Green Software Development and Design for Environmental Sustainability

Sedef AKINLI KOCAK
Ryerson University,
Environmental Applied Science and Management
Data Science Laboratory
350 Voctoria Street, Toronto, M5B 2K3
+1-647-866-9450
sedef.akinlikocak@ryeson.ca

ABSTRACT

While research results exist in energy efficient hardware and its components to achieve environmental sustainability, major research is needed to relate energy consumption of hardware to energy consumption of its executing software. Since software is playing an increasing role in supporting our society its energy efficiency and environmental impact become more important. Green IT and green software aim to achieve environmentally sustainable computing. However applying this concept to the existing software systems to meet business demand or designing new green and sustainable software are complex tasks. Applying environmental requirements brings new trade-offs in addition to software quality requirements. Therefore new trade-off analysis needs to be done as regards to environmental sustainability requirements. Moreover, how quality requirements relate to environmental sustainability also needs to be investigated.

This doctoral research focuses to investigate the trade-off between software quality requirements and environmental sustainability (i.e., become greener) by means of empirical analyses and controlled experiments on different software contexts (e.g. improving database software functionality and developing new software). The research methodology of this study is based on quantitative research methods with the integration of empirical and case study methods.

Some preliminary results are already obtained from two different analyses; 1) empirical energy consumption analysis on database software, 2) qualitative analysis on quality and sustainability of software. The deviations for energy consumption suggesting that there are significant opportunities to save energy.

Categories and Subject Descriptors

D.2.1 **[Requirements/Specifications]:** Software Engineering Requirements/Specifications;

D.2.9 **[Management]:** Software Engineering, Software Process Models, Software Quality

General Terms

Measurement, Experimentation, Verification

Keywords

Green IT, Green Software, Environmental Sustainability, Energy Consumption, Energy Efficiency, Legacy Systems, Quality analysis and evaluation

1. INTRODUCTION

Information technologies (IT) can contribute to sustainability in at least two ways. First, by being more energy efficient, using less resource and resulting fewer CO_2 emissions. Second, by making IT processes more sustainable, i.e. decreasing the energy consumption and emissions of companies and individuals. In similar vein, software may contribute to decrease energy consumption (i.e., become greener) in at least two ways. Over the years, IT accounts for approximately 2% of world CO_2 emissions, a figure equivalent to aviation, according to Gartner estimates [1]. In fact, this 2% includes only the in-use phase of hardware. Software on the other hand in still the remaining 98% operationalizes the private sector in doing its business and the public sector in supporting society, as well as delivering end-user. Therefore, reducing the energy consumption and related carbon emission of the IT systems contribute to environmental sustainability. This global issue promotes the competition and forces companies to implementing energy efficient products and energy efficient technology services. In this context green hardware product design and production, as well as green service operation, gained a lot of importance to achieve environmental sustainability. However, software as the ultimate cause of hardware requirements shifts slowly into focus. Although lots of hardware solutions are proposed in the literature, there are rare cases which focus on software [2]. While low-level solutions and products already exist to understand energy efficiencies such as applications enabling consolidation via software virtualization [3] tools and methods to measure power consumption [4], they often rely on estimates or focus only on hardware rather than software. Recently, Naumann et al. [5] stated that there is a lack of models and descriptions regarding computer software.

When analyzing sustainability for a specific software system, there are four major dimensions (economic, social, environment and technical) which have to be considered. Economic, environmental and social dimensions from the Brundtland report [6], whereas technical is added for an adequate discussion of software-intensive systems. When all dimensions are in balance, greenness or sustainability of the software may only be achieved [7]. Therefore regarding new software product, sustainability aspects are needed to consider as early as possible into its design process.

On the other hand, software affects all aspects of our lives under ever-renewed forms, leveraging existing systems to sustainable and green are also important for the companies. They need to keep their software on demand with high quality level with respect to end users' requirements. This is also a challenging for the IT companies. Since quality requirements may create a rebound effect [8] which can turn savings in energy consumption. Each integrated quality feature is accompanied by increasing levels of energy consumption. Therefore, it is hard to maintain and sustain software as environmental friendly.

As a result, focusing on software as object of the optimisation with regard to sustainability this optimisation potentially needs to be analysed in terms of energy efficiency, business processes and quality requirements which are associated with sustainability aspects [9]. In terms of energy efficiency a key challenge is the definition of standard configurations of hardware and software to be measured. On the other hand for business processes and quality requirements the key challenges are more focus on methodological issues on measuring sustainability, investigating new trade-offs, and added values in terms software characteristics. Therefore companies and universities need to develop new empirical methods for green software patterns and practices addressing sustainability issues and energy efficiency.

