [43], [6], applicability [43], [6], and green [47]. Table I illustrates the specific definition of the evaluation criteria. The criteria definitions aid in the accurate understanding of the reference standard (yardstick) and this would help the decision maker know exactly what characteristics are to be analyzed. All the criteria will be examined and evaluated by the experts in the empirical phase.

TABLE I
EVALUATION CRITERIA DESCRIPTION

No.	Evaluation Criteria	Description
1.	Measurement Scale (MS)	Scale for measure metrics such as nominal, ordinal, interval, ratio, and absolute [6], [38], [39]
2.	Measurement Independence (MI)	The ability of metric to offer the same outcome of measurement [6][40][39]
3.	Automation (AUT)	The effort required to measure the metric using an automatic device [39]–[41]
4.	Simplicity (SIM)	Metric’s definition clarity [6][40]–[42]
5.	Accuracy (ACC)	Measure what is supposed to be measured [41][42][39][6]
6.	User Type (UT)	The kind of user involved in the calculation of the value of a metric 40], [42]
7.	Cost (CO)	Cost of using the metrics for evaluation [41][42][39][6],
8.	Environmental (ENV)	Associated with the type of measurement process [39]
9.	Programming Language Independence (PLI)	Metrics should not be dependent on the programming language syntax [33], [37]
10.	Feedback (FEE)	Provide with information that can lead to a higher-quality end product [34]
11.	Comparable (COM)	The criteria of goodness were generated after performing a comprehensive study for selecting the most popular metric [43][6]
12.	Applicability (APP)	The suitability of metric to the output of different phases of the system development life cycle [43][6]
13.	Green (GRE)	The ability of metrics to reduce energy consumption and promote sustainability [47]

C. Building the Reference Standard

The reference standard is the benchmark used by the assessors to perform the assessment. The model defines what needs to be assessed through the evaluation criteria and how these evaluation criteria are assessed. Each evaluation criterion is assigned with appropriate practices to archive the specified evaluation criterion. Each evaluation criterion is assigned with weight value, and the score achieved. In order to systematically organize the “whats” and “hows,” the Quality Function Deployment (QFD) approach is adapted. There are five main areas in QFD adapted in this study as the reference standard: “whats,” “hows,” relationships between “whats” and “hows,” weight for each evaluation criterion, and evaluation criteria scores (Figure 2). QFD has been used successfully to support many areas such as strategic maintenance technique selection [48], upgrading the service quality of mobile banking [49], as professional skill indicators [50], service quality assessment design [51], PSS design [52], in advanced biofuel policies in which a novel method was applied [53], evaluation of the performance of industrial waste environmental service providers [54], and in monitoring the quality of ready-mixed concrete (RMC) [55].

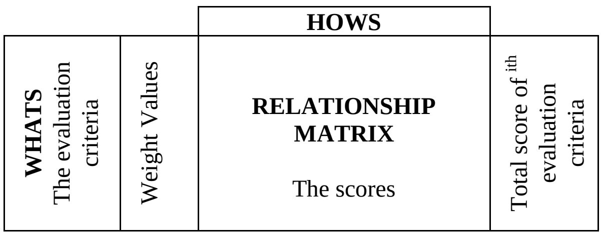

Fig. 2 Quality Function Deployment

D. Determining the Data Gathering Technique

The data gathering technique is a technique used for assessing criteria under analysis. This technique can help in suggesting better consideration for the evaluation team and providing better confirmation on the evaluation made. The proposed model adapted the multiple data gathering technique such as document review, interviews, and observation [48], [56], [57].

E. Determining the Evaluation Process

The evaluation process gives guidance on how to perform the evaluation process. There are two main phases for conducting the evaluation, which is the process of pre-evaluation and evaluation. Several processes and activities make up each of these phases. Primarily the structure of the evaluation process was adapted from the study of [11] and [58]. Pre-evaluation phase consists of the process of developing a committee to evaluate software metrics. The committee may consist of management-level employees, project leader, or a software developer. The committee should plan the activities carried out during the evaluation phase.

