Metric-based Measurement and Selection for Software Product Quality Assessment: Qualitative Expert Interviews

Zubaidah Bukhari[1], Jamaiah Yahaya[2]
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, 43600
Bangi, Selangor, Malaysia

Aziz Deraman[3]
School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu, Kuala Terengganu,
Terengganu, Malaysia

***Abstract*—A systematic and efficient measurement process can assist towards the production of quality software product. Metric-based measurement method often used to assess the product quality. Currently several hundreds of metrics have been proposed by previous researchers. However, there is no specific and structured mechanism for metrics selection process. Lack of awareness, knowledge and experience lead to selecting inappropriate and unsuitable metrics for assessment of software product quality done by the practitioners and stakeholders in the industry. Literature study found that the existing selection models are irrelevant and insufficient for assisting and supporting metrics selection process in which it should consists of criteria, and systematic and practical methods of selection process. A qualitative interview was conducted involving 12 experts and practitioners to reveal current issues in software measurement, to identify elements relevant for software metric selection process and to identify the appropriate and valid software metric selection criteria. Finding from this expert interview revealed important input from industry which are: Five main issues in software measurement, six elements associated with metric selection process and 13 criteria relevant for software metric selection.**

Keywords—Software product metric; metric selection criteria; software quality; software measurement; selection process; qualitative study

I. Introduction

Systematic measurement is an important procedure to ensure and maintain the quality attributes of product deliverables to customers or users. Making the measurement process works in organisation requires collecting correct and relevant metrics based on organisation's objectives and goal. In order to obtain metrics and measurements that address the needs of organizations, the measurement process must be structured, systematic and guided. Software measurement based on quality model and software metric has been introduced and investigated by previous researchers such as Fernando Pinciroli, Yahaya & Aziz, Bouwers, Deursen & Visser, and Ahmad Fadzlah & Deraman [1][2][3][4]. Current and available quality models developed by previous researchers offered general and imprecise criteria for software quality assessment [2][5][6][7].

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" [8]. There are many studies that proposed different types of metrics such as security metrics [9], usability metrics [4], and web application metrics [10][11]. Software metrics will affect the measurement program and eliminating inaccurate metrics will improve software performance and reduce wastage[10]. However, there is no consensus on which metrics are relevant and worth for selection [2][5][12][13].

This research is funded partly by Universiti Kebangsaan Malaysia, The Research University Grant (AP-2017-005/3).

A number of previous researches [14][15][16] stated use of standards as a success factor in metric selection (e.g. ISO/IEC 15939 [17], ISO/IEC 9126[18], ISO/IEC 25000 [19] and ISO/IEC 14598 [20]). However, there is still no consensus in the software measurement area on which standard(s) to use. Most standards present only quality metrics or basic project management metrics such as size (Function Points, cyclomatic complexity etc.).

Studies have revealed that after the second year of implementing measurement metrics, 50%-80% of these measurements are not maintained [14][15]. It is also found that a very high failure rate in metric implementation which is 66.7%. Even though software metric has been introduced by previous researchers, managing and maintaining the assessment program is a challenge and mostly because of lack of commitment from staff [16], no guideline for implementation [17], lack of experts [15] and also there is no metric repository for effective and efficient metric selection to the practitioners and stakeholders [2][5][12].

This paper presents the qualitative expert interviews and findings on software product quality assessment based on metric-based measurement from industrial perspectives. It starts the discussion with background and related work in Section 2, and continues with the qualitative interview in Section 3. Section 4 discusses the analysis and findings, and Section 5 presents the result and discussion. This paper concludes with a conclusion in Section 6.

II. Background and Related Works

This section discusses the current issues, challenges, concepts and related works regarding metric-based measurement and selections.

A. *Software Quality Models*

Literature study has revealed several quality models available to measure and assess software product quality such models are: McCall [21], Boehm [22], FURPS [23], ISO 9126 [18], ISO 25010 [19], Pragmatic Quality Factor or PQF [24]. Current user's requirements and expectation demand for software quality model that is easy, accurate and practical to use not only for the developers and practitioners but also to be used by the users, customers and stakeholders [12].

McCall model is among the earliest software quality model and is known as factor criteria metric [25]. It consists of integration of 11 factors and 23 criteria for software product quality. The main contribution of this model is the relationship between quality characteristic and metrics even though there is a claim saying that not all metrics are objective to be measured. Boehm model was developed based on McCall with additional characteristics which cater for maintenance and system utility.

