Evaluation of Software Product Quality Attributes and Environmental Attributes using ANP Decision Framework

Sedef Akınlı Koçak
Data Science Laboratory
Environmental Applied Science
Ryerson University
Toronto, Canada
Sedef.akinlikocak@ryerson.ca

Gülfem Işıklar Alptekin
Computer Engineering
Galatasaray University
Istanbul, Turkey
gisiklar@gsu.edu.tr

Ayşe Başar Bener
Data Science Laboratory
Mechanical and Industrial
Engineering, Ryerson University
Toronto, Canada
Ayse.bener@ryerson.ca

Abstract—**Green software development is a relatively new research area within green IT. Software development industry has started getting pressure from regulators to consider green software development. As a result, green attributes of software products are gaining importance as quality attributes. In this study, we evaluate environmental sustainability and software quality criteria using a well-known multi criteria decision making approach: Analytical Network Process (ANP). The aim is to determine the relationships among quality and environmental attributes and relative priorities of attributes. The results are presented as a guide for green software developers. The priority weights of attributes may be used to analyze trade-off between conflicting product quality and environmental requirements.**

Index Terms— **Green Software, Software Quality, Environmental Sustainability, Analytic Network Process**

I. Introduction

In software development, comprehensive specification and evaluation of software product quality is a key factor in ensuring desired level of quality. A desired level of quality for software may be achieved by defining appropriate quality characteristics, taking into account the purpose of usage of the software product.

Similarly, green information and communication technologies (ICT) and green software have become important research areas that aim to achieve environmentally sustainable computing systems. Designing green and sustainable software that also satisfies customer needs in new software products and/ or legacy software are complex tasks. Developing green software necessitates identification of the development requirements as well as quality metrics. There have been various software quality models in the literature. These models propose sets of metrics to evaluate both product and process quality [1, 2, 3, 4].

Quality requirements (aka quality attributes) and customer requirements (aka functional requirements) sometimes may conflict. In these cases, practitioners make trade-off decisions or prioritization to solve the conflict. However, in most cases functional requirements and quality attributes depend on each other, thereby complicating the trade-off decisions [5]. In addition, sustainability concerns induce environmental requirements (green requirements) to the system. Those green requirements generally create conflicts on both functional requirements and quality attributes. Interdependencies between environmental concerns and quality attributes are typically hidden compared to the dependencies between functional requirements and quality attributes. Hence, in order to develop environmentally sustainable software products, we need to address these conflicts and prioritize concerns and requirements. A multi-criteria decision making (MCDM) approach is suitable for this aim since it incorporates multiple objectives for prioritizing conflicts [6]. MCDM approaches are known for their flexibility in handling complex information, and their ability to construct values during priority-setting in trade-off analysis.

In our earlier work [7], we built a framework using ANP which is a widely used MCDM approach to prioritize quality and environmental criteria in a general sense. The work in this paper extends our previous framework by defining sub-criteria of quality and environmental criteria. The main contribution of this work is that it expands the framework that enables decision makers determining the importance of each requirement when deciding on the technical specifications of the software. A trade-off analysis may also be applied during requirement analysis step where making decisions regarding sustainability affects both software product development and software system itself.

Another key consideration for delivering business value with environmental responsiveness is deciding what requirements to develop and how to develop. At this point, trade-offs among different requirements and stakeholders emerge. Trade-offs are inevitable and need to be balance wisely. Our research seeks to analyze the various trade-off possibilities.

This paper is organized as follows: Section II summarizes the concepts of software quality models and evaluations, requirement prioritization, software quality and environmental attributes. This section also presents the related literature. In Section III, the methodology used in the evaluation framework is briefly explained. The analysis, the results and the discussion with trade-off analysis are presented in Section IV. Section V gives conclusions and future work.

II. RELATED WORK AND BACKGROUND

Researchers have been studying sustainability of software from different perspectives: hardware energy efficiency in terms of power consumption; optimization of algorithms and / or software architecture; effective and efficient usage of software engineering practices and tools as defined in ISO/IEC system and software engineering standard series [8, 9, 10].

A. *Software Quality Models*

Software quality is defined as a *"set of features and characteristics of product or service that bears on its ability to satisfy the stated or implied needs"* [8]. Software development projects need to be completed within certain constraints such as time, budget, and resources. Therefore, managers need to make trade-off decisions to determine how much functionality/ or which features will be included in each release. In software engineering literature, various software quality models have been proposed. Some of the most standard and well-known quality models are McCall's model [1], Boehm's model [2], Bowen's model [3], Dromey's model [4], ISO/EC 9126-1 [8], and ISO/EC 25010 [10] models. The considered quality criteria in each model is summarized in Table 1. The common criteria found in the majority of these models are: efficiency, reliability, usability, functionality, maintainability, and portability.

