Quality in Use and Software Greenability

Coral Calero, Mª Ángeles Moraga
Instituto de Tecnologías y Sistemas de la Información
University of Castilla-La Mancha
Ciudad Real, Spain
{Coral.Calero; MariaAngeles.Moraga}@uclm.es

Manuel F. Bertoa
Departamento de Lenguajes y Ciencias de la Computación
University of Malaga
Malaga, Spain
bertoa@lcc.uma.es

Leticia Duboc
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brasil.
leticia@ime.uerj.br

***Abstract*— Software sustainability has recently begun to gain importance. However, although proposals concerning what it is and how to achieve it are starting to appear, until now, there have been very few proposals on how to model it. Sustainable development consists of three dimensions: Social sustainability, Economic sustainability and Environmental sustainability; the latter being more closely related to technical aspects. There are also three environmental impact levels for ICTs: direct environmental effects of production and use of ICTs, indirect environmental impacts related to the effects of ICTs and indirect effects on the environment. In this paper, we focus on environmental sustainability and the first environmental impact level, and more concretely, on the direct environmental effect of software use. We specifically propose a greenability in use characteristic to be considered as part of the quality in use model proposed by the ISO 25010 standard. This model can be used using measures, indicators, or even Bayesian Networks in order to link it with product quality. We therefore present an example of a Bayesian Network that links product quality to greenability in use. Our eventual goal is to provide developers with indicators and guidelines on how to develop an environmentally friendly software product.**

***Index Terms*—Software quality, green software, greenability**

I. Introduction

Quality is currently among organizations' main goals because nowadays the industry made efforts to obtain the ISO9000 or CMMI (even been mandatory in the USA if a company want to collaborate with the government). The SWEBOK (Software Engineering Body of Knowledge) [16], the main reference guide of software engineers includes a specific chapter about software quality and how to apply it to the software engineering discipline. A large number of organizations provide similar products, thus permitting consumers to choose from a wide variety of brands. The survival of these organizations depends, to an increasing extent, on the quality of the products and services provided.

The need for quality is also present in the software industry, which has consequently become concerned about software quality. This has led to the appearance of the ISO/IEC 25000 series of standards, representing the evolution of the ISO/IEC 9126 and the ISO/IEC 14598 series. This family is divided into five divisions, one of which, the ISO/IEC 2501n (and more concretely the 25010), presents various software quality models.

However, none of these models considers sustainability or the ecological aspects of software products. This is, from our point of view, an important weakness of the standards, since sustainability is gaining more and more importance in society in general and in industry in particular. We believe this characteristic should also be considered in the software quality context. Our proposal is to complete the quality models of the standard in such a manner that we will be able to take sustainability into consideration when developing or evaluating a software product. This will allow to create the necessary foundation to incorporate into and assess greenability of a software product. In this paper, we present our progress with respect to this goal.

The reminder of this paper is organized as follows: Section two presents important aspects of sustainability in general. Section three discusses the importance of sustainability in the software context and presents the focus of our research. Section four describes the relevant aspects of the ISO/IEC 25010 standard and its quality models. The product greenability characteristic, proposed by the authors in a previous work, is also shown in the fourth section. In Section five, we propose the *greenability in use*, a new characteristic to be added to the quality in use model of the standard. Some examples of the levels of impact of this new characteristic are shown in Section six. Section seven shows how to combine the product quality and the quality in use models by means of Bayesian networks. Finally, Section eight presents our conclusions and future work.

II. Sustainability

One of humanity's current challenges is to conserve the environment and attain a sustainable economic and social development.

Sustainable development is commonly defined as "development that meets the needs of the present without compromising the ability of future generations to meet their own needs" [9]. According to the United Nations World Commission on Environment and Development [17], sustainable development needs to satisfy the requirements of the three dimensions of society, economy, and the environment.

Sustainability has recently become more and more important to businesses. A business that fails to have a sustainable development as one of its top priorities could receive considerable public criticism and subsequently lose market legitimacy [6]. The authors of [6] state that, according to a global IBM survey in 2008, 47% of organizations have started to redesign their business models on the basis of sustainability, treating sustainable development as a new source of innovation, an opportunity for cutting costs, and an mechanism by which to gain competitive advantages; all of which can be summarized under the concept of "strategic sustainability", introduced by [15] .

As noted by [6], technology is doubly important to pursue strategic sustainability. On the one hand because it helps organizations tackle environmental issues (using web conferences, repositories, etc.) and on the other because technology is often responsible for major environmental degradation (amounts of energy consumed by the engineering processes needed to manufacture products). This mixed role puts technology organizations under tremendous conflicting pressures. Internally they are pressed to transform existing engineering processes to make them more environmentally friendly, while externally they are expected to design new products that improve the sustainability of society.

III. SOFTWARE SUSTAINABILITY

Although there have been initiatives related to Green IT, efforts in the Green software area are still in early stages. Software development should not remain indifferent to the need to construct software products that contribute towards sustainability during both their creation and use. Software is the core of any IT technology, and the way by which it is developed can greatly influence the activities that need this software to be accomplished, such as the functions provided or how the IT infrastructure is used.