The goal of this doctoral research is to contribute to the effort of the scientific community towards the reliable measurement the level of greenness of software systems and dynamic tradeoff model regarding software quality and environmental sustainability requirements for decision makers and companies. The main topics are:

1. **Energy efficiency of the software systems**
    - Goal: assess the impact of software functionality applications, provide appropriate measurement, metrics and methods to improve software sustainability
2. **The quality of green software**
    - Goal: design and develop dynamic decision making model on the quality of green and sustainable software

Although sustainability of the software can only be achieved when all of the dimensions are in balanced, in order to narrow down the scope of this research, only environmental sustainability is taken into consideration.

1.1 Issues to get advice on

The research questions of this study are not fully defined. Therefore we need advice on how to narrow my scope and what to focus on. Green and sustainable software is a new study area in the software engineering domain and it is challenging to go from this topic to specific research questions in my research. I would also like to discuss different theories I may use.

2. RELATED WORK

This section provides some related works on energy efficiency, software engineering and sustainability and software quality and sustainability.

2.1 Energy Efficiency

Efficiency defines how software behaves when it comes to saving resources and avoiding waste [10]. To reduce energy costs and contribute to global environmental goals, organizations consider green strategies increasingly often.

Most of the research on energy efficiency focuses to explore characterization of observed behavior via architectural or system-level simulations and further estimation of energy efficiency of the system [11, 12]. While early studies are centered on the embedded systems in general [13] the later studies are more focus on mobile systems especially java-based systems [14]. Software as a system may induce changes in energy consumption therefore how the software consumes energy and cause energy performance regressions. In this sense Hindle [15] introduced the green mining "an attempt to measure and model how software maintenance impacts a system's power usage". He argues that current research tends to focus on resource usage and ignores the actual patterns power consumption induced by software evolution and change.

These works provide significant insights to theoretical and system level however, there is a research gap on assessing the impact of improving functionality of software systems on energy consumption. Existing software systems or components can be modified to improve performance or other related attributes, to correct faults or adapt to a change environment. We believe it is important to do empirical analysis on software system as a starting point.

2.2 Software Engineering and Sustainability

The impacts of IT on sustainability with different levels have been discussed in many publications [16], [17], [18].

Software engineering domain is recently paying attention to sustainability and trying to contribute mostly in economic and environment aspects. Recent literature review shows that 44% of the studies were published in year 2012. This means that there is an emerging trend related to the research on software sustainability [19].

Software development process as well as use of the software systems may allow applying all of the sustainability aspects. Although definition of sustainability is well known the impact of dimensions are still tackling by software system in its application domain. For example, Tate [20] characterizes Sustainable Software Engineering "as the ability to react rapidly on any change in the business or technical environment" and considers only economic aspects. Following that Mahaux et al.[21] stated information technology changes behavior and therefore it has considerable effect on society and environment and analyzed of the usage processes of a software system with respect to social and environmental aspects. With this study social aspect is also taken into account. Johann et al. [22] offered an integrated view in regard to economy, society and the environment, and defined *Green and Sustainable Software* and *Green and Sustainable Software Engineering*. Shenoy and Eeratta [23] developed a model and described appropriate steps for developing green software. But this model just considers the environmental aspect of sustainability. Naumann et al. [5] observed that there is a lack of models and descriptions covers all sustainability aspects in the area of computer software. And they developed a reference *Greensoft* model inspired by lifecycle of a software product. The model shows that it is important to include all the negative and positive impacts of production process as well as the software product itself.

2.3 Software Quality and Sustainability

Various systems (e.g. energy systems, management systems, and computer systems) are brought sustainability as objectives for quality. In that sense models, tools and indexes are developed for the sustainability assessment of those systems. For example, Mocigemba [24] brought Sustainable Computing model with the focus on product, production process and consumption process assessments with regard to hardware and software Recently, Afgan [25] introduced the multi-criteria assessment method with economic, environmental and social indicators, as an appropriate tool for the quality assessment of the energy system, since the energy system is a good example for the identification of potential for sustainability development. In general, these efforts have resulted that the multi-dimensionality of sustainability requires interdisciplinary approach. Current discussions on the

sustainability requirements fosters on how to define, measure and assess sustainability as quality attribute of software [26]. The recent quality models are introduced by ISO (ISO/9126 and ISO ISO/IEC 25010) [27]. But there is no sustainability assessment considered as another quality aspect.

A first quality model for green and sustainable software was developed by Kern et al. [28]. It refers to a quality factors from ISO /IEC 25000 based on the direct and indirect related criteria of software. The quality model gives an overview of potential aspects which may be taken as sustainability criteria and metrics for software products. The model just considers the product quality factors, however, the quality aspects standardized in ISO /IEC 25000 are also related the quality in use. Calero et al. [29] consider sustainability as a new factor that affects software quality factors both product and quality in use. They presented a new quality model (ISO 2510+S) based in ISO/25010. In this model authors differentiate the quality factors according to the sustainability impact and describe related and unrelated sub-characteristics. All these studies discuss the relation with the software quality aspects and sustainability in general. And point out that product as well as the quality in use needs to be considered when assessing the sustainability of the software. On the other hand, any of the studies have mentioned on sustainability dimensions. Therefore my study may assist the software companies to make careful tradeoffs among not only technical and economic aspects but also social and environmental dimensions.