TABLE I. QUALITY MODELS IN RELATED LITERATURE

Criteria/Goals	McCall [1]	Boehm [2]	Bowen [3]	Dromey [4]	ISO-9126 [8]	ISO-25010 [10]
Maintainability	+		+	+	+	+
Flexibility	+		+			
Testability	+	+				
Correctness	+		+			
Efficiency	+	+	+	+	+	
Reliability	+	+	+	+	+	+
Integrity	+		+			
Usability	+		+	+	+	+
Portability	+	+	+	+	+	+
Reusability	+		+	+		
Interoperability	+		+			
Understandability		+				
Functionality				+	+	+
Modifiability		+				
Performance						+
Supportability						
Security						+
Compatibility						+

B. *Decision Making in Software Requirements*

Software development is a design intensive process. Every person involved in this process has to make a number of decisions, each of them with several possible choices. Most of the time, decisions are made according to personal knowledge and experience. But, as software development projects get complex and diverse, decision making becomes a group activity where people need to use decision tools and techniques to communicate and coordinate their decisions.

Requirement prioritization is a process of managing the relative importance and determination of different requirements within the limited resources. It is a challenging activity to find the right balance among competing quality requirements. In this paper, ANP, a well-known MCDM approach, is used to prioritize the criteria. It improves the quality of decisions by providing information on trade-offs, and it increases confidence in decisions and provide insights into the criteria and the alternatives. In decision support systems studies, Analytical Hierarchy Process (AHP), hybrid ANP/AHP method, fuzzy AHP [11], AHP-goal model, binary search tree creation, greedy-type algorithms [12] are the most commonly used methods. Most of the above mentioned tools and methods recognize the AHP as a reference method. However, AHP is difficult to use with large sets of requirements. In order to solve this problem, a case-based ranking method [13] is introduced for decision support system based on a machine learning approach. Another technique is called quality function deployment (QFD) which helps the organization from the early stages of production process (i.e. designing phase) to reach the customer's satisfaction with proper technical requirements. QFD has been adopted as a software quality function deployment (SQFD) [14] and utilized to plan and design new software or improved software. Although the number of research on applying SQFD to software industry is limited compared to manufacturing industry, the use of QFD in software engineering has gained more attention in recent years.

C. *Software Quality Attributes and Sustainability*

Various systems (e.g. energy systems, management systems, and computer systems) take sustainability as an objective for reaching a high quality level. Penzenstadler, Mahaux, and Salinesi [15] stated that current discussions on the sustainability requirements foster on how to define, measure and assess sustainability as quality attribute of software. A first quality model for green and sustainable software was developed by Kern, Dick, Naumann, Guldner, and Johann [16]. It refers to quality factors from ISO /IEC 25000, and it is based on the directly and indirectly related criteria of software. It gives an overview of potential aspects which may be taken as sustainability criteria and metrics for software products. Calero and Bertoa [17] considered sustainability as a new factor that affects software product and process quality. In their work, a new quality model (ISO 2510+S) is presented based on ISO/25010. The authors differentiated the quality factors in respect to the sustainability impact and described related and unrelated sub-characteristics. All of these studies discuss the relation of the software quality with the sustainability in a general manner. They point out that the product, as well as the quality in use, needs to be considered when evaluating the sustainability of the software. So far, to the best of our knowledge, there is no empirical study that uses the quality attributes with sustainability attributes.

D. *Environmental Criteria and Green Metrics*

With the purpose of clarifying different ways that ICT can contribute to the reduction of CO_2 emissions and sustainable development, three levels of effects of ICT have been identified [18, 19]: direct, indirect and systematic. In terms of identifying environmental criteria, in this paper, we are interested in direct effects. Direct effects refer to those caused by software product, i.e. the resource consumption when using the applications. Environmental performance of the software systems can be evaluated with environmental criteria and green metrics. They are also necessary for practitioners to develop, deploy and control the system sustainability. Here, we address this point by analyzing green metrics as a criteria set. Those criteria may be used in setting up environmental objectives and evaluating the sustainability of software products. In order to refine such a group of criteria, we reviewed recent green metrics studies. Kansal and Zhao [20] presented an energy profiling metric for applications at design phase. The energy consumed by an application is divided into two parts: the performance required by the application and the system resources used. Their application energy profiling approach considers how energy consumption is spanned across the involved resources such as CPU, disk, memory and network interfaces during different states. Lago and Jansen [21] focused on process, service and people aspects and proposed a framework which makes environmental strategies and green indicators available as-a-service. Their adaptation strategies are based on one single metric for each. The metrics are not analyzed together and their dependencies are not considered. However, in our proposed approach, we analyze priorities based on the dependencies of metrics using ANP approach.