In the evaluation phase (as shown in Figure 3), there are four main components for conducting the software metrics evaluation, which are goals elicitation, questions elicitation, questions metrics mapping, and decision making [11]. In the first evaluation process, the goals and relationships among goals need to be identified beforehand. First, the eliciting and documentation of organizational long-term and short-term goals are done and the value of their significance rated, followed by the final process of identifying the related measurement goals [11]. In the second evaluation process, the questions to achieve the goals are elicited. The study used examples in previous research to guide in the development of the questions [54], [60]. The next process was to map the questions to the corresponding metrics. GQM-SMeS incorporated a software metrics. Mapping of relevant metrics to the identified questions was performed. Finally, the synthesis technique was employed in the process of decision-making (see section F).

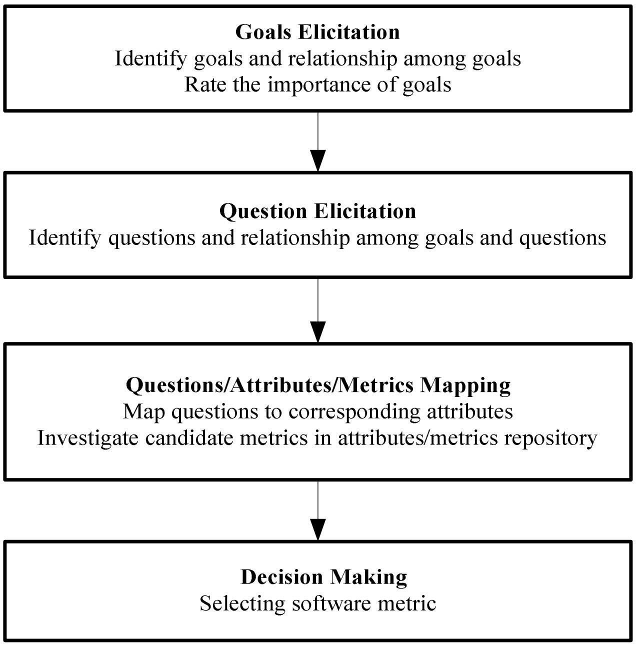

Fig. 3 Evaluation Phase adapted from [11]

F. Determining the Synthesis Technique

As stated previously, 13 identified criteria need to be considered in selecting software metrics. The identified criteria need to be weighted according to their relative importance. However, with an increased number of criteria to measure, the allocation of weights to the relevant criteria becomes more difficult. Some stakeholders might be involved in the selection process; thus resulting in views and considerations that are different such as the consideration of which criteria is important, how to measure that importance, and how to combine the various important criteria. These processes need a synthesis technique to organize and synthesize the information and to obtain results for evaluation and selection. The technique is known as Multi-Criteria Decision Making (MCDM) and is commonly implemented whenever decisions are controversial due to multiple conflicting criteria [66], [67].

A common and popular method of the MCDM technique is the Analytical Hierarchy Process (AHP). AHP represents a problem using hierarchical structures and then based on the opinion of stakeholders, develops priorities for alternatives [68] (see Figure 4 for the AHP tree). In this method, the view of each stakeholder, regarding the right weight age to assign to each criterion, is captured. Each stakeholder must compare each pair of criteria (with n criteria—there are n (n-1)/2 pairs), and assess the relative importance of each criterion in a ratio form in a process called pairwise comparison. In this way each stakeholder will be assigned a matrix created just for him or her. Using the weight age method, the matrices are then combined and analyzed. AHP has been used successfully for supplier selections [68], [69], evaluating cost database components [70], software selections [69], and prioritizing key success factors of software projects [71].

G. Determining the Synthesis Technique

As stated previously, 13 identified criteria need to be considered in selecting software metrics. The weighting of each identified criteria based on their relative importance must be done. However, the increased number of criteria also increases the difficulty of weight allocation to the relevant criteria. Moreover, some stakeholders might be involved in the selection process; thus resulting in views and considerations that are different such as the consideration of which criteria is important, how to measure that importance, and how to combine the various important criteria. These processes need a synthesis technique to organize and synthesize the information and to obtain results for evaluation and selection. The technique is known as the Multi-Criteria Decision Making (MCDM) and is commonly implemented whenever decisions are controversial due to multiple conflicting criteria [66], [67].