3. RESEARCH OBJECTIVES

Given the great interest in energy efficiency in IT industry there is a clear need of creating the knowledge base on green and sustainable software. Therefore given the importance of quality of green software this is a relevant area to study. Although some methodologies, measurements and tools are investigated to a great level, in my research we want to investigate gaps that need to be researched by practical experience mentioned in section 2.

3.1 Research Questions

The final research questions remains to be defined. The aim of the study is to investigate the tradeoff between software quality requirements and of sustainability (i.e., become greener) by means of empirical analyses and controlled experiments on different software contexts (e.g. improving legacy software and developing new software). Therefore there are 3 major challenge areas to indentify the research question [30]:

Measurements: How we find principles for engineering sustainable software as, for example, available for dependable systems? How principles used for and in functionality, performance or usability? What are the green software metrics that cover for sustainable and/or energy efficient software to summarize the software system as a whole?

Requirements: What types of requirements that guides to green and sustainable software development? How do they differ from traditional software quality requirements? How do sustainable software requirements used in traditional tradeoff models?

Quality: How does sustainability differ from software quality aspects? What would be new measures for all the characteristics of the software quality model that include sustainability and green requirements in order to support the developers in moving to a more sustainable software development culture?

For any of these challenges and questions, it is important to distinguish between the software being green and the software purpose being sustainable and green.

4. RESEARCH APPROACH, EMPRICAL STUDY DESING AND ARRANGEMENTS

The research methodology of this study is based on quantitative research methods with the integration of empirical and case study methods.

Applied research in the field of IT and software engineering commonly involves empirical analysis and formulation during the development of solutions to research questions.

This research will take place in three parts. The first part of the research Goal Question Metric (GQM) approach is used [30]. Two main goals are defined: 1) Assess the impact of improving functionality of software system on environmental sustainability 2) Assess the impact of combine effect of new features on energy consumption. This part includes the collection of existing empirical studies and the recognition of the state of the practice (tools and methodologies). Activities related to this step are:

- Collect empirical studies in literature in the scope of setting energy efficient software criteria,
- Identify in literature efficient measurement techniques,
- Select database software as a case study and its functionality features,
- Identify green software metrics for summarizing the system regarding computational efficiency and data efficiency.

Empirical design of the first part will allow demonstrating causality between an intervention and an outcome at a single software system.

The next activities are the empirical experiments driven by the following goal: assessing the relationship software functionality and energy efficiency. Activities related to this step are:
- Test the system measurements during features runtime.
- Analyze data and interpret findings.

The second part of the research, combination of Multi Criteria Decision Making (MCDM) and goal modeling (GM) approaches dynamic trade-off modeling for developing green and sustainable software. Two main goals are defined: 1) Investigate how software quality characteristics relate to sustainability as focus on the quality of green software 2) Prioritize the criteria and improve the quality of decision by providing information on trade-offs. The analysis includes experiments and type of survey that seeks to gather more relevant and specific objective and subjective information about software quality and sustainability aspects in order to arrive at a consensus on which criteria need to be included or prioritized as part of sustainable green and sustainable software.
Activities related to this step are;
- Identify software quality attributes,
I have adopted internal software product quality as the set of six different factors which are internally measured, i.e. as described by the ISO-IEC standard 25010 (see Figure 1).

Identify sustainability requirements for green and sustainable software,

- Identify conflicts between quality attributes and sustainability factors,

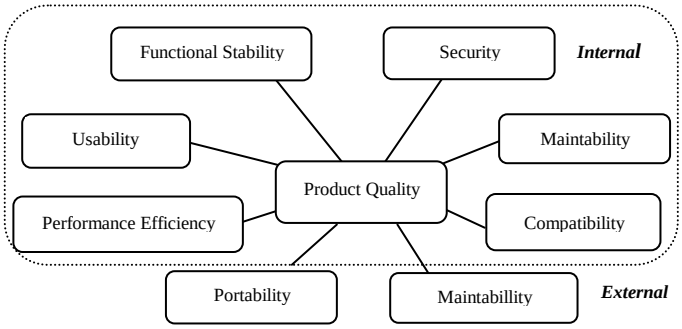

Figure 1. ISO 25010 Software Product Quality Factors

- Review the methods and models in the literature that used for decision making, trade off and requirement prioritization

- Develop the model that has to adapt from the system according to runtime status.

The last part is dynamic trade off modeling regarding requirement prioritization, software quality and sustainability. The aim of this part is to build recommendations using the model in order to support the developers and decision makers to reach more sustainable and green software development.

4.1 Definition of most important metrics

Green metrics are the criteria to quantify the green performance of IT systems. According to Kipp et al [30] green metrics are energy consumption-related metrics. The authors have stated that these indicators 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 have defined four clusters of metrics based on the Green Performance Indicators (GPI).