In a recent systematic literature review, Bozzelli, Gu, and Lago [22] identified that most of the measures were defined to measure energy consumption, either directly or as a performance indicator. Following that study, Agarwal, Nath, and Chowdhur [23] indicated that software performance depends on the basic computing resources like the CPU and the memory. Software is required for faster CPUs and increased memory. In this situation, sustainability is affected from the increasing resource consumption. Therefore, we concluded that we need to evaluate software products sustainability in terms of resource usage.

According to Kipp, Jiang, and Fugini [24], green metrics are the key drivers to identify the "greenness" of an IT application and to indicate the energy consumption, energy efficiency, and energy saving possibilities. They classify four clusters of metrics: a) IT resource usage that is related to energy consumption, b) lifecycle metrics that monitor energy consumption and metrics that develop energy aware indicators, c) energy impact metrics that are related to the lifecycle impact on the environment (i.e. the electricity, the power supply, the consumed material, and the CO_2 emissions), and d) organizational metrics that consider the assessment of additional costs due to energy-related initiatives. The following table (Table II) summarizes the main categories of metrics used for each cluster [24].

Using the same classification, Ferreira, Pernici, and Plebani [25] recently have presented two different types of indicators for different system layers: Key Performance Indicators (KPIs) and Green Performance Indicators (GPIs). KPIs measure system attributes that are based on quality models, while GPIs measure the system greenness with respect to energy consumption. Their approach focuses on GPIs, in particular, usage and energy related indicators. Although most of them may represent different system layers, their proposed framework turns indicators through normalization functions such that they can be compared and aggregated within meaningful clusters.

TABLE II. GREEN METRICS CLASSIFICATION

IT resource usage	Lifecycle	Energy Impact	Organizational
- CPU usage - Memory usage - Storage usage - I/O usage	- Lifecycle cost - Process engineering - Quality of service	- Consumables - System power/energy usage - Greenhouse gas (e.g. CO_2 emissions)	- Laws and regulations - Resource efforts - Greenhouse gas credits - Green solutions

In our study, we focus on the criteria that are directly related to the quality and environmental attributes. Hence, we aggregate two sub-criteria sets for our study using Kipp, Jiang, and Fugini [24] and Ferreira, Pernici, and Plebani [25] classifications: resource consumption that have the higher influence over energy consumption and energy impact which is closely related to the resource usage. The cluster criteria, sub-criteria and their indicators are listed in Section IV.B.

III. ANALYTIC NETWORK PROCESS (ANP)

The ANP generalizes a widely used multi-criteria decision making tool, the AHP, by replacing hierarchies with networks. The AHP is a well-known technique that decomposes a problem into several levels in such a way that they form a hierarchy [6]. ANP uses a network structure with interrelationships among the decision levels and attributes [26]. A relative ratio scale of each element measurement is derived from pairwise comparisons of the elements in a level of the hierarchy with respect to an element of the preceding level. ANP provides a general approach to deal with decisions considering the dependence of higher level elements to lower level elements and the dependence of the elements within a level. In general the ANP consists of three stages:

1. Defining the goal, criteria, sub-criteria,
2. Defining the interdependencies and the network, and
3. Calculation of the priorities of the elements [26].

The relative importance value according to pairwise comparisons between criteria and sub-criteria are performed using a scale of 1–9, where a score of 1 indicates equal importance between the two elements and 9 represents the extreme importance of one element compared to the other one. The relations are calculated as in Equation.1

$$a_{ji} = 1/a_{ij} \tag{1}$$

where a_{ij} denotes the importance of the i^{th} element compared to the j^{th} element, and a_{ii}=1 are preserved in the pairwise comparison matrix to improve the consistency of the judgments. The ANP requires that the consistencies of pairwise comparison matrices are checked to prove the consistency of comparisons. A zero value would indicate perfect consistency whereas larger values indicating increasing levels of inconsistency. Saaty [26] has indicated that the consistency should be about 10% or less to be acceptable. If it exceeds the 10% level, value judgment may need to be revised.

After calculating the importance a_{ij}, the composite weights are extracted considering the interrelationships between criteria and sub-criteria [27]. We utilized ANP approach to define priority weights of the quality and environmental criteria and their sub-criteria. This approach best suits in our aim since it identifies the interdependencies between criteria and their sub-criteria in the network structure.