IV. CONCLUSION

This paper has presented a conceptual framework for evaluating and selecting software metrics that integrate the Goal-Question-Metric (GQM) method and the Evaluation Theory. The next step in this research is to conduct interviews with selected respondents to examine the identified processes and criteria defined in the conceptual framework and to investigate further the current practices for evaluating and selecting software metrics. Findings from this interview as well as from the literature will be used as input to the developing the framework for evaluating and selecting software metrics to support software quality process and implementation.

ACKNOWLEDGMENT

Universiti Kebangsaan Malaysia, under the Research University Grant (AP-2017-005/3), partly funded this research.

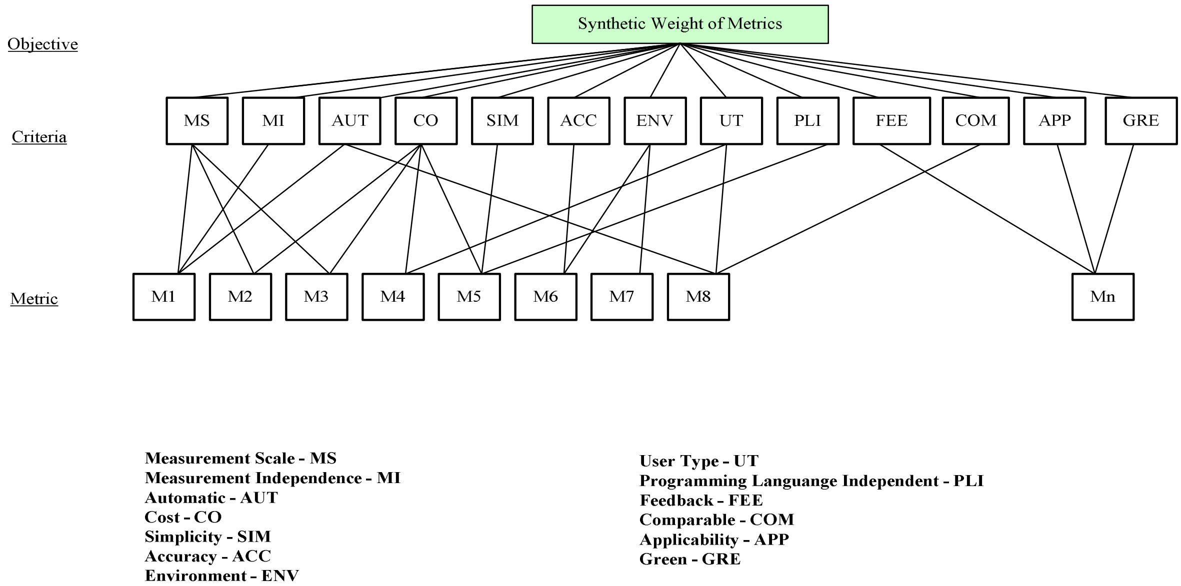

Fig. 4 Analytic Hierarchy Model

REFERENCES

[1] J. E. Gaffney Jr, "Metrics in software quality assurance," in *Proceedings of the ACM'81 conference*, 1981, pp. 126–130.

[2] I. R. Management Association, *Application Development, and Design: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2017.

[3] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809–3824, Jun. 2014.

[4] Y. Jamaiah, D. Aziz, K. Siti Sakira, and A. Ruzita, "Intelligent Software Quality Model: The Theoretical Framework," *Proc. 3rd Int. Conf. Comput. Informatics, ICOCI*, 2011.

[5] H. Wang, T. M. Khoshgoftaar, R. Wald, and A. Napolitano, "A comparative study on the stability of software metric selection techniques," *Proc. - 2012 11th Int. Conf. Mach. Learn. Appl. ICMLA 2012*, vol. 2, pp. 301–307, 2012.

[6] M. Noman, T. Tahir, and G. Rasool, "An assessment of key factors for implementing measurement processes in SMEs," *2017 Int. Conf. Open Source Syst. Technol.*, no. March 2018, pp. 60–66, 2017.

[7] C. Jones, *A Guide to Selecting Software Measures and Metrics*. CRC Press, 2017.