• **IT resource usage metrics** that are related to energy consumption of IT resources,

• **Lifecycle metrics** that allow setting applications to monitor energy consumption and develop energy aware indicators,

• **Energy impact metrics** that are related to the lifecycle impact on the environment, including the electricity, the power supply, the consumed material, and the CO_2 emissions, and,

• **Organizational metrics** that consider the assessment of additional costs due to energy-related initiatives.

I need to define two sets of metrics for the each part of the study. For the first part of the study which is the investigation the impact of software functionality on energy efficiency I have identified the metrics which are shown in Table 1. This is a simple metric set and it summarizes the system as a whole. The metrics are adopted from Kipp et al.[31], [32].

The computational efficiency and data efficiency are the most impact on software energy efficiency. The goal of computational efficiency is to complete a task more quickly. Therefore, measuring the CPU accomplishes the task in instructions then the overall energy required to complete the task will be important.

These metrics are used to evaluate different system components, for instance evaluating the CPU when executing instructions, evaluating a disk drive when performing I/O requests, or evaluating a server executing compression on a piece of data.

Table 1. Green metrics used to evaluate energy efficiency of the system

Chosen Green Metrics	Unit
IT Resource Usage Metrics	
CPU usage	%
I/O usage	%
Storage usage	%
Lifecycle Metrics	
Application performance	W/tps
Energy Impact Metrics	
System energy usage	kWh

Application performance metric allows us to measure the energy consumption per computing unit, e.g. #Transactions / kWh for a specific application type. This metric is given in Computation Unit / kWh. However, for the purposes of this research, it is more convenient to use the inverse value of application performance: The energy required to compute a single unit. This metric (TPC, 2010) is measured in Watt per transaction per second (W/tps) and is calculated as:

Work performance = Energy consumption / Work completed (1)

Data efficiency and data storage are another important impact on software energy efficiency. Therefore I examined storage usage metrics which refers to the entire storage utilization percentage for data read and writes operations on the corresponding storage device:

Storage usage = Used disk space / Allocated disk space (2)

In part of this research, the relationship between storage usage and space saving [33] have been examined. The following expressions have been used:

Space saving = 1 - (Compressed data size / Uncompressed data size), or *(3)*

Space saving = 1 – (Actual data size / Raw data size) (4)

I believe that these metrics give us an understanding of the system performance. The system energy usage metric implies the energy efficiency, but it ignores the system performance. The application performance metric helps to compare different systems and configurations, but is not adequate for independent evaluation of energy savings or performance. Therefore, I need to evaluate all the metrics together to determine the level of environmental impact of software.

As for the part two, product quality factors and sustainability criteria needs to be identified. For the software quality I have adopted internally measured software quality factors from ISO 9126-1 and its successor ISO 25010 (Figure 1). For the environmental sustainability I have identified four criteria adopted from Kipp et al [31] and Mahmoud and Ahmad [34]. Energy consumption which is total electricity consumption during operation; CO_2 emission which is amount of average carbon dioxide emissions; Green energy usage which is a usage of renewable energy; Return of green investment which is time it takes for green solutions to pay off or recuperate.

5. INITIAL RESULTS

5.1 Impact of Functionality

During my work on the first part of the study I began with a set of small exploratory experiments using one database software system in order to identify the effect of software functionality and energy consumption. In addition, these experiments helped me to identify the metrics. They will allow demonstrating causality between an intervention and an outcome at a single software system. At this part, the results obtained are due to interaction of running feature type and energy consumption.

Working with colleagues, two preliminary works [35], [36], [37] has been conducted. In our first study [35], [36], we have discussed the impact of improving existing software functionality or leveraging software systems according to end-users' requirements on the environment. Data compression is one of the software features that reduce the number of I/O operations while increasing CPU utilization. We have focused on the impact of data compression on energy consumption of software and investigated the trade-off between improving software functionality and reducing energy consumption of a software product.

Selected software under first part of the study is IBM-DB2 for Linux, UNIX and Windows Version 10.1. DB2 is a good candidate for the database software analysis, because it is a large software product present on the market since 1992 with a considerable market share. The latest release of DB2, version 10.1, introduced "*adaptive data compression"* a new type of data compression feature. This feature utilizes a number of compression techniques, including table-wide and page-wide compression. These compression techniques lead to significant reduction of storage space. However using this feature may lead to CPU overhead associated with compression and decompression of the data.

Data efficiency and data storage has important impact on software energy efficiency. Disk storage systems may often be the most expensive components of a database solution. Therefore, even a small reduction in the storage subsystem may result in substantial cost savings for the entire database solution. To this end, data compression reduces storage requirements, improves Input/ Output (I/O) efficiency, and provides quicker access to the data from the disk.

The effect of data compression actions on the amount of resources (time and electricity) needed to complete a certain workload was measured. Two database configurations were used: with compression and w/o compression.