IV. EVALUATION FRAMEWORK

A. *Integrating Environmental Requirements in Decision Making*

Users as well as other stakeholders, such as project managers, regulators, other decision makers, recognize the limited resources of the system and introduce green requirements to the development process. In addition to quality requirements, environmental concerns possess multiple viewpoints and conflicts. They must be identified and eventually resolved. The value here is not only identifying the conflicts but also eliciting the opportunities resulting from the trade-offs.

Resource concerns, efficiency and time constraints have direct effects on software development life cycle. Integrating green requirements to the development process creates diversity in understandings, source of knowledge, and opportunities based on interdependencies rather than conflicts between requirements. Resource usage, for example, directly leads to changes in efficiency and reliability, which affects the quality. While resource usage affects functionality of the product, it also brings the opportunity to save energy [28, 29].

As a result, presence of green requirements in the process of developing the decision framework provides an opportunity to explore and construct different development outcomes regarding different goals.

The objective of our proposed framework is to determine the priority weights of software quality and environmental characteristics. Those weights will help to identify technical requirements and their correlations that will be useful in design and development processes of green software. Calculation of the metrics used in the framework are not detailed in this paper since the main focus is on the relative importance and priorities of the criteria on their cluster. The decision process addresses the problem of selecting the most important criteria that a software developer needs to focus in the design process. The process can be divided into three main steps:

1. Identifying the criteria and sub-criteria and their interdependencies.
2. Making pairwise comparisons using questionnaire.
3. Obtaining the interdependent priorities of the criteria and sub-criteria.

Each step is presented in more detail in the following sub-sections.

B. *Determination of Criteria and Sub-Criteria*

Quality criteria (functionality, usability, efficiency, reliability) and their related sub-criteria were adopted from common software product criteria in the literature. Environmental criteria were adopted from IT resource usage and energy impact categories (see sub-section II.D) [24]. Table III describes the criteria and the selected sub-criteria. Figure 1 depicts our decision framework. Here metrics and indicators are used interchangeably.

For environmental dimension of sustainability, the classification is important since all the interdependencies related to both quality and environment are elicited for the trade-off analysis. Similar classification may also be used to identify other aspects of sustainability dimensions that correspond to quality requirements.

TABLE III. SOFTWARE QUALITY AND ENVIRONMENTAL CRITERIA

Software Quality Criteria (*Q*) [8]		
Q_1	Functionality	A set of attributes that bear on the existence of a set of functions and their specified properties; the functions are those that satisfy stated or implied needs.
Q_2	Usability	A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.
Q_3	Reliability	A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
Q_4	Efficiency	A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.
Environmental Sustainability Criteria (*E*) [24, 25]		
E_1	Resource usage	The energy consumption by a software relates both to the nature of the application and to the system configuration of the run time environment
E_2	Energy impact	Metrics relate to the impact of the application on the environment, considering electricity, power supply, consumed material, and emissions

C. *Questionnaire and Pairwise Comparisons*

Our proposed approach has two major criteria in the first level, six sub-criteria in the second level and twenty one sub-criteria in the third level. In our previous study [7], we calculated the priority weights of two criteria (quality and environment) and related sub-criteria *(Q_i, E_i)*.

We prepared a questionnaire for experts in order to find out the weights for criteria and interactions among all the second and third level indicators. Hence, we extended the framework by integrating third level indicators. The questionnaire was filled out by the same expert group as in our first study. Experts are from four different software development companies. They are software experts with an experience of 4 to 20 years in the field. Khorramshahgol and Moustakis [30] pointed out that best suited number of experts as an interviewee is in general 5–15 persons.

In order to avoid any vagueness or hard readability that may affect the answers given by the experts, and facilitate the understanding about criteria and the ANP model, we conducted personal interviews. At the beginning of the questionnaire, the meaning of the criteria and comparison of criteria were provided to advance understanding of the contents in the questionnaire by the experts. The geometric mean of 10 questionnaires is used to form the output of the pairwise comparisons [27].