ISO/IEC 9126 is a well-known software quality model aims for quality standardisation of software product. ISO/IEC 9126 defines quality in six main characteristics which are functionality, reliability, usability, efficiency, maintainability and portability. These characteristics are further broken down into sub characteristics [3] [18]. It has been invented since 1991 and today, it is still being used in researchers that work with software product quality. However, at the same time it has the disadvantage of not showing clearly how these quality characteristics can be measured and the model only focusing on developer view of the software [3]. In 2011, ISO/IEC 9126 was reviewed and a new international standard was introduced for software product quality assessment, ISO/IEC 25000 (System and Software Quality Requirements and Evaluation or SQuaRE). ISO/IEC 25000 is the result of the evolution of several other standards; specifically, from ISO/IEC 9126, which defines a quality model for software product evaluation and assessment. The product quality model defined in ISO/IEC 25010 comprises the eight quality characteristics which are Functional Suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability [19] as listed in Table I. This standard defines a product quality model composed of characteristics (which are further subdivided into sub characteristics) that relate to static properties of software.

Later model of software quality is called Pragmatic Quality Factor or PQF. It was developed based on ISO 9126 model and added two more attributes: integrity and impact [3]. This model divides attributes into two main classifications which are behavioural attribute and impact attributes. The attributes are broken down into several sub attributes and metrics. PQF defines behavioural attribute that comprises of usability, efficiency, functionality, maintainability, reliability, portability and integrity. While the impact attribute comprises of user perception and user requirement. This model has included user factors and these characteristics were not included in previous models. User factors are considered essential and important since user nowadays are more demanded and recognised for good quality software and thus relevant to their perspectives for quality. Different users may have different perspective and requirement toward quality product. Therefore, in PQF model comprises of weight value for each of the quality characteristics to represent individual and organisational need on quality measurement and assessment [3].

TABLE I. ISO25010 QUALITY MODEL: SOFTWARE PRODUCT QUALITY [19]

Characteristic	**Sub Characteristic**
Functional Suitability	• Functional Completeness • Functional Correctness • Functional Appropriateness
Performance Efficiency	• Time Behaviour • Resource Utilization • Capacity
Compatibility	• Co-existence • Interoperability
Usability	• Appropriateness Recognisability • Learnability • Operability • User Error Protection • User Interface Aesthetics • Accessibility
Reliability	• Maturity • Availability • Fault Tolerance • Recoverability
Security	• Confidentiality • Integrity • Non-Repudiation • Accountability • Authenticity
Maintainability	• Modularity • Reusability • Analysability • Modifiability • Testability
Portability	• Adaptability • Installability • Replaceability

B. *Software Measurement*

Measurement is essential and important in everyday life as well as in scientific and engineering discipline. Measurement is the assignment of a number to a characteristic of an object or event, which can be compared with other objects or events [26]. It cannot be done if the underlying measures are not objective but rather subjective.

The main objective of software development in organisation is to produce good quality software products. Measurement can be used to measure or assist in measuring the product quality. Without measurement, assessment and evaluation are considered as subjective matter and unable to compute and compare. Metrics or measures provide indirect measuring towards software quality [27] and enable the quality to be quantifiable and countable [28].

Software metrics are important for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses. In this case, metrics are used to measure characteristics or attributes of software product [3]. Using certain rules will illustrate meaning and guidance regarding software's characteristics and behaviour. Mostly all quality model discussed in this paper are embedded with measurements and metrics to quantify and assess software product quality. As an example:

Attribute: Functionality
Sub attributes -> {metrics}
Sub attribute1:
Accuracy -> {M1, M2, M3}
Accuracy -> {M1=Incomplete result, M2=Incorrect result. M3=Unexpected results issued}
Sub attribute2:
Interoperability-> {M4, M5}
Interoperability-> {M4=Data format, M5=Data exchange}
Sub attribute3:
Suitability-> {M6, M7, M8, M9}
Suitability -> {M6=Functional Implementation coverage; M7=Functional specification stability; M8=Functional implementation correctness; M9=Functional implementation completeness}

The structure of this hierarchy (the attribute, sub attribute and metrics) is shown in Fig. 1.

The decomposition of sub attributes is at Second Level of this hierarchy. Functionality is considered as unmeasurable characteristic and thus involves indirect measurement. In order to convert the unmeasurable characteristic to a measurable characteristic, sub attributes of functionality are decomposed into lower level of hierarchy which is the third level. At the third level, the sub attributes are decomposed into metrics which are used to measures software product quality.