Using the tools provided with the workload, we populated the database with 1 GB of raw data and generated 240 distinct queries associated with this dataset. Our reference workload was TPC-H[1]. It is created by the Transaction Processing Performance Council[2] and is used as the industry standard for measuring database performance. The workload consists of a set of business-oriented ad-hoc queries. The database has been designed to have broad industry-wide relevance[3]. The queries were executed sequentially for approximately two hours in a circular fashion on a Lenovo ThinkPad T60 laptop with 3GB of RAM, operating with Linux Ubuntu v.12.04.

The number of statements executed in a given time interval will be counted and measured the amount of electricity consumed by DB2 for each configuration. We have discovered that improvement of performance with data compression brings reduction of energy consumption per unit of work, reduces the cost of database maintenance and makes the database environmental friendly. As a result we have said that as software systems are modified, the impact of new features on the environment needs to be evaluated, and the option that satisfies green requirements should be chosen.

In our second work [38], we have extended our first experiments to incorporate new software features and functionalities to the analysis of DB2. Besides adaptive compression feature, '*DB2 design advisor"* examined to measure performance per watt of workload. The DB2 Design Advisor is a tool that can help significantly improve workload performance [39]. Two design advisor objects have been examined, index advisor and materializes query tables (MQT). The effects of three features were examined: (1) data compression, (2) index object suggested by design advisor, and (3) index and MQT suggested by design advisor. Therefore, we have been covered more real life database system scenarios. The experiment conducted with 1 GB of raw data on Lenovo ThinkPad T400 laptop with 4 GB of RAM and 2 CPU cores operating with Linux Ubuntu v.12.04. We focused on the CPU, I/O, work performance and total energy consumption of the software when processing a specific workload using 3 different features in six different scenarios. Most of the software products contain much functionality, so that more than one benefit may be possible. From this aspect, there is a need to use appropriate green metrics to characterize software systems with respect to their functionality and energy consumption. The results show that there is no pattern of relationship between software functionality improvement and energy consumption. Moreover the results demonstrate that different implementations of features create different IT resource usage behavior and different levels of energy consumption. Notably, combined effect of all the features is more significant than the individual effect of each improvement. We believe that software development managers would be able to make a trade-off between energy consumption and new software features if they are provided such an analysis.

This prior work is going to be extended with adding more functionality related features. Additionally comparative analysis will be performed with another industrial database system using similar performance features

5.2 Requirements Prioritization Framework for Green and Sustainable Software

Our latest work [39] evaluates environmental sustainability and software quality criteria using a well-known multi criteria decision making (MCDM) approach: Analytical Network Process (ANP). The aim is to prioritize green software criteria in order to use in trade-off models. ANP involves identification of the interrelationship and the intensity of importance and influence between different criteria.

MCDM provides a useful set of tools for understanding trade-offs and gaining insight into alternatives in the presence of multiple, usually conflicting decision criteria. The method is also used to prioritize the criteria and improve the quality of decision by

[1] http://www.tpc.org/tpch/

[2] http://www.tpc.org/information/about/abouttpc.asp

[3] Transaction Processing Performance Council, TPC-H Specifications,2012, http://www.tpc.org/tpch/spec/tpch2.14.4.pdf

providing information on trade-offs, increase confidence in decisions and provide insight into the criteria and alternatives [40]. In this study we proposed ANP framework that may be used to help identifying critical requirements and new trade-offs introduced by sustainability requirements.

The procedure for the prioritization of subjective criteria consists of three steps:

1. Identifying the criteria and sub-criteria.
2. Building of the ANP model.
3. Employing questionnaire and making pairwise comparison
4. Analyzing the interdependencies between the criteria and sub-criteria of the same cluster.

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

5.2.1 Identifying the criteria and sub-criteria

Major criteria related to sustainability of a software product are determined as described in the related literature. Then, several criteria that have direct relationships with major criteria are selected as sub-criteria. Table 2 summarizes all the criteria and sub-criteria that have been used to construct the ANP model. In this preliminary study, quality criteria were adapted from ISO/9126-1 which has been well known and studied in decade. The successor of the ISO/9126, ISO/25010 has been released recently. Further phases of the thesis, ISO/25010 has been considered to adopt quality criteria.

5.2.2 ANP Framework

The framework of our proposed ANP model illustrates the interactions among the goal and the criteria and the sub-criteria (Figure 2). The proposed decision model consists of two levels. The objective (developing green and sustainable software) is at the first level. In the second level, the criteria are listed. In the proposed hierarchy, outer dependencies and inner dependencies among sub-criteria as well as interaction between criteria are assumed to be present. The interdependencies and outer dependencies are shown in arrows.

