Goal Oriented Measurement for Software Sustainable Evaluation Metric Focused on Environmental Dimension

Ruzita Ahmad, Azham Hussain, and Fauziah Baharom

Universiti Utara Malaysia, Sintok Kedah, rita_azura08@yahoo.com, {azham.h, fauziah}@uum.edu.my

ABSTRACT

Sustainability is a complex concept that investigated in interdisciplinary dimension which are environment, economic, and social. Software sustainability has moved towards new paradigms of research and it is claimed as still immature due to lack of integration on these three dimensions. Currently, there are studies on software sustainability evaluation that defined the evaluation criteria. However, most of the studies are lack of integrating the three dimensions of software sustainabiltiy. In addition, the evaluation goals are also not clearly defined. Therefore, the objective of this study is to define the evaluation goals for each proposed characteristic and sub-characteristic with focused to environmental dimension. Goal Question Metric (GQM) is used as a method to identify the correct goals in this study. The adaptation of goal oriented measurement can contribute to define the precisely goals by determining the purposes, perspectives, point of views in the following context of environment with respect to achieve software sustainability.

Keywords: Software sustainability, Goal Oriented Measurement (GOM), evaluation criteria, environmental dimension

I INTRODUCTION

Sustainability is the development that meet the needs of the present generation without compromising the ability of future generation to meet their own needs (Brundtland Commission Report, 1987). Several years ago the concept of sustainability has been practiced in various domain such as manufacturing, construction, restoration of natural disasters, soil and erosion, ecosystems and biodiversity, and so forth. Thus, the transformation view of sustainable development has a strong commitment to the social equity, with a sight that linked to livelihood, health, welfare, resources, economic and political decision are connected each other. Dealing to the context of software engineering, sustainability has been highlighted in software development and known as software sustainability (Koziolek et al., 2011). Software sustainability refers to the software development in which the resources use aims to meet the needs of present generation until future with integrating the aspects of environment, economic and social towards long living software (Ahmad et al., 2015).

Currently, most of the software products and processes are developed with either an environment, economic or social benefits and not intended to serve with either an environment, economic and social purposes. For instance, the software development nowadays are merely focused on environment and social aspects and did not highlighted the economic aspects as the important element (Penzenstadler et al., 2013; Koziolek et al., 2011). In addition, some of developers focuses on economic and social aspect and ignoring the environment aspect (Penzenstadler et al., 2013; Calero et al., 2013). These development trends are the main reason why the important of sustainability in software engineering. Therefore, the integration of environment, economic and social dimension in software development can support the production towards long living software.

In order to monitor the production of software towards long living software, the needs of measurement mechanism is significantly to guide the software product and process to meet the needs of sustainability requirements. Several software sustainability evaluations has been developed recently. The best known practices models in the literatures such as model proposed by Sarkar et al. (2008), Koziolek et al. (2011), Durdik et al. (2012), Kocak et al. (2012), Venters et al. (2013), Penzenstadler et al. (2013) and Penzenstadler et al. (2014). The models are built with rich of important features towards long living software and claimed to fulfill the sustainability requirements in their assessment mechanism. Unfortunately, most of them are regardless to show the systematic measurement process that only focused on what need to be measured instead of who, when, where, why, and how to measure. Furthermore, the goals for each proposed features in their models does not clearly defined based on the criteria that they aimed to achieve. Therefore, this study intend to improve the limitations of previous works in defining the goals for the proposed features of software sustainability evaluation by using Goal Question Metric (GQM) with focused on environmental dimension.

II LITERATURE REVIEW

This section presents the overview of software sustainability evaluation with discusses the best practices of software sustainability evaluation model in the literature. The outline is continued to the goal oriented measurement for expressing the objective in this study.

A. Software Sustainability Evaluation

Software sustainability evaluation has been expressed by several researchers in different ways. Most of them are based on Tripple Bottom Line (TBL), System Life Cycle (SLC), Life Cycle Assessment (LCA) approach and also several quality models to express their sustainability measurement models. Beneficially, most of them aimed to achieve software sustainability in their own ways based on their theories and practices, activities, opinions and experiences. The best known model of software sustainability metric evaluation in the literatures are proposed by Sarkar et al. (2008), Koziolek et al. (2011), Durdik et al. (2012), Kocak et al. (2012), Venters et al. (2013), Penzenstadler et al. (2013) and Penzenstadler et al. (2014).