Various software metrics have been proposed by previous researchers to support assessment of software product quality and also to predict quality and other maintenance activities [29][30][31]. However, the emergent of various metrics has introduced new challenge to the practitioners and stakeholders in order to select and use the appropriate metric that meet organisational goal and objectives. Some metrics are too complex and difficult to understand and use [5][13][32].

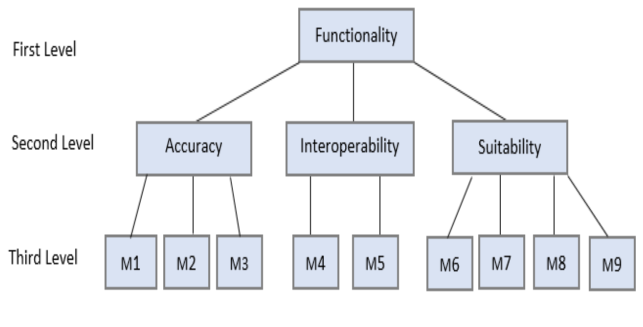

Fig. 1. Structure of Decomposition and Hierarchy.

Literature study has identified previous studies which focused and proposed specific criteria for metric selection. Such criteria are measurement theory [33][34], IEEE standard [35], Kaner Framework [36] and search-based approach[37]. Most of the criteria are applicable for internal measurement. However, studies have shown that software that meet and fulfil the internal measurement criteria do not guaranteed the successful and effectiveness of the software from user's perspective[38][39]. In order to ensure the quality of the software, measurement from external view that focuses on user acceptance and satisfaction are also required [5][11][12]. Previous studies have revealed that user acceptance and satisfaction are the main factors to foresee the successful of software product [40][41].

Our study focuses on external measurement based on software metrics as the scope of this study. We do not cover the internal metrics for internal measurement as discuss in this section.

C. Issues and Challenges in Software Measurement

Software evolution has seen the emergent of different types of software for different purposes. Nowadays, software has become very important in everyday life of everyone and thus the quality of the software is also an essential issue to be highlighted and focused. Even though several software quality models have been introduced and developed such as McCall, Boehm, ISO9126, PQF and ISO25010 as discussed in this paper, but the implementation of measures and metrics were not being mentioned and discussed in detail. A good measurement program has appropriate and relevant measurement metrics [10], comprehensive data collected [42], and consistent with the organisational goal [43]. Several issues are still underpinning in this matters.

1) Lack of commitment: The benefits and advantages for measurement program must be explained and accepted throughout the organisation. Without commitment from the organisation top level and staff, it is difficult to obtain accurate and up-to-date data on measurements. This also links with the commitment from top management to support the measurement program [43]. Thus, only relevant and appropriate software measurement metrics will be collected to ensure the organisational goal is achieved.

2) Absence of guideline and standard: The information, communication and technology strategy was developed during the planning phase for identifying the requirements and specification for ICT implementation. The ICT framework and strategy have been revised accordingly based on new additional and modified requirements. Current measurement process and program do not provide the mechanism for maintaining the measurement framework for organisations [32]. Therefore, there is a requirement to have the guideline and mechanism to support the organisation's software measurement and assessment program to support ICT strategy and organisation's goal.

3) Limited of expert resources: Literature study revealed that one of the reasons for failing in software measurement was due to limited expert in selecting metrics relevant and appropriate with organisational strategy and goal [44]. Limitation in expert resources may be because of lack of graduate with knowledge or experience in software measurement area. The measurement topic and subject are only offered for graduate study and not during undergraduate study [44].

4) Limited measurement metric resources: Previous works proposed several numbers of metrics for software measurement and assessment but the application and implementation of these metrics in real environment is ambiguous without systematic guideline of the usage [2][12][45]. There is no guideline for metric selection based on organisational strategy and goal. Thus, there is requirement to gather all the metrics with associated mechanism to guide in the implementation and application. This will encourage reusable of metrics in similar purposes and goals.

III. Qualitative Expert Interview

The objectives of this study were to identify current practices and issues related to software metric and measurement, to identify elements needed for software metric selection process, and to identify software metrics selection criteria from real industrial input.