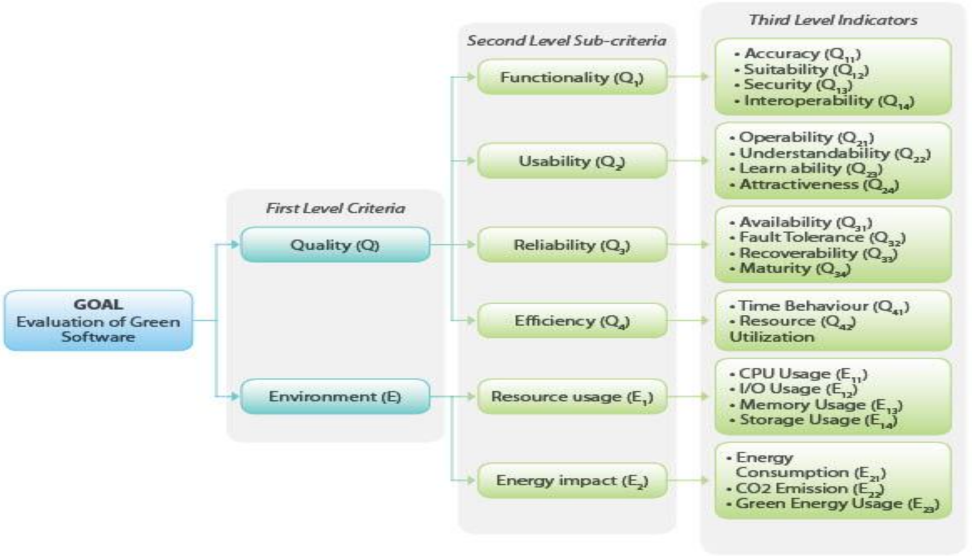

Fig. 1. Research Framework

D. Analysis of the interdependencies between the criteria and sub-criteria

In this step of the methodology, we applied the ANP approach. We used the Super Decisions[1] software tool in order to calculate the results of final priorities of the sub-criteria (Figure 1). The Super Decisions is a simple and easy-to-use software for constructing decision models with dependence and feedback. The consistency indexes of all these matrices were found to be smaller than 0.10, which is in the acceptable range. Small consistency indicates the consistency of experts' evaluations. The pairwise comparisons enabled us to retrieve relative weights of criteria and sub-criteria and indicators.

E. Results and Discussion

Each expert has to decide which one of the two given criteria is more important in developing a green and sustainable software product, along with its relative importance degree. They defined the degree of importance using the 1-9 scale. (For instance, 9 means that the first criterion is extremely important compared to the second one). In order to determine aggregate individual judgments, we took the geometric average of the responses. Table IV through Table XI represent the geometric means of pairwise comparisons between environmental sustainability criteria and quality criteria.

TABLE IV. PAIRWISE COMPARISON OF ENVIRONMENTAL (E) SUB-CRITERIA

	E_1	E_2
E_1	1	2
E_2	1/2	1

TABLE V. PAIRWISE COMPARISON OF QUALITY (Q) SUB-CRITERIA

	Q_1	Q_2	Q_3	Q_4
Q_1	1	2	3	l/3
Q_2	1/2	1	1/2	l/3
Q_3	1/3	2	1	l/4
Q_4	3	3	4	1

TABLE VI. PAIRWISE COMPARISON OF FUNCTIONALITY (Q_1) INDICATORS UNDER THE QUALITY

	Q_{11}	Q_{12}	Q_{13}	Q_{14}
Q_{11}	1	4	2	2
Q_{12}	1/4	1	1/2	1/2
Q_{13}	1/2	2	1	1/2
Q_{14}	1/2	2	2	1

TABLE VII. PAIRWISE COMPARISON OF USABILITY (Q_2) INDICATORS UNDER THE QUALITY

	Q_{21}	Q_{22}	Q_{23}	Q_{24}
Q_{21}	1	2	1	3
Q_{22}	1/2	1	2	2
Q_{23}	1	1/2	1	3
Q_{24}	1/3	1/2	1/3	1

[1] Super Decisions (ver. 2.2.6) [Open source Software]. Available from: http://www.superdecisions.com

TABLE VIII. PAIRWISE COMPARISON OF RELIABILITY (Q_3) INDICATORS UNDER THE QUALITY

	Q_{31}	Q_{32}	Q_{33}	Q_{34}
Q_{31}	1	1	2	3
Q_{32}	1	1	1	1
Q_{33}	1/2	1	1	2
Q_{34}	1/3	1	1/2	1

TABLE IX. PAIRWISE COMPARISON OF THE EFFICIENCY (Q_4) INDICATORS UNDER THE QUALITY

	Q_{41}	Q_{42}
Q_{41}	1	3
Q_{42}	1/3	1

TABLE X. PAIRWISE COMPARISON OF THE RESOURCE USAGE (E_1) INDICATORS UNDER THE ENVIRONMENT

	E_{11}	E_{12}	E_{13}	E_{14}
E_{11}	1	2	2	4
E_{12}	1/2	1	2	3
E_{13}	1/2	1/2	1	3
E_{14}	1/4	1/3	1/3	1

TABLE XI. PAIRWISE COMPARISON OF THE ENERGY IMPACT (E_2) INDICATORS UNDER THE ENVIRONMENT

	E_{21}	E_{22}	E_{23}
E_{21}	1	2	1
E_{22}	1/2	1	2
E_{23}	1	1/2	1

These comparison values were used as input to the ANP methodology, and we calculated the priorities which are summarized in Table XII. As the last step of the proposed procedure, the final priorities of the each criteria, sub-criteria and indicators were calculated.