Earliest studies by Sarkar et al. (2008) have been proposed the software sustainability metric evaluation based on the scenario-based evaluation. The scenario-based evaluation is single interaction of many scenarios obtained from the nature that can be re-manufactured the software production towards sustainability (Beloff et al. 2004). This concept provides the eliciting, documenting as to evaluate the software development with related to the previous scenarios against the requirements. Although, Sarkar et al. (2008) have been used the Triple Bottom Line (TBL) approach in developing their metric evaluation but they only focusing on environment and economic dimension only without highlighting the social dimension individually.

Further studies by Koziolek et al. (2011) introduces the concept of metric-based evaluation to enhance the limitation studies by Sarkar et al. (2008) in which the researcher has classified the metric evaluation into indicators, indices, and framework through the TBL approach. Even though, the model uses TBL concept with defining the metric evaluation into three sustainability dimension, unfortunately the developer failed to show the integration concept between them. However, the proposed dimensions of evaluation mechanism in this model highlightes the environment and economic dimension, while the social dimension is keep soundless.

Durdik et al. (2012) investigates the software sustainability is necessary to be highlighted using system life cycle approach towards long living systems. The researcher creates a catalog of sustainability guidelines for the stakeholders such as project managers, software architects, and developers. The evaluation mechanism is an explicit consideration of sustainability during systems design, development, operation, and maintenance. The guidelines consists of selected methods, techniques and tools with reflected to sustainability including method descriptions, information of their industrial validation, supporting tools, potential benefit, connected risks, checklist and references. The researchers expresses the ideas pertaining to the strongly relationship between sustainability and quality model in the literatures. They has defined software sustainability development as the ability for cost efficient maintenance which is influenced by the quality attributes at the architectural level of a software system and the evolution is limited to an economical perspective. The researchers are claimed that the sustainability development related to the whole life-cycle of a software system and much contributed to environmental of the long living systems in the final product. However, the contribution does not performed the evaluation criteria to achieve sustainability.

Kocak et al. (2012) proposes green metrics to quantify the green performance of software systems. The developers defines four clusters of metrics based on the Green Performance Indicators (GPI) namely as IT resource usage metrics, Lifecycle metrics, Energy Impact metrics, and organizational metrics. Eventhough, the proposed characteristic and sub-characteristic by Kocak et al. (2012) are adopted from ISO/IEC 9126 and ISO/IEC 25010, unfortunately they are only addressed for environmental dimension only. For instance, they are assessing the greenness elements of an IT application and to indicate the energy consumption, energy efficiency, and energy saving possibilities. However, the assessment mechanisms are based on what need to be measured instead of who, when, where, why, and how to measure.

In Venters et al. (2013) embeddes the theory adopted from McCall's model in developing metric evaluation based on the merit of the represented entity through the weight given by the stakeholders. The value assigned by the stakeholders are used as the input into their proposed metric evaluation through the standard model recommended by Mc Call through software quality model. This model represents the assessment mechanism by defining the relationship between the proposed characteristic and sub-characteristic. Most of the proposed metrics evaluation are represented in frameworks and the weights are given by the

respected stakeholders. Unfortunately, the definition of goals between the characteristic and sub-characteristics in the proposed model are facing vulnerabilities in measuring the features towards sustainability. In fact, the proposed concept of metric evaluation as same as the previous works, in which the environment and economic dimension are most importantly than social dimension. Consequently, the integration of sustainability dimension does not exist.

The new concept of value viewpoint is highlighted by Penzenstadler et al. (2013) in software development towards software sustainability. The researchers introduces the values, indicators, regulations and activities to be practiced in order to achieve the level of sustainability. An indicator can be qualitative or quantitative metric that will be used to express a specific degree or score with regards as a value. For example: the indicator is risk of investment (ROI) will be used to assess the level of long term profit value and indirectly will be supported the economic sustainability. In the software sustainability, the indicator such as line of code (LOC) will be influenced to the value of (maintainability and efficiency) and indirectly will be supported the technical sustainability which is involved human behavioral (Penzenstadler et al. 2014).

The concept is appropriated through the activities which are measurement technique used to contribute to a specific value or a set of values (Penzenstadler et al., 2013). However, Penzenstadler et al. (2014) added the element of values and regulations into the concept of metric evaluation initiated by Koziolek (2011). The model proposed by Penzenstadler et al. (2013) and Penzenstadler et al. (2014) are the latest sustainability evaluation model that integrated the three pillars dimension of sustainability. Even though the suggested elements are performed in value-based perspective, unfortunately the goals of the values added are did not clearly defined to the other matters such as who, when, why, where and how to measure.

As explained from the previous studies, most of them are proposing the variety of concept association that have been used in the assessment mechanism. The most important element need to be highlighted in the assessment meachanism is the integration of environment, economic and social dimensions towards developing software sustainability. Unfortunately, most of them did not observed the sustainability paradigm as well. However, there only one model is observed the sustainability standard that proposed by Penzenstadler et al. (2013) and Penzenstadler et al. (2014). Despite that, the measurement process for all models in literatures only identify what to be measured and did not to attend who, when, why, and how to measure.