A. The Protocol Design

The interview protocol was designed based on qualitative approach and divided into three parts as follows:

1) Part I: Introduction to Metric: This part is to discover information regarding the implementation and the use of metric in software development and assessment. It consists of eight questions in related to software metric practices, the importance of software metric and success factors for metric implementation. The questions are:-

- What is your opinion regarding software metric?
- What are the examples of software metrics that you use during software development?
- How would software metrics benefit to software development activity?
- In what way metric is used during software development?
- Software development involves several phases. In which phase the metrics can be used and applied?
- There are a few software metrics currently available such as line of code (LOC), cyclomatic complexity (CK), Halstead metrics, fog index and fin-in/fan-out metric. Do you use these metrics during software development or assessment?
- If No (for question (f)), why?
- What are the factors that influence of not using software metrics during development and assessment process?

2) Part II the elements: This part of the interview protocol requires to investigate the elements needed in software metric evaluation and selection process. It consists of eight questions associated with criteria for metric selection.

- Who will use software metrics in your department/organisation?
- What is the technique used for metric assessment and selection?
- Is there necessary to have standard in software metric selection?
- What is the method used to collect data regarding assessment goal setting in your department/organisation?
- What is the synthesis technique used during metric selection process to measure the appropriateness of the metric?
- Do you think that each metric should be assigned with appropriate suitability level?
- Is there any list of software metric for Malaysian Public Sector?
- Is there any metric selection repository to be used among public sector organisation?

3) Part III software metric selection criteria: Part III consists of three main questions related to components and techniques in software metric evaluation and selection. Previous studies have proposed and suggested numerous metrics for software assessment and evaluation. At the same time, the issue arises: how to evaluate and select appropriate metric based on organisation’s requirements? The questions asked in this part of the protocol include:

- In literature study, we discovered several criteria or characteristics for metric evaluation. In your opinion, what are the appropriate criteria for evaluating software metric in the industry?
- How would these criteria and characteristic be used in metric evaluation process?
- Can you think of any other suitable criteria for metric evaluation?

This study has invited two senior university academicians to involve and participate as pilot study. They are chosen based on their expertise in software engineering and qualitative method. The academic experts played as a role to review and validate the protocol. The protocol which consists of questions as mentioned as Part I, Part II and Part III were corrected and refined before the actual interviews were conducted.

B. The Sampling

This study was carried out through series of interviews with 12 selected expert informants. The selection criteria of informants are based on expertise in software engineering and more specific in evaluation, measurement and testing. The duration of working experience also considered as selection criteria where at least they have working experience more than five years in the industry. The duration of working experience for the informants is based on the years suggested by [46]. Table II shows the informant's background who involve in this interview. Majority of the informants have working experience more than 10 years in the industry. 83% of the informants are working in public sector and 17 % are working in private sector. In this paper, the informants are labelled as A, B, C, D, E, F, G, H, I, J, K and L respectively.

In this qualitative study, we found it was hard to find informant or people who have knowledge directly on software metric either in public or private sector. Thus, the informants were selected based on their experience in software evaluation and software metric throughout their working experience. Most of the informants are from public sector because the scope of this study is in public sector.

TABLE II. INFORMANT'S BACKGROUND

Informant	Job Description	Expertise	Years of Working Experience	Sector
A	Researcher	Software Evaluation	>20 years	Public
B	Researcher	Software Evaluation	>30 years	Public
C	Researcher	Software Project Management	>15 years	Public
D	Researcher	Software Evaluation	>15 years	Public
E	Researcher	Software Metric	>20 years	Public
F	Researcher	Software Metric	>30 years	Public
G	Software Engineer	Software Testing	>5 years	Semi-Government
H	Software Development Manager	Software Evaluation	>15 years	Private
I	Software Development Manager	Software Evaluation	>20 years	Private
J	Project Manager	Software Testing	>20 years	Public
K	Project Manager	Software Testing	>20 years	Public
L	Software Engineer	Software Testing	>10 years	Public

IV. Analysis and Findings

A. *The Analysis*

The analysis was carried out in five steps which adapted from Creswell [47]. The steps are:

1) Step 1: Organize and prepare the data for analysis. This involves transcribing interviews, optically scanning material, typing up field notes, cataloguing all of the visual material, and sorting and arranging the data into different types depending on the sources of information.

2) Step 2: Read the whole text or scripts. This step provides a general sense of the information and an opportunity to reflect on its overall and clear meaning of the text.

3) Step 3: Coding. This is the process of organising the data by connecting chunks (or text or image segments) and writing a correct word representing a specific category [48]. It involves taking text data or pictures gathered during data collection, segmenting sentences (or paragraphs) or images into categories, and labelling those categories with a term, often a term based in the actual language of the participant.