TABLE XII. SUB-CITERIA AND INDICATOR LEVEL WEIGHTS

Quality Criteria	**Weight**	**Indicators**	**Relative Weight**
Reliability	0.43	Availability	0.37
		Fault Tolerance	0.24
		Recoverability	0.23
		Maturity	0.15
Functionality	0.13	Accuracy	0.44
		Suitability	0.22
		Security	0.22
		Interoperability	0.11
Usability	0.14	Operability	0.36
		Understandability	0.28
		Learnability	0.26
		Attractiveness	0.11
Efficiency	0.28	Time Behaviour	0.75
		Resource Utilization	0.25
Environmental Criteria	**Weight**	**Indicators**	**Relative Weight**
Resource Usage	0.66	CPU Usage	0.43
		I/O Usage	0.24
		Memory Usage	0.24
		Storage Usage	0.09
Energy Impact	0.33	Energy Consumption	0.41
		CO$_2$ Emission	0.33
		Green Energy Usage	0.26

The results indicate that, among the two environmental criteria, the resource usage criterion has the highest priority with 66%. This shows that it has a high impact on green software development. Among the four quality related sub-criteria, the reliability has the highest priority with 43%, followed by the efficiency and the functionality with 24%, and 23%, respectively.

The reliability of a system is a measure of its ability to provide a failure-free operation. This also means the actual usage time of the product by user. Therefore, the reliability is correlated with the behavior of software with respect to time, and it is one of the sub-criteria of the efficiency. The efficiency is the second important quality criteria, with 28% of weight, followed by usability and functionality. Software efficiency considers the amount of information technology resources that are used efficiently in terms of energy. Undoubtedly, the priority weights may depend on the software type.

The results related to environmental criteria show that CPU usage has the highest priority with 43% followed by I/O usage and memory usage with 24% and storage usage with 9%. This demonstrates that the most important energy related resource is the CPU. The resource usage has a direct influence on the energy consumption sub-criteria, which has the higher relative weight with 41% in the energy impact criteria followed by CO$_2$ emission with 33%.

Figure 2 shows the comparison of results of our findings in this paper with our previous study [7]. The results reveal that the priority ranking of the criteria is different. In our previous work, the efficiency had the highest priority (34%) followed by reliability (24%), functionality (23%) and usability (19%). However, in this work, the reliability of the product has the highest weight (43%) followed by efficiency (28%), usability (14%) and functionality (13%). The reason is that the integration of more detailed sub-criteria and indicators enables DMs to better analyze interdependencies. Therefore, the final priority ranking may change if we ignore the interactions of detailed sub–criteria and indicators.

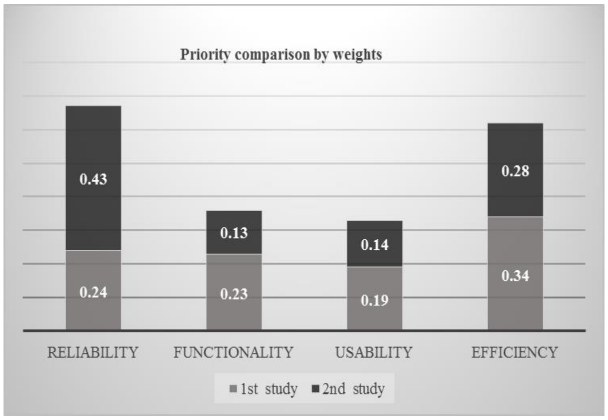

Fig. 2. Comparison of priority weight results with our previous work [7]

F. *Assessing the Criteria and Trade-off Analysis*

The sub-criteria describe the entire system with a given goal and are, therefore, usable and widely understood indicators of quality and green attributes of software. Criteria with higher influence to the system's goal get higher weights. For instance, considering the energy consumption, criteria related to the CPU power consumption have higher importance than I/O. Hence, we can conclude that CPU related events represent higher risk for both quality and environment.