[8] M. Unterkalmsteiner, *Coordinating Requirements Engineering and Software Testing*. 2015.

[9] T. Tahir, G. Rasool, and M. Noman, "A Systematic Mapping Study on Software Measurement Programs in SMEs," *e-Informatica Softw. Eng. J.*, vol. 12, no. 1, pp. 133–165, 2018.

[10] M. Díaz-Ley, F. García, and M. Piattini, "Implementing a software measurement program in small and medium enterprises: a suitable framework," *IET Softw.*, vol. 2, no. 5, pp. 417–436, 2008.

[11] C. Gencel, K. Petersen, A. A. Mughal, and M. I. Iqbal, "A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS," *J. Syst. Softw.*, vol. 86, no. 12, pp. 3091–3108, 2013.

[12] T. Tahir, G. Rasool, and C. Gencel, "A systematic literature review on software measurement programs," *Inf. Softw. Technol.*, vol. 73, pp. 101–121, 2016.

[13] S. A. Ansar and R. A. Khan, "A Phase-wise Review of Software Security Metrics," in *Networking Communication and Data Knowledge Engineering*, Springer, 2018, pp. 15–25.

[14] R. M. Carvalho, R. M. de Castro Andrade, and K. M. de Oliveira, "AQUArIUM-A suite of software measures for HCI quality evaluation of ubiquitous mobile applications," *J. Syst. Softw.*, vol. 136, pp. 101–136, 2018.

[15] N. H. Mansol, N. H. M. Alwi, and W. Ismail, "Managing organizational culture requirement for business continuity management (BCM) implementation using goal-question-metric (GQM) approach," *J. Teknol.*, vol. 78, no. 12–3, pp. 13–22, 2016.

[16] T. Tahir and A. Jafar, "A Systematic Review on Software Measurement Programs," *Front. Inf. Technol. (FIT), 2011*, vol. 73, pp. 39–44, 2011.

[17] D. Aziz, Y. Jamaiah, and Z. A. Zaiha Nadiah, "Software Ageing Measurement Framework Based on GQM Structure," *J. Softw. Syst. Dev.*, vol. 2014, 2014.

[18] P. Berander and P. Jönsson, "A goal question metric based approach for efficient measurement framework definition," *Proc. 2006 ACM/IEEE Int. Symp. Int. Symp. Empir. Softw. Eng. - ISESE '06*, p. 316, 2006.

[19] N. Kumar, R. Dadhich, and A. Shastri, "Quality Models for Web-based Application: A Comparative Study," *Int. J. Comput. Appl.*, vol. 125, no. 2, 2015.

[20] S. Islam and P. Falcarin, "Measuring security requirements for software security," in *Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on Cybernetic Intelligent Systems*, 2011, pp. 70–75.

[21] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, "Linking software development and business strategy through measurement," no. April, pp. 57–65, 2010.

[22] F. Yahya, R. J. Walters, and G. B. Wills, "Using Goal-Question-Metric (GQM) Approach to Assess Security in Cloud Storage," in *Enterprise Security*, Springer, 2017, pp. 223–240.

[23] V. Garousi, M. Felderer, M. Kuhrmann, and K. Herkiloğlu, "What industry wants from academia in software testing?: Hearing practitioners' opinions," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2017, pp. 65–69.

[24] S. A. Muaz, Y. K. Chiam, and B. S. Galadanci, "A GQM-Based Method to Support Elicitation of Sustainability Requirements for Mobile Applications," *Adv. Sci. Lett.*, vol. 24, no. 2, pp. 1268–1272, 2018.

[25] K. Petersen, C. Gencel, N. Asghari, and S. Betz, "An elicitation instrument for operationalising GQM+ Strategies (GQM+ S-EI)," *Empir. Softw. Eng.*, vol. 20, no. 4, pp. 968–1005, 2015.

[26] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, "Linking software development and business strategy through measurement," *Computer (Long. Beach. Calif).*, vol. 43, pp. 57–65, 2010.

[27] M. Villavicencio, "Facts and Perceptions Regarding Software Measurement in Education and in Practice: Preliminary Results," *J. Softw. Eng. Appl.*, vol. 4, no. 4, pp. 227–234, 2011.