Consequently, the goals for software sustainability features are not clearly defined in the measurement process to achieve sustainability. Therefore, this study need to improve the limitations of previous works to defined the goals of characteristic and sub-characteritic of software development using the Goal Question Metric (GQM) approach. This approach is encouraging and motivating this research to enhance the software sustainability evaluation metric by utilizing the purposes, perspectives, point of views and the context of environment with fully focused on sustainability dimensions. The details will be discussed in the next sub-section.

B. Goal Oriented Measurement

Goal oriented measurement is a fundamental approach to monitor that all measurement activities be carried out in the context of a well-defined the measurement goal (Morasca, 2002). Basically, the measurement goal should be clearly connected between the proposed features and sub-features in the software development. Furthermore, the concept of software measurement in which the relationship between entity i.e. (software process, software product) and attributes i.e. (external attributes or internal attributes) that need to be measured must be specified entirely consisting of what, who, when, where, why, and how to measure (Pfleeger et al., 2001).

In order to identify the specified measurement goal, Goal Question Metric (GQM) is used in this study. GQM paradigm provides a framework for deriving measures that consists of goals, questions, and metric in a hierarchical as a guideline to the users. Goal is defined on a conceptual level as the main point that is compulsory to be attained. Goals can be derived by investigating the policy and the strategy of the organization that uses the GQM. The way to present goals are must be documented in a structured way and using templates for easier referencing.

The measurement goal can be defined by adapting the templates as proposed by Basili et al. (1994). The templates consisting of Purposes i.e. (to characterize, evaluate, predict, motivate and etc) that is pointed out to the object under study i.e. (process, product, model and etc) in order to clarify the object under study i.e. (to understand, assess, manage, engineer, learn, improve, and etc). The second element is Perspective that related to the specific issues or features that is need to be examined i.e. (cost, effectiveness, correctness, defects, changes, product metrics, reliability, and etc), from the point of views of the i.e. (user, developer, manager, customer, corporate

perspective and etc). Next, the third element is Environment focuses on the context of i.e. (process factore, people factors, problem factors, method, tool, constraint and etc). Table 1 illustrates the adapted templates to define goals in the specified measurement. The element of puposes and perspective are remained the same to the original templates, while element of environment is modified to the context of environment, economic, and social dimension of sustainability.

Table 1. Adapted Templates to Define Goals.

Element	Description
Purposes	To (characterize, evaluate, predict, motivate) the (process, product, model, metric) in order to (understand, assess, manage, engineer, learn, improve) it.
Perspective	Examine the (cost, effectiveness, correctness, defects, changes, product metrics, reliability, and etc). From the point of view of the (developer, manager, customer, corporate perspective and etc).
Environment	In the following context of (environment, economic, and social dimension).

III THE GOAL FOR SOFTWARE SUSTAINABILITY EVALUATION

This section presents an example of goals for software sustainability evaluation with focused on environment dimension using the adapted templates illustrated in Table 1 above. The characteristic and sub-characteristic for software sustainability evaluation is proposed in Ahmad et al. (2015) via Systematic Literature Review (SLR). The proposed characteristics and sub-characteristics are defined based on ISO/IEC 9126 (2002) – Product quality namely Efficiency, Functionality, Reliability, Usability, Maintainability, and Portability. In addition, there are new added characteristic in this research namely Integrity and User Conformity. The proposed characteristics and sub-characteristics are organized into the dimension of sustainability such as environment, economic, and social as illustrated in Table 2 below (Ahmad et al., 2015).

Table 2. The Proposed Characteristics and Sub-Characteristics of Software Sustainability Evaluation.

Dim.	Characteristic	Sub-Characteristic
Environment	Efficiency	Time Behaviour, Resource Utilization
	Functionality	Suitability, Accuracy, Interoperability, Security
	Portability	Adaptability, Installability, Co-Existence, Replaceability
	Reliability	Maturity, Fault Tolerance, Recoverability
	User Conformity	User Perception, User Requirement
	Integrity	Data Protection
Economic	Efficiency	Time Behaviour, Resource Utilization
	Maintainability	Analysability, Changeability, Stability, Testability
	Reliability	Maturity, Fault Tolerance, Recoverability
	Usability	Understandability, Learnability, Operability, Attractiveness
	Portability	Adaptability, Installability, Co-Existence, Replaceability
	User Conformity	User Perception, User Requirement
	Integrity	Data Protection
Social	Maintainability	Analysability, Changeability, Stability, Testability
	Functionality	Suitability, Accuracy, Interoperability, Security
	Portability	Adaptability, Installability, Co-Existence, Replaceability
	Usability	Understandability, Learnability, Operability, Attractiveness
	User Conformity	User Perception, User Requirement
	Integrity	Data Protection

Based on Table 2 above, the goal of one characteristic has been defined in Table 3 below as following the templates of goal definition by Basili et al. (1994).