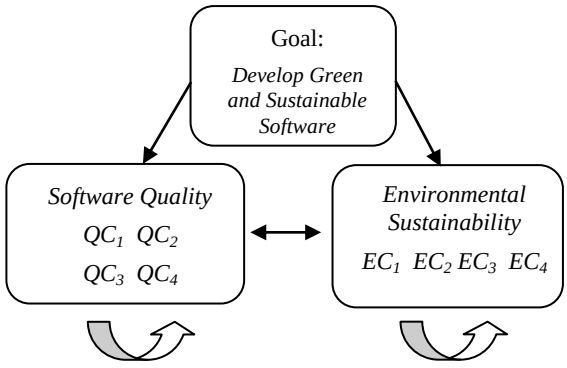

Figure 2. Framework and Interactions among Criteria

5.2.3 Employing questionnaire and making pairwise comparisons

Our proposed model has eight criteria. Thus, to compare the impact of each pair of elements on the main goal, we used questionnaires. For each criterion and its sub-criteria, we have prepared questions and pairwise comparison matrices for all the sub-criteria and the criteria to reflect the impact of criteria on each criterion. The questionnaire was filled out by ten decision makers (DM) from four different software development companies. Each DM is an expert in the software engineering field. The geometric mean of 10 questionnaires was used to form the output of the ANP approach [41].

Table 2. Selected criteria

Software Quality Criteria (QC) (ISO / 9126-1, 2001)		
QC_1	Functionality	System resources used to achieve required software functionality.
QC_2	Reliability	Totality of essential functions that the software product provides.
QC_3	Usability	Capability of the system to maintain its service provision under defined conditions for defined periods of time.
QC_4	Efficiency	Ease of use of a given function.
Environmental Sustainability Criteria (EC) (Kipp et al 2011, Mahmoud/Ahmad 2012)		
EC_1	Energy Consumption	Total electricity consumption during operation.
EC_2	CO_2 Emission	Amount of average carbon dioxide emissions.
EC_3	Green Energy Usage	Usage of renewable energy.
EC_4	Return of Green Investment	Time it takes for green solutions to pay off or recuperate.

5.2.4 Analysis of the interdependencies between the criteria and the sub-criteria and Results

The ANP model has been applied to the matrix operations in order to determine the pairwise comparisons, weight of each criterion and the overall priorities of the criteria. Super Decisions software product [42] has been used in order to calculate the results of final priorities of the eight sub-criteria.

The DM has to decide which of the two criteria sets, software quality and environmental sustainability, is more important in developing a green and sustainable software product. The geometric means of the responses are taken in terms of Saaty's scale of measurement (Table 3) [41], in order to determine aggregate individual judgments. We applied the ANP method on each comparison matrix and, calculated final weight of each sub criteria and criteria to identify priorities and interdependence.

The result of this study demonstrated that, the environmental sustainability criterion is the most preferred one to develop green and sustainable software with a weight 0.86. Among the four sub criteria of environment, the energy consumption criterion has the highest priority with 0.35. It is followed by CO_2 emission and return of green investment with 0.24 and 0.23 respectively. This means that energy consumption is the most important issue in developing sustainable green software companies regarding environment, followed by CO_2 which is closely related to energy consumption. Among the four sub criteria of quality, the efficiency has the largest priority with 0.34, followed by reliability and functionality with 0.24 and 0.23 respectively. This means that software efficiency considers how much information technology resources are used efficiently in terms of energy.

The traditional tradeoffs are made between conflicting quality attributes. Since quality attributes are prioritized, tradeoffs can be done using the priority weights. Without weights, the tradeoff reflects a more limited point of view. Since sustainability introduced environmental criteria with regards to green software, we may look at tradeoffs in different ways in terms of considering both environment and quality priority weights. Therefore, the

weights will enable DMs to make more accurate decisions when dealing with trade-off models.

Further work will be adopting new ISO 25010 quality model and constructing a dynamic trade-off model adopting ANP and GM regarding software quality and sustainability requirements.

6. Validity Threats and Their Control

The use of reliability and validity are common in quantitative research therefore it is important for this study for understanding the complex issues in measurement in theoretical and applied research settings.

Regarding to the validity, this research design and experimentation are well-founded with its concept, measurements and conclusion and corresponded accurately to the real world. The methodology of this research study does allow for internal, external and construct validity.

Construct validity: Exploratory experiments have been design to control whether the tests correspond to the cause thought to have been controlled and altered and the observed outcomes correspond to the effect thought to be measuring. Therefore, in this proposed research it is expected that every implemented feature for the software system highly correlate with increase rates of energy consumption. Moreover, the set of metrics is easily obtained. I used a proper set of metrics that are well known in the literature [31], [32], [33].

Internal validity: Confronting factors represent a major threat to the internal validity in such empirical studies. Selection bias is a prevalent problem and limits the validity of the studies. However, legacy software and corresponding features for testing are highly well-known and mostly used at the business environment. Therefore criteria are preventing this study from selection bias.