[28] M. Scriven, *Evaluation Thesaurus*. SAGE Publications, 1991.

[29] F. H. Al-Tarawneh, "A framework for COTS software evaluation and selection for COTS mismatches handling and non-functional requirements," Universiti Utara Malaysia, Kedah, Malaysia, 2014.

[30] M. Zarour, "Methods to evaluate lightweight software process assessment methods based on evaluation theory and engineering design principles," Ecole de Technologie Superieure (Canada), 2009.

[31] S. T. Acuña, A. De Antonio, X. Ferré, M. López, and L. Maté, "The Software Process: Modelling, Evaluation and Improvement," *Handb. Softw. Eng. Knowl. Eng.*, vol. 1 Fundamen, no. 0, pp. 193–237, 2001.
[32] S. F. P. Mohamad, "A process based approach software certification model for agile and secure environment," Universiti Utara Malaysia, Kedah, Malaysia, 2015.
[33] M. Zarour, "The use of evaluation theory and square standards to develop user needs experience evaluation method," *J. Eng. Technol.*, vol. 6, no. 1, 2018.
[34] *IEEE Standard Dictionary of Measures to Produce Reliable Software*. IEEE, 1996.
[35] J. Dobbins, "IEEE guide for the use of IEEE standard dictionary of measures to produce reliable software," *Inst. Electr. Electron. Eng. New York, NY, USA, Tech. Rep. IEEE Std 982.2–1988*, 1989.
[36] Y. Xia, G. Yan, X. Jiang, and Y. Yang, "A new metrics selection method for software defect prediction," *2014 IEEE Int. Conf. Prog. Informatics Comput.*, pp. 433–436, May 2014.
[37] S. Gu, S. Y. Kim, H. Jeong, and K.-A. Sohn, "Constructing and Exploiting Software Metrics Networks for Software Quality Assessment," *2015 5th Int. Conf. IT Converg. Secur.*, pp. 1–5, 2015.
[38] H. Bahadur and D. Kumar, "Construction of Membership Function for Software Metrics," *Procedia - Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 933–940, 2015.
[39] N. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*. CRC Press, 2014.
[40] A. Stefani and M. Xenos, "Meta-metric evaluation of e-commerce-related metrics," *Electron. Notes Theor. Comput. Sci.*, vol. 233, pp. 59–72, 2009.
[41] R. S. Pressman, *Software Engineering A Practitioner's Approach Seventh Edition*. 2010.
[42] C. Jones, "Evaluating Software Metrics and Software Measurement Practices," *Ma,cook*, pp. 1–100, 2014.
[43] K. V Padmini, H. M. N. Dilum Bandara, and I. Perera, "Use of software metrics in agile software development process," in *Moratuwa Engineering Research Conference (MERCon), 2015*, 2015, pp. 312–317.
[44] Z. G. Dand and H. Vasishtha, "Analysis and Evaluation of Quality Metrics in Software Engineering," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 4, 2015.
[45] R. A. Austin and J. M. Case Jr, "Software metrics useful tools or wasted measurements," DTIC Document, 1990.
[46] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005.
[47] T. Hall and N. Fenton, "Implementing effective software metrics programs," *IEEE Softw.*, vol. 14, no. 2, pp. 55–64, 1997.
[48] P. Bozzelli, Q. Gu, and P. Lago, "A systematic literature review on green software metrics," *VU Univ. Amsterdam*, 2013.
[49] R. Baidya, P. K. Dey, S. K. Ghosh, and K. Petridis, "Strategic maintenance technique selection using combined quality function deployment, the analytic hierarchy process and the benefit of doubt approach," *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 1–4, pp. 31–44, 2018.
[50] M.-C. Tsai, Y.-Y. Chien, and C.-C. Cheng, "Upgrading service quality of mobile banking," *Int. J. Mob. Commun.*, vol. 16, no. 1, pp. 82–115, 2018.
[51] N. Raissi, "Using QFD method for assessing higher education programs: an examination of key stakeholders' visions," *Int. J. Manag. Educ.*, vol. 12, no. 1, pp. 70–93, 2018.