4) Step 4: Interpreting the data. Use the coding process to generate a description of the setting or themes for analysis. Advance how the description and themes will be represented in the qualitative narrative. The most popular approach is to use a narrative passage to convey the findings of the analysis. This might be a discussion that mentions a chronology of events, the detailed discussion of several themes (complete with subthemes, specific illustrations, multiple perspectives from individuals, and quotations) or a discussion with interconnecting themes.

5) Step 5: Validation of Findings. The data analysis is finalised by validation process to ensure the findings are correct and accurate. The process is carried out with the experts to validate and verify the findings.

B. *Findings*

Twelve interview scripts have gone through verification analysis and texts were read repeatedly to understand the implicit intent. From the analysis, 112 codes have been identified and created. The codes were sorted based on similar meaning or categorisations. There are 24 codes obtained through the analysis process. After the theme categorisation process, codes are grouped into three which are issues in software measurement, elements for software metric selection, and criteria for software metric selection.

The content analysis discovered several codes associated with group and categorisation. In group one which is issues in software metric, the analysis identifies five codes and for group two which is about elements for software metric selection, six codes are grouped in this category. While in the third group which is related to metric selection criteria, the analysis reveals 13 codes from the coding analysis and theme representing process. Table III shows the findings.

1) Issues in software measurement: Based on the findings of this study, it revealed that there are still issues and challenges in implementing software measurement. The view and opinion are similar in government and private sector and they revealed lack of commitment, no guidelines or systematic procedure, limited expert resources, and limited metric resources gave impact and consequence toward software measurement and assessment program. The frequency analysis shows that no guideline or systematic procedure achieve 16 times more often given by the informants. This means that no guidelines are the highest and important issue given by informants of this study. While 15 times were given and highlighted by informants on the issues of lack of commitment and experts in metrics selection. Furthermore, the informants gave 13 times highlighted on lack of software metric resource and 10 times occurrence in the scripts for non-compliance to goal and objective. The detail frequency analysis is shown in Table IV.

2) The elements for software metric selection: In the effort of preparing the structured approach in software metric selection, informant's views and opinion were asked regarding the necessary elements during the selection process. The identified elements will be used as the main elements or components needed in the structured software metric selection model. Findings for Part B of the interview instrument are shown in Table V. It shows that evaluation criteria is the most popular element identified by the interview informants where it appears 14 times more frequent in the interview scripts. While the second highest in term of times frequent are the target and data collecting technique with 12 times. Next, is standard reference with 10 times highlighted by informants and follows by synthesis technique and evaluation process with eight times highlighted and appeared in the scripts. The detail frequency analysis is illustrated in Table V.

C. Criteria for Software Metric Selection

Informants of this survey expressed their views and opinions on essential criteria for software metric selection process. Based on frequency shown in Table VI, measurement scale received high frequency which is 16 times given by informants. This shows that informants highlighted 16 times saying that measurement scale is the important criteria during evaluation of metric selection. Second highest frequency is measurement independence (14 times) and third highest frequency is cost and programming language independence (13 times). This follows by automation (12 times) and accuracy and simplicity with 11 times. Meanwhile, environment, feedback and applicability appear 10 times occurrence in the informant's scripts. The last three criteria which are green ability, type of users and comparable receive nine times occurrence in the informant scripts, respectively.

TABLE III. CONTENT ANALYSIS ACCORDING TO GROUP AND CATEGORISATION

Group 1	Issues in software measurement	1. Lack of commitment 2. Lack of expert 3. No metric resource 4. No guideline 5. Non-compliance to organisation's objective
Group 2	Elements for software metric selection	1. Target 2. Selection criteria 3. Reference standard 4. Data collection technique 5. Synthesis technique 6. Evaluation process
Group 3	Metrics Selection Criteria	1. Measurement scale 2. Measurement independence 3. Automation 4. Cost 5. Accuracy 6. Simplicity 7. Environment 8. Type of users 9. Programming Language Independence 10. Feedback 11. Comparable 12. Applicability 13. Green ability

TABLE IV. FREQUENCY OF ISSUES IN THE INFORMANT SCRIPTS

Issues	**Informant**												**Total**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	
No guideline	1	1	1	1	2	1	1	1	1	2	2	2	16
Lack of commitment	1	1	1	1	1	1	1	1	1	2	2	2	15
Lack of Expert Resources	1	1	1	1	1	2	0	1	1	2	2	2	15
Lack of metric resources	1	1	1	1	1	1	0	1	1	2	2	1	13
Non-compliance to goal & objective	0	0	0	1	1	1	0	2	1	2	1	1	10

V. Results and Discussion

Findings from this study shows that failure in measurement program still exists and is relevant in today's software quality challenge. The failure of this program causes by lack in commitment among software developers, practitioners and stakeholder, lack of guideline, limited number of expertise in this area, lack of metric resources and non-compliance to organisational objective. The first part of the interview reveals that we still need a new model for measurement program, a systematic guideline for metric selection and a repository for available software metric which can be accessed by many people in this area and compliance with the organisational goal and objectives.