External validity: It is difficult to draw general conclusions from empirical studies in software engineering and our results are limited to the analyzed data and context. The experiments are not designed for the production environment. The testing environment is a laptop, which is tuned to minimize electricity consumption, sacrificing efficacy with consumer-grade operating system. However, the results could be extrapolated to a production system.

7. SUMMARY

This research presented an approach to discovering the tradeoff between software quality requirements and of sustainability (i.e., become greener) by means of empirical analyses and controlled experiments on different software contexts (e.g. improving legacy software and developing new software). Summarizing the previous work and the experience collected within the exploratory study, I believe that properties of this approach and its implementation in the framework appear to be very promising.

My next step will be to construct a dynamic goal model to help the new sustainability requirements trade-offs. I will also include all of the sustainability aspects to the decision making model and see how this affects the prioritization and trade-off. I will carry distribute more the questionnaire to different DMs.

8. ACKNOWLEDMENT

I would like to thank my supervisor Ayse Basar Bener and my co-supervisor Gulfem Isiklar Alptekin for their precious advices, and the reviewers of IDOESE 2013 for their observations that let me improve my plan.

9. REFERENCES

[1] Gartner Inc. 2008. Green IT: The new industry shockwave. In Presentation at symposium/ITXPO conference (2008)

[2] Dick, M., Naumann, S., 2010. Enhancing software engineering processes towards sustainable software product design, In *Proceedings of the EnviroInfo 2010 (Greve K., Cremers, A. B, (Eds.)*, EnviroInfo. (Cologne/Bonn, Germany, 2010), 706–715.

[3] Mergen, M. F., Uhlig, V., Krieger, O., and Xenidis, J. 2006. Virtualization for high-performance computing. ACM *SIGOPS Operating Systems Review*, 40, 8-11.

[4] Tiwari, V., Malik, S., Wolfe, A., & Lee, M. T. C. 1996. Instruction level power analysis and optimization of software. *In Technologies for wireless computing*, 139-154. Springer US.

[5] Naumann, S., Dick, M., Kern, E., Johann, T., 2011. The GREENSOFT model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and System*, 1, 294– 304.

[6] United Nations World Commission on Environment and Development. Report: Our Common Future. In United Nations Conference on Environment and Development, 1987

[7] Penzenstadler B. and H. Femmer. 2013. A generic model for sustainability with process- and product-specific instances. *In First Intl. Workshop on Green In Software Engineering and Green By Software Engineering*

[8] Hilty, L. M., Kohler, A., Scheele, F., Zahn, R. & Ruddy, T. 2006. Rebound effects of progress in information technology. *Poseis & Praxis: International Journal or Technology and Assessment and Ethics of Science*, 4, 19-38

[9] Tiwari, V., Malik, S., Wolfe, A., Tien-Chien Lee, M., 1996. Instruction Level Power Analysis and Optimization of *Software. The Journal of VLSI Signal Processing*, 13, 223–238.

[10] Taina, J. (2011): Good, Bad, and Beautiful Software - In Search of Green Software Quality Factors. *CEPIS UPGRADE, XII*, 4, 22–27.

[11] Yue, J., Zhu, Y., Cai, Z., & Lin, L. 2010. Energy and thermal aware buffer cache replacement algorithm. *In Mass Storage Systems and Technologies (MSST)*, (2010 IEEE 26th Symposium, May, 2010), 1-10.

[12] Li, Z., Grosu, R., Sehgal, P., Smolka, S. A., Stoller, S. D., Zadok, E., 2011. On the energy consumption and performance of system software, *In: Proceedings of the Israeli Experimental Systems Conference* (ACM SYSTOR '11, Haifa, Israel), 1-12.

[13] Li, Y., and Henkel, J. 1998. A framework for estimation and minimizing energy dissipation of embedded HW/SW systems. *In Proceedings of the 35th annual Design Automation Conference* (ACM), 188-193.

[14] Seo, C., Edwards, G., Malek, S., and Medvidovic, N. 2008. A framework for estimating the impact of a distributed software system's architectural style on its energy consumption. *In Software Architecture, 2008. WICSA 2008*. (Seventh Working IEEE/IFIP Conference on IEEE), 277-280.

[15] Hindle, A., 2012. Green mining: A methodology of relating software change to power consumption. *In: Proceeding of the Mining Software Repositories (MSR), In: Proceedings of the 9th IEEE Working Conference on IEEE*, (Zurich, Switzerland), 78-87.

[16] Erdmann, L., Hilty, L. , Goodman, J., Arnfalk, P. 2004. The future impact of ICTs on environmental sustainability. Institute for Prospective Technological Studies.

[17] Hilty et al. 2006. The relevance of information and communication technologies for environmental sustainability. *Env. Modelling & Software*, 21, 1618 – 1629.

[18] Tochtermann, K., Granitzer, G., Pillmann, W., & Geiger, W. 2008. ICT-ENSURE–A 7th Framework Program Support Action for Building the European Research Area in the Field of ICT for Environmental Sustainability. *In Environmental İnformatics and Industrial Ecology: Proc. of the 22nd Internat. Conf. on Informatics for Environmental Protection* (EnviroInfo 2008, Lüneburg), 456-63.