The set of quality and environmental criteria are highly related to the time behaviour that has high impact on resource usage, energy consumption and efficiency. The results indicate two major trade-offs: First one is between reliability, which is threatened by performance degradation, system behavior and functionality loss, and limited resource usage. The second one is between energy impact which is threatened by energy consumption and related CO_2 emissions, and efficiency of the product which is threatened by time behaviour. Although resource usage has high priority, resource utilization does not show the same behaviour as quality criteria.

The analysis reveals that future action would be converting reliability, efficiency and resource usage trade-offs into the opportunities for saving energy.

G. *Threats to Validity*

The overall objective of the study is to evaluate the priorities of the criteria to achieve green software. We introduced a prioritization method to gain a better understanding of interdependencies of the quality and environmental criteria in green software development. Selection of the criteria may introduce bias. We chose the quality criteria from well-known and mostly used quality models. We selected the environmental criteria from the IT related environmental studies. Environmental applications on IT systems are fairly new research area. Therefore, the environmental criteria might be different according to the application area. The number of questionnaires filled by the experts may be another threat to validity. Since a continuous nature of the study, we needed to interview the same DMs in order to be consistent and be able to make comparisons.

V. Conclusion and Future Work

Most software development companies need to adapt their development processes to meet new software business standards. One of these standards is the development of environmentally friendly products. In many cases, these objectives are achieved at a cost of reducing or increasing the consumption of various resources within the organization. As a result, the software development industry needs making certain trade-offs between the demands of end-users and the requirements for environmental sustainability. This work is an initial attempt to prioritize quality and environmental attributes, and further to build trade-off models. We made use of ANP, a well-known MCDM technique, for dealing with conflicting criteria.

The proposed ANP framework is subject to the judgment of experts for the comparison among the criteria, sub-criteria and indicators. In the process of creating the ANP decision framework, the discussion with experts was made before the relationships among levels and criteria are given. Therefore, the ANP structure in this research is reliable and valid.

Using our approach and framework, decision makers may determine requirements' priorities and use the framework as the basis to see the potential outcomes and impacts of the criteria. Additionally, this approach offers an opportunity to properly shape software development projects from the beginning to the maintenance phase in a strategic way.

Next steps will be constructing a QFD model for new green software product development, by using the ANP weights obtained from this study. We plan to integrate a goal programming technique to QFD in order to consider resource and budget limitations.

References

[1] J. A. McCall, P. K. Richards, and G. F. Walters. Factors in software quality, General Electric, National Technical Information Service, 1977.

[2] B. W. Boehm, J. R. Brownand, and J. R. Kaspar, Characteristics of Software Quality, TRW Series of Software Technology, Amsterdam, North Holland, 1978.

[3] T .P. Bowen, G. B. Wigle, and J. T. Tsai. "Specification of software quality attributes," Technical Report RADC-TR-85-37, Rome Air Development Center, 1985.

[4] R. G. Dromey, "A model for software product quality," Software Engineering, IEEE Transaction, vol.21, no.2, 1995, pp. 146-162.

[5] E. E. Karsak, S. Sozer and S. E. Alptekin "Product planning in quality function deployment using a combined analytic network process and goal programming approach,", Computers & Industrial Engineering, vol. 44, 2002, pp. 171–190.

[6] T. L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, New York, NY, 1980.

[7] S. Akınlı Koçak, G. Gonzales Calienes, G. Işıklar Alptekin, and A. Başar Bener, "Requirements prioritization framework for developing green and sustainable software using ANP - based decision making," In Proceedings of the Environmental Informatics Conference, 2013 (Hamburg, Germany, September 2-4, 2013).

[8] ISO/IEC 9126-1. Information technology - Software quality characteristics and metrics - Part 1: Quality characteristics and sub-characteristics, ISO, Int. Electrotechnical Commission, Geneva, 2001.

[9] ISO/IEC IS 12207. Systems and software engineering- Software life cycle processes, International Organization for Standardization, Geneva, 2008.

[10] ISO/IEC, IS 25010. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, Geneva, 2011.

[11] Z Sener and E. E. Karsak. "A fuzzy regression and optimization approach for setting target levels in software quality function deployment," Software Quality Journal, 2010, vol.18, pp.323–339.

[12] G., Ruhe, A., Eberlein, and D. Pfahl, "Quantitative Win-win: A New Method for Decision Support in Requirements Negotiation," In proceeding of the 14th Int'l Conf. Software Eng. and Knowledge Eng., (Ischia, Italy), 2002, pp.159-166.