While the second part of the interview instruments and analysis revealed that the essential elements for selection metric are: target, selection criteria, reference standard, data collection technique, synthesis technique, and evaluation process as demonstrated in Table V. Even though some of these items are not being practiced currently by the informants in the industry but they agree that these elements are needed to support the selection and evaluation process. Lack of standard and structured approach or mechanism will avert the successful of measurement program in organisation.

Furthermore, this expert interview study discovers and verifies that selection criteria supports organisation in metric selection process based on certain criteria and unique characteristic of metric. The identified criteria for metric selection process can be used in systematic software evaluation. Organisations and stakeholders may understand more on the importance of the selected metrics suitable and appropriate for their requirements based on organisation's goal and objectives. The verified criteria are shown in Table VI.

Based on these findings and also supported by literature study, the definition and detail description on each of the criteria are presented in Table VII.

TABLE V. FREQUENCY OF ELEMENTS IN THE INFORMANT SCRIPTS

Elements in Metric Selection	**Informant**												**Total**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	
Evaluation Criteria	1	1	1	1	2	2	1	1	1	1	1	1	14
Target	1	1	1	1	1	1	1	1	1	1	1	1	12
Data Collection Technique	1	1	1	1	1	1	1	1	1	1	1	1	12
Reference Standard	1	1	1	2	1	2	1	0	0	0	0	1	10
Synthesis Technique	1	1	1	1	1	2	1	0	0	0	0	0	8
Evaluation Process	1	1	1	1	1	1	1	0	0	0	0	1	8

TABLE VI. FREQUENCY OF CRITERIA IN THE INFORMANT SCRIPTS

Criteria for metric selection	**Informant**												**Total**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	
Measurement scale	2	1	1	2	2	2	1	1	1	1	1	1	16
Measurement independence	1	1	1	1	1	3	1	1	1	1	1	1	14
Cost	2	1	1	1	1	1	1	1	1	1	1	1	13
Programming language independence	1	2	1	1	2	2	1	1	1	0	0	1	13
Automation	1	1	1	0	1	2	1	1	1	1	1	1	12
Accuracy	1	1	1	0	1	3	1	1	1	0	0	1	11
Simplicity	1	1	1	1	1	2	1	1	1	0	0	1	11
Applicability	1	1	1	0	2	1	1	1	1	0	0	1	10
Environment	1	1	1	1	1	1	1	1	1	0	0	1	10
Feedback	1	1	1	0	2	1	1	1	1	0	0	1	10
Type of users	1	1	1	0	1	1	1	1	1	0	0	1	9
Comparable	1	1	1	0	1	1	1	1	1	0	0	1	9
Green ability	1	1	1	0	2	1	1	1	0	0	0	1	9

TABLE VII. THE DESCRIPTION OF METRIC CRITERIA

	Criteria	**Description**
1	Measurement scale	Scale that being used for categorising and measuring certain metric. Four main scales normally used are: nominal, ordinal, interval and ratio.
2	Measurement independence	The ability to obtain same result for different users. The consistency and stability of the metric.
3	Automation	The effort of measuring using support tool.
4	Cost	Referring to the cost implication in metric. Simple metric will reduce cost and complex metric will increase cost.
5	Accuracy	The accuracy of the measure.
6	Simplicity	Metric should be easy to be used and understood by the users.
7	Environment	Is the metric require control environment such as in lab? Or in the uncontrolled environment such as at home?
8	Type of users	Type of users involve in the metric evaluation. If larger target group or users, more cost will be needed.
9	Programming language independence	Metric should be independent from any programming language or any specific programming syntax.
10	Feedback	Metric should provide further information or prediction on product quality.
11	Comparable	Metric should be able to compute and compare to understand the real situation.
12	Applicability	Metrics should be applied and appropriate for certain phase in software life cycle
13	Green ability	Metric should support green with minimum or less effect on environment.