Table 3. Example Defining Goal of Software Sustainability Evaluation for Environment Dimension.

Characteristic	**Efficiency**
Goal	*Purposes*: To predict the process impact and product impact in order to improve it. *Perspectives*: Examine the performance of energy impact of time behavior and the effectiveness of resource utilization from user, developer, maintainer point of view. *Environment*: In the following context of environmental impact.
Sub-Characteristic	**1. Time behaviour**
Sub- Goal	*Purposes*: To evaluate the response time behavior in order to improve it. *Perspectives*: Examine the time taken to complete a specified task from the user, developer, maintainer and SQA point of view. *Environment*: In the following context of concurrent tasks and system utilization of the real-time processing, user expectation of business needs or observation of user reaction towards environmental impacts.
Sub-Characteristic	**2. Resource Utilization**
Sub-Goal	*Purposes*: To evaluate the utilized resources behavior of computer system during testing or operating in order to improve it. *Perspectives*: Examine the input output resource utilization from user, developer, maintainers and SQA point of view. *Environment*: In the following context of quality of use of resources towards environmental impacts.

IV DISCUSSION

As shown in Table 3 above, the definition of goals for Efficiency characteristic are focused on environmental dimension with purposes for software process impact and software product impact. The investigation is centered to the perspectives of energy impact performance of time behavior and the effectiveness of resource utilization. The candidates involved in the measurement are gathered from point of views of user, developer, maintainer, and software quality assurance. The definition of sub-goals are more details in which the purposes, perspectives, and environment's context are fully described as to support the achievement of goal defined. However, the connection of the elements in templates should be related to the theory measurement highlighted by Pfleeger et al. (2001). Table 4 below illustrates the relationship between the Basili's templates and measurement theory by Pfleeger et al. (2001).

Table 4. Relationship Between Basili's Templates and Pfleeger's Measurement Theory.

Basili et al. (1994)	Pfleeger et al. (2001)
Purposes	What
Perspectives	What, who, where, why, and how
Environment	When, where

The adaptation of Basili's templates in this study can solve the limitation of the previous works that is the measurement process focused on what need to be measured instead of who, when, why, where and how to measure. By using the templates, the definition of goals can be specified into the *Purposes* are responsed to what, *Perspectives* are related to what, who, where, why and how, and *Environment* is answered when and where. Finally, the inclusive of goals definition process can monitor the measurement mechanism in developing the metric evaluation towards software sustainability development.

Dealing to the context of software engineering, the environment sustainability dimension is focused on the way of software is created, used, maintained and disposed with minimal impact on environmental (Amri et al., 2014). In addition, the environmental dimension is also referred to the green software in which the properties are influenced by two aspects such as energy consumption and resources consumption. The energy consumption is related to the efficiency of the systems in using the energy efficiency such as runtime efficiency, CPU intensity, memory usage, peripheral intensity, idleness and algorithmic efficiency (Amri et al., 2014). While, the resources consumption aspect related to the software products consists of software and hardware configuration, materials use i.e. print paper, storage media, ink toner and coverage can influence to the level of sustainability in environmental dimension (Penzenstadler et al., 2014).

The Bruntland Commission Report (1987) declares the sustainable development based on environment dimension is the development that preserves the diversity of biological species which is related to the essential ecosystems and ecological processes. The particular environmental sustainability is focused to the human well-being as to improve the human welfare by protecting the natural resources. The element consists of water, land, air, mineral and

ecosystems services. In addition, the elements will be contributed to the consumptions of sources of raw materials used for human needs that centered to the human wastes are under controlled (Gibson, 2006; Ciegis et al., 2009). Therefore, the presented criterias are necessary to be examined for evaluating the level of sustainability achievement in the software development.

V CONCLUSION

The application of GQM is recently used in business driven quality improvement approach very well in many domains. However, this approach currently beneficial to the researcher in developing evaluation metric for software and merely very helpful in defining the goals that need to be achieved towards software sustainability. GQM has much assists in defining the accurate goal and sub-goal for each characteristic and sub-characteristic in this study respectively with fully descriptions on the purposes, perspectives, the point of views, and the context of the environment that are needed to be highlighted. The future work is moved to develop the questions and metrics for each characteristic and sub-characteristic of the proposed list.