[13] P. Avesani, C. Bazzanella, A. Perini, and A. Susi. "Supporting the Requirements Prioritization Process. A Machine Learning Approach," In proceeding of 16th Int'l Conf. Software Eng. and Knowledge Eng. (Maurer F. and Ruhe, G., Eds, Banff, Canada), 2004, pp.306-311.

[14] S. Haag, M. Raja, and L. L. Schkade. "Quality function deployment usage in software development," Communications of the ACM, 1996, vol.39, no.1, pp.41-49.

[15] B. Penzenstadler, M. Mahaux, and C. Salinesi. "Requirements engineering for sustainable systems, In proceeding of the International Working Conference on Requirements Engineering: Foundation for Software Quality. (Essen, Germany), 2012, vol.52, pp.8-13.

[16] E. Kern, M. Dick, S, Naumann, A, Guldner, and T. Johann, "Green Software and Green Software Engineering – Definitions, Measurements, and Quality Aspects," In proceeding of the First International Conference on ICR for Sustainability (Zurich, Switzerland, 14-16 February 2013), 2013, pp.87-94.

[17] C. Calero and M. F. Bertoa. "25010+S: A software quality model with sustainable characteristics. Sustainability as an element of software quality," In Proceeding of the GIBSE 2013, (Fukuoka, Japan), 2013.

[18] Organization for Economic Co-operation and Development (OECD), 2010, OECD Greener and Smarter: ICTs, the environment and climate change 2010. OECD Pub. Paris, France.

[19] L. M., Hilty, P. Arnfalk, L. Erdmann, J. Goodman, M. Lehmann, and P. A. Wäger, "The relevance of information and communication technologies for environmental sustainability–A prospective simulation study," Environmental Modelling & Software, 2006, vol.21, no.1, pp.1618-1629.

[20] A. Kansal, F. Zhao, "Fine-grained energy profiling for power-aware application design," SIGMETRICS Performance Evaluation Review, 2008, vol.36, no.2, pp.26–31.

[21] P. Lago, T. Jansen, "Creating Environmental Awareness in Service Oriented Software Engineering," In: Maximilien, E.M., Rossi, G., Yuan, S.-T., Ludwig, H., Fantinato, M. (eds.) ICSOC 2010. LNCS, Springer, Heidelberg, 2011, vol. 6568, pp. 181–186.

[22] P. Bozzelli, Q. Gu, and P. Lago. "A systematic literature review on green software metrics," Technical Report: VU University Amsterdam, 2013, Available at: http://www.sis.uta.fi/~pt/TIEA5_Thesis_Course/Session_10_2013_02_18/SLR_GreenMetrics.pdf

[23] S. Agarwal, A. Nath, and D. Chowdhury, "Sustainable Approaches and Good Practices in Green Software Engineering," International Journal of Research and Reviews in Computer Science (IJRRCS), 2012, vol. 3, no.1. pp. 1425-1428.

[24] A.Kipp, T. Jiang, and M. Fugini, "Green metrics for energy-aware IT systems," IEEE Computer Society, 2011, vol.42, pp.241-248.

[25] A. M. Ferreira, B. Pernici, and P. Plebani. "Green performance indicators aggregation through composed weighting system," ICT as Key Technology against Global Warming. Springer Berlin Heidelberg, 2012, pp.79-93.

[26] T. L. Saaty, Decision making with dependence and feedback: The analytic network process. RWS Publications, 1996, Pittsburgh, PA.

[27] T. L. Saaty and L. G. Vargas "Diagnosis with dependent symptoms: Bayes theorem and the analytic hierarchy process" Operations Research, 1998, vol.46, no.4, pp.491–502.

[28] S. Akınlı Koçak, A. Miransky, G. Işıklar Alptekin, A. Başar Bener, E. Cialini, "The impact of improving software functionality on environmental sustainability," In: Proceedings of the First International Conference on ICT for Sustainability (ICT4S, Zurich, Switzerland), 2013, pp.104-110.

[29] A. Miransky, S. Akınlı Koçak, E. Cialini, and A. Başar Bener, "A. Save energy with the DB2 10.1 for Linux, UNIX, and Windows data compression feature," IBM DeveloperWoks, Technical Library, 2013, Available at: http://www.ibm.com/developerworks/data/library/techarticle/dm-1302db2compression/

[30] Khorramshahgol, R., and Moustakis, V. S. "Delphic hierarchy process (DHP): A methodology for priority setting derived from the Delphi method and analytical hierarchy process," European Journal of Operational Research, 1988, vol.37, no.3, pp.347-354.