[19] Calero, C., Bertoa, M. F., Moraga, M. A. 2013. A systematic literature review for software sustainability measures. IEEE/ACM Workshop on Green and Sustainable Software (GREENS).

[20] Tate, K.2005. *Sustainable software development: An agile perspective.* Addison-Wesley Professional

[21] Mahaux, M., Heymans, P., Saval. G. 2011. Discovering sustainability requirements: *An experience report. Requirements Engineering: Foundation for Software Quality*, Springer, 19-33.

[22] Johann, T., Dick, M., Kern, E., Naumann, S. 2011. Sustainable development, sustainable software, and sustainable software engineering: An integrated approach. *International Symposium on Humanities, Science & Engineering Research*, 34-39.

[23] Shenoy, S., Eeratta. R. 2011. Green software development model: An approach towards sustainable software development. Annual IEEE India Conference (INDICON)

[24] Mocigemba, D., 2006. Sustainable computing. *Poiesis & Praxis: International Journal of Technology Assessment and Ethics of Science*, 4,.163–184.

[25] Afgan, N. H. 2010. Sustainability Paradigm. *Intelligent Energy System.Sustainability*, 2(12), 3812-3830.

[26] Penzenstadler, B., Mahaux, M., Salinesi,C.2012. Requirements engineering for sustainable systems. *International Working Conference on Requirements Engineering: Foundation for Software Quality*.

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

[28] Kern, E., Dick, M., Naumann, S., Guldner, A., Johann, T. 2013. Green software and green software engineering: definitions, measurements, and quality aspects. Int. Conf. ICT4S.B. Calero. 2013. 25010+s: A software quality model with sustainable characteristics, sustainability as an element of softwaremquality. Workshop on Green in Software Engineering Green by Software Engineering (GIBSE).

[29] Caldiera, V. R. B. G. and Rombach, H. D. 1994. The goal question metric approach. *Encyclopedia of software engineering*, 2, 528–532.

[30] Lago P., Kazman, R., Mayer, N., Morisio, M., Muller, H.A., Paulisch, F., 2013. Exploring Initial Challenges for Green Software Engineering. *ACM SIGSOFT Software Engineering Notes* (ICSE 2012-GREENS workshop), 38, 31–32.

[31] Kipp, A., Jiang, T., Fugini, M., 2011. Green Metrics for energy aware IT systems. In*: Proceedings of the 5th International Conference on Complex, Intelligent, and Software Intensive Systems* (CISIS Seoul, Korea), 241–248.

[32] Kipp, A., Jiang, T., Fugini, M., Salomie, I., 2011. Layered Green Performance Indicators. *Future Generation Computer Systems*, 28, 478-489.

[33] Salomon, D., 2006. *Data Compression: The Complete Reference*. 4th Ed., Springer.

[34] Mahmoud, S.S., Ahmad, I., 2012. Green Performance Indicators for Energy Aware IT Systems: Survey and Assessment. *Journal of Green Engineering*, 3, 33–69.

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

[36] Miranskyy, A., Akinli Kocak, S., Cialini, E., and Basar Bener, 2013. A. Save energy with the DB2 10.1 for Linux, UNIX, and Windows data compression feature, IBM DeveloperWoks, Technical Library, Available at: http://www.ibm.com/developerworks/data/library/techarticle/dm-1302db2compression/

[37] Akınlı Koçak, S., Miransky, A., Işıklar Alptekin, G., Başar Bener, A., Cialini, E., 2013. The Impact of New Software Functionality on Energy Consumption, *Environmental Modeling and Software*, Under Review.

[38] Zilio, D.C., Rao, J., Lightstone, S., Lohman, G., Storm, A., Garcia-Arellano, C., Fadden, S., 2004. DB2 design advisor: integrated automatic physical database design. In*: Proceedings of the 13th International Conference on Very Large Databases*, 30, 1087-1097.

[39] Akınlı Koçak, S., Gonzales Calienes. G., Işıklar Alptekin, G., Başar Bener. 2013. Requirements Prioritization Framework for Developing Green and Sustainable Software using ANP - based Decision Making, *In: Proceedings of the Environmental Informatics 2013* (EnviroInfo), Hamburg, Germany.

[40] Campanella, G., Ribeiro, R.A. 2011. A framework for dynamic multiple-criteria decision making. *Decision Support Systems*, 52, 52–60.

[41] Saaty, T.L., and Vargas, L. 2006. *Decision making with the Analytic Network Process: Economic, political, social and technological applications with benefits, opportunities, costs, risks*. Springer, NY.

[42] Creative Decisions Foundation. 2013. Super Decisions (ver. 2.2.6) [Open source Software]. Retrieve February 2, 2013. Available from: http://www.superdecisions.com.