VI. CONCLUSION

This paper has presented the findings from qualitative expert interview on three main issues which are issues in software measurement, elements for software metric selection process, and metrics selection criteria. The aims are to identify the current practices in the industry, issues and challenges in metric selection and evaluation, metric selection and evaluation process in industry specifically in public sector. The empirical study was conducted in Malaysia that involved 12 experts and practitioners in software evaluation, testing and measurement. The study has discovered five main issues related to software measurement face by the industry as discussed in this paper. Furthermore, it revealed 13 essential criteria and six main elements for software metric selection process. This finding will be applied and used in construction of the Structured Software Metric Selection Model as our future work.

VII. FUTURE WORK

For decades, measurement and metrics is important activities due to the growing interests of software companies in the improvement of the productivity and quality of delivered products. Future research is needed to explore the potentials of measurement program to have a software metric selection model which integrate software metric selection elements and criteria, systematic software metric selection guideline and a comprehensive repository for available software metrics which compliance with the organisational goal and objectives. Last, software metrics selection process needs to adapt the model, guideline and repository in order to ensure software product quality.

ACKNOWLEDGEMENT

This research is funded by Universiti Kebangsaan Malaysia, the Arus Perdana Research University Grant (AP-2017-005/3).

REFERENCES

[1] F. Pinciroli, “Improving software applications quality by considering the contribution relationship among quality attributes,” Procedia Comput. Sci., vol. 83, pp. 970–975, 2016.

[2] E. Bouwers, A. van Deursen, and J. Visser, “Towards a catalog format for software metrics,” in Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics, 2014, pp. 44–47.

[3] J. Yahaya and A. Deraman, “Measuring unmeasurable attributes of software quality using Pragmatic Quality Factor,” in Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, 2010, vol. 1, pp. 197–202.

[4] Z. Zali, “An initial theoretical usability evaluation model for assessing defence mobile e-based application system,” in 2016 International Conference on Information and Communication Technology (ICICTM), 2016, pp. 198–202.

[5] A. Stefani and M. Xenos, “Meta-metric evaluation of e-commerce-related metrics,” Electron. Notes Theor. Comput. Sci., vol. 233, pp. 59–72, 2009.

[6] H. Rashidi and M. S. Hemayati, “Software Quality Models: A Comprehensive Review and Analysis,” J. Electr. Comput. Eng. Innov., vol. 6, no. 1, 2019.

[7] P. Nistala, K. V. Nori, and R. Reddy, “Software Quality Models: A Systematic Mapping Study,” in Proceedings of the International Conference on Software and System Processes, 2019, pp. 125–134.

[8] J. E. Gaffney Jr, “Metrics in software quality assurance,” in Proceedings of the ACM’81 conference, 1981, pp. 126–130.

[9] N. Shahriza, A. Karim, A. Albuolayan, T. Saba, and A. Rehman, “The practice of secure software development in SDLC : an investigation through existing model and a case study,” Secur. Commun. Networks, vol. 9, no. November, pp. 5333–5345, 2016.

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

[11] J. Yahaya, A. Deraman, A. Hamdan, and Y. Jusoh, "User-Perceived Quality Factors for Certification Model of Web-Based System," waset.org, vol. 8, no. 5, pp. 632–638, 2014.

[12] Y. Jamaiah and D. Aziz, "Software Certification Modeling: From Technical to User Centric Approach.," Aust. J. Basic Appl. Sci., vol. 7, no. 8, pp. 9–18, 2013.

[13] J. M. Voas and D. R. Kuhn, "What Happened to Software Metrics?," Comput. (IEEE Comput., vol. 50, no. 5, 2017.

[14] M. Staron and W. Meding, "Measurement-as-a-Service–A New Way of Organizing Measurement Programs in Large Software Development Companies," in Software Measurement, Springer, 2015, pp. 144–159.

[15] M. Staron et al., "Measuring and visualizing code stability - A case study at three companies," Proc-Jt. Conf. 23rd Int. Work. Softw. Meas. 8th Int. Conf. Softw. Process Prod. Meas. IWSM-MENSURA 2013, pp. 191–200, 2013.

[16] V. Antinyan et al., "Identifying risky areas of software code in Agile/Lean software development: An industrial experience report," in 2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE), 2014, pp. 154–163.

[17] ISO/IEC 15939, ISO/IEC 15939 International standard 1st edition 2002: Software engineering - software measurement process, 2002.

[18] ISO/IEC, "ISO/IEC 9126-2 - Software Engineering - Product quality-Part 2: External Metrics," 1991.

[19] ISO/IEC 9126-2 - Software Engineering - Product quality- Part 2: External Metrics, ISO, Geneva, Switzerland, 2001.