[52] E. Bulut, O. Duru, and S. T. Huang, "A multidimensional QFD design for the service quality assessment of Kansai International Airport, Japan," *Total Qual. Manag. Bus. Excell.*, vol. 29, no. 1–2, pp. 202–224, 2018.
[53] T. T. Sousa-Zomer and P. A. C. Miguel, "A QFD-based approach to support sustainable product-service systems conceptual design," *Int. J. Adv. Manuf. Technol.*, vol. 88, no. 1–4, pp. 701–717, 2017.
[54] R. S. Schillo, D. A. Isabelle, and A. Shakiba, "Linking advanced biofuels policies with stakeholder interests: A method building on Quality Function Deployment," *Energy Policy*, vol. 100, pp. 126–137, 2017.
[55] M. K. Santos, A. M. F. Danilevicz, and R. Tubino, "Environmental service providers assessment: A multi-criteria model applied to industrial waste," *J. Clean. Prod.*, vol. 159, pp. 374–387, 2017.
[56] D. Sarkar and R. Panchal, "Quality Function Deployment (QFD): A Six Sigma Tool for Performance Monitoring of Ready Mixed Concrete," *framework*, vol. 4, no. 2, 2017.
[57] B. Fauziah, Y. Jamaiah, D. Aziz, and H. Abdul Razak, "SPQF: Software Process Quality Factor," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, 2011, pp. 1–7.
[58] K. Nanath and R. Pillai, "A Model for Cost-Benefit Analysis of Cloud Computing.," *J. Int. Technol. …*, pp. 93–118, 2013.
[59] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, "Linking Software Development and Business Strategy through Measurement," no. April, pp. 57–65, 2010.
[60] N. Alshahwan, M. Harman, A. Marchetto, R. Tiella, and P. Tonella, "Crawlability metrics for web applications," in *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on*, 2012, pp. 151–160.
[61] L. Olsina and G. Rossi, "Measuring Web application quality with WebQEM," *MultiMedia, IEEE*, vol. 9, no. 4. pp. 20–29, 2002.
[62] L. Baresi, F. Garzotto, and P. Paolini, "From web sites to web applications: New issues for conceptual modeling," *Int. Conf. Concept. Model. Springer Berlin Heidelb.*, pp. 89–100, 2000.
[63] D. R. Danielson, "Web navigation and the behavioral effects of constantly visible site maps," *Interact. Comput.*, vol. 14, no. 5, pp. 601–618, 2002.
[64] R. Dörner and P. Grimm, "Three-dimensional Beans—creating Web content using 3D components in a 3D authoring environment," in *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, 2000, pp. 69–74.
[65] Z. Zhou, "Evaluating websites using a practical quality model.," 2009.
[66] B. Clifton, *Advanced web metrics with Google Analytics*. John Wiley & Sons, 2012.
[67] A. Kolios, V. Mytilinou, E. Lozano-Minguez, and K. Salonitis, "A comparative study of multiple-criteria decision-making methods under stochastic inputs," *Energies*, vol. 9, no. 7, pp. 1–21, 2016.
[68] R. R. Kumar and C. Kumar, "A Multicriteria Decision-Making Method for Cloud Service Selection and Ranking," in *Advances in Computer and Computational Sciences*, Springer, 2018, pp. 139–147.
[69] I. C. Study and A. Jayant, "An Analytical Hierarchy Process (AHP) Based Approach for Supplier Selection : An Automotive An Analytical Hierarchy Process (AHP) Based Approach for Supplier Selection : An Automotive Industry Case Study," *Int. J. Latest Technol. Eng. Manag. Appl. Sci.*, vol. VIII, no. January, 2018.
[70] A. Awasthi, K. Govindan, and S. Gold, "Multi-tier sustainable global supplier selection using a fuzzy AHP-VIKOR based approach," *Int. J. Prod. Econ.*, vol. 195, pp. 106–117, 2018.
[71] A. Hammouri, "An integrated AHP-topsis methodology to evaluate for adoption cots database components based on usability," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 1, pp. 270–281, 2018.
[72] T. Yaghoobi, "Prioritizing key success factors of software projects using fuzzy AHP," *J. Softw. Evol. Process*, vol. 30, no. 1, 2018.