ACKNOWLEDGEMENT

The authors wish to thank the Ministry of Education, Malaysia for funding this study under the Long Term Research Grant Scheme (LRGS/b-u/2012/UUM/Teknologi Komunikasi dan Infomasi).

REFERENCES

Ahmad, R., Baharom, F., & Hussain, A. (2015). A Systematic Review on Characteristic and Sub-Characteristic for Software Development towards Software Sustainability. *World Scientific and Engineering Academy and Society (Wseas) 2015, 23 –25 April 2015, Kuala Lumpur, Malaysia* http://www.wseas.org/cms.action

Ahmad, R., Baharom, F., & Hussain, A. (2014). A Systematic Literature Review on Sustainability Studies in Software Engineering. *Knowledge Management International Conference (KMICe) 2014, 12–15 August 2014, Malaysia* http://www.kmice.cms.net.my/

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. *Encyclopedia of Software Engineering*, *2*, 528–532. doi:10.1.1.104.8626

Basili, V. R., Heidrich, J., & Lindvail, M. (2007). Bridging the Gap between Business Strategy and Software Development Why Measurement ? *International Conference on Information Systems (2007)*, 1–16.

Basili, V. R., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Münch, J., Trendowicz, A. (2010). Linking software development and business strategy through measurement. *IEEE Computer Society*, *43*(4), 57–65. doi:10.1109/MC.2010.108

Beloff, B., Lines, M., & Pojasek, R. . (2004). *Sustainable Development Performance Assessment Beth Beloff* (pp. 1–16).

Ciegis, R., Ramanauskiene, J., & Martinkus, B. (2009). The Concept of Sustainable Development and its Use for Sustainability Scenarios. *Engineering Economics*, (2), 28–37.

Calero, C., & Bertoa, M. F. (2013). Sustainability and Quality : icing on the cake. *Journal of Green Engineering*.

Calero, C., Moraga, M. A., & Bertoa, M. F. (2013). Towards a Software Product Sustainability Model. *Journal of Sustainability*, *25010*, 4. Retrieved from http://arxiv.org/abs/1309.1640

Durdik, Z., Klatt, B., Koziolek, H., Krogmann, K., Stammel, J., & Weiss, R. (2012). Sustainability guidelines for long-living software systems. *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, 517–526. doi:10.1109/ICSM.2012.640531

Gibson, R. B. (2006). Beyond the three pillars : Sustainability Assessment as a Framework for Effective Integration of Social, Economic, and Ecological Considerations in Significant Decision Making. *Journal of Environmental Assessment Policy and Management*, *8*(3), 259–280.

International Organization for Standardization. (2002). Software engineering- Product quality-Part 2: External Metrics (pp. 1–97). Retrieved from http://www.iso.org/iso/sustainable_events_iso_2012.pdf

Kocak, S. A. (2012). Green Software Development and Design for Environmental Sustainability. *Journal of Green Engineering*.

Koziolek, H. (2011). Sustainability Evaluation of Software Architectures : A Systematic Review. *Journal of Environmental Assessment Policy and Management.*

Koziolek, H., Domis, D., Goldschmidt, T., & Vorst, P. (2013). Measuring Architecture Sustainability. *IEEE Software*, *30*(6), 54–62. doi:10.1109/MS.2013.101

Penzenstadler, B., Raturi, A., & Richardson, D. (2014). Safety, Security, Now Sustainability: The Nonfunctional Requirement for the 21st Century. *IEEE Software*, *31*, 40–47. doi:10.1109/MS.2014.22

Penzenstadler, B., & Femmer, H. (2013). A generic model for sustainability with process- and product-specific instances. *Proceedings of the 2013 workshop on Green in/by software engineering - GIBSE '13*, 3.

Pfleeger, S. L. (2001). *Software Engineering: Theory and Practice", 2nd ed*. Upper Saddle River, N.J: Prentice Hall.

Sarkar, S., Kak, A. C., & Rama, G. M. (2008). Metrics for measuring the quality of modularization of large-scale object-oriented software. *IEEE Transactions on Software Engineering*, *34*(5), 700–720. doi:10.1109/TSE.2008.43

United Nations World Commission on Environment and Development (1987). Report of the World Commission on Environment and Development: Our Common Future In United Nations Conference on Environment and Development, 1987.

Venters, C. C., Lau, L., Griffiths, M. K., Holmes, V., Ward, R. R., & Xu, J. (2013). The Blind Men and the Elephant: Towards a Software Sustainability Architectural Evaluation Framework. *Journal of Green Engineering*.