[20] ISO/IEC 14598 Information Technology - Software Product Evaluation - Parts 1-6, ISO, Geneva, Switzerland, 2004.

[21] G. F. McCall, J.A., Richards, P.K., Walters, "Factors in Software Quality," US Rome Air Dev. Cent. Reports, US Dep. Commer. USA, vol. I. II, and III, 1977.

[22] M. Boehm, B. W., Brown, J.R., Kaspar, H., Lipow, M., McLeod, G., "Characteristics of software quality," 1978.

[23] R. B. Grady, Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, Englewood Cliffs, Nj, USA, 1992.

[24] A. Deraman, Memburu Kualiti Perisian (Inaugural Speech). UKM Publisher. ISBN, 2010.

[25] S. Wagner, Software product quality control. Springer, 2013.

[26] N. E. Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach, Revised." 2014.

[27] I. Sommerville, Software engineering (10th edition). 2016.

[28] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," IEEE Std, vol. 610121990, no. 121990, p. 3, 1990.

[29] A. F. Allah, L. Cheikhi, R. E. Al Qutaish, and A. Idri, "E-government portals best practices: a comprehensive survey," Electron. Gov. an Int. J., vol. 11, pp. 101–132, 2014.

[30] L. Hussain and M. Kadhim, "Design and Implementation of a Website Usability Model," muc.edu.iq, vol. 2014, no. 21, pp. 31–58, 2014.

[31] P. Morrison, D. Moye, R. Pandita, and L. Williams, "Mapping the field of software life cycle security metrics," Inf. Softw. Technol., vol. 102, no. May, pp. 146–159, 2018.

[32] 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.

[33] L. Briand, S. Morasca, and V. Basili, "Property-based software engineering measurement," Softw. Eng. IEEE Trans. Softw. Eng., vol. 22, no. 1, 1996.

[34] J. C. Jacobs and J. J. M. Trienekens, "Towards a metrics based verification and validation maturity model," Softw. Technol. Eng. Pract. 2002. STEP 2002. Proceedings. 10th Int. Work., pp. 123–128, 2002.

[35] IEEE Computer Society, "Standard for a Software Quality Metrics Methodology," Revis. IEEE Stand., vol. 1998, pp. 1061–1998, 1998.

[36] C. Kaner and W. P. W. Bond, "Software engineering metrics: What do they measure and how do we know?," in 1oth International Software Metrics Symposium Metrics, 2004, vol. 8, pp. 1–12.

[37] M. Ó Cinnéide, I. Hemati Moghadam, M. Harman, S. Counsell, and L. Tratt, "An experimental search-based approach to cohesion metric evaluation," Empir. Softw. Eng., vol. 22, no. 1, pp. 292–329, 2017.

[38] J. Yahaya, A. Deraman, and A. R. Hamdan, "Continuously ensuring quality through software product certification: A case study," in 2010 International Conference on Information Society, 2010, pp. 183–188.

[39] D. Aziz, Memburu Kualiti Perisian (Inaugural Speech). Bangi: Penerbit Universiti Kebangsaan Malaysia, 2011.

[40] M. S. Janita and F. J. Miranda, "Quality in e-Government services: A proposal of dimensions from the perspective of public sector employees," Telemat. Informatics, vol. 35, no. 2, pp. 457–469, 2018.

[41] N. Veeramootoo, R. Nunkoo, and Y. K. Dwivedi, "What determines success of an e-government service? Validation of an integrative model of e-filing continuance usage," Gov. Inf. Q., vol. 35, no. 2, pp. 161–174, 2018.

[42] A. Mughal, "A Framework for Implementing Software Measurement Programs in Small and Medium Enterprises," 2017.

[43] 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.

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

[45] M. Abdellatief, A. B. M. Sultan, A. A. A. Ghani, and M. A. Jabar, "A mapping study to investigate component-based software system metrics," J. Syst. Softw., vol. 86, no. 3, pp. 587–603, Mar. 2013.

[46] N. H. Zulkifli Abai, J. Yahaya, A. Deraman, A. R. Hamdan, Z. Mansor, and Y. Yah Jusoh, "Integrating Business Intelligence and Analytics in Managing Public Sector Performance: An Empirical Study," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 9, no. 1, p. 172, 2019.

[47] J. W. Creswell and J. D. Creswell, Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications, 2017.

[48] G. B. Rossman and S. F. Rallis, Learning in the field: An introduction to qualitative research. Sage, 2011.