However, while sustainability is a standardized practice in a number of engineering disciplines, efforts in software engineering are recent and still immature [14], and the way to achieve sustainable software is mainly by improving its power consumption [2]. However, this is a very restrictive interpretation of what software sustainability is.

Sustainable software is a "software whose direct and indirect negative impacts on economy, society, human beings and the environment that result from the development, deployment and usage of the software are minimal and/or have a positive effect on sustainable development" [5].

This idea can be extended to cover the whole software development process. Sustainable Software Engineering can thus be referred as "the art of defining and developing software products in such a way that the negative and positive impacts on sustainability that result and/or are expected to result from the software product over its whole lifecycle are continuously assessed, documented, and optimized" [4].

The UN identifies three dimensions for sustainable development – social sustainability, economical sustainability and environmental sustainability [17]. We relate them to software as follows:

- Social sustainability is related to software use (by whom, how and under what circumstances software may be used);
- Economic sustainability is related to aspects of the software business, but not to its development;
- Environmental sustainability deals with technical aspects of software development.

Software product development mainly affects the environment via the consumption of resources that occurs during its use and production. The most direct (and obvious) impact of a software product is energy consumption, but other resources may also have a negative impact on the software's sustainability. The use of a processor, increased needs of memory and disk storage, network utilization and bandwidth, the potential relocation of software production and use, among others, are also elements to take into account.

We believe that it is of prime importance to pay the necessary attention to the environmental dimension of sustainability in the software context, which we term as *green software* or *software greenability*. figure 1 provides a graphic representation of our research focus.

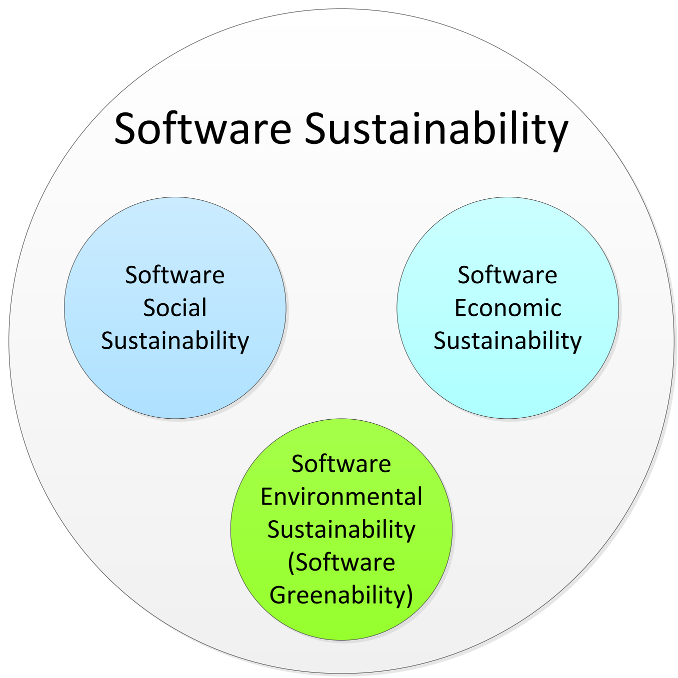

Fig. 1. Greenability as environmental sustainability

Furthermore, according to [1], there are three environmental impact levels of ICTs:

- 1st order: direct environmental effects of production and use of ICTs;
- 2nd order: indirect effects of using the system (resource and energy conservation owing to optimization or substitution of product materials);
- 3rd order: indirect, long term, effects of using the system (lifestyle changes that may impact on the environment).

We believe that, during development, the 1st level impact can be taken into account more easily because it is related to the direct effects of the software. The other levels will depend on how the software will be used, environmental aspects, and other external factors out of the developer's control. Therefore, our work focus on the 1st level impacts, which will to some extent have an influence on the second level (figure 2).

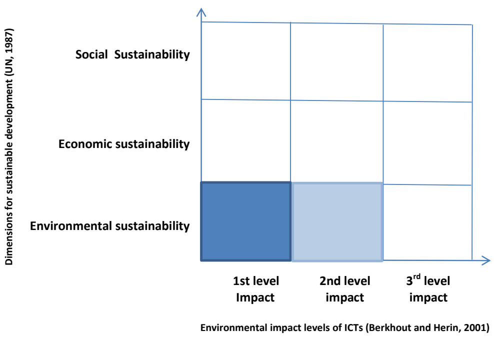

Fig. 2. TIC impact levels and the greenability

IV. SOFTWARE QUALITY AND GREENABILITY

When a software product is developed it is necessary to specify the requirements that the product should satisfy. Software requirements can be classified into functional and non-functional requirements.

The former should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs [8]. The functional requirements are therefore related to the "What" of a software product.

Non-functional requirements can be seen as requirements that constrain or set some quality attributes upon functionalities [7]. This means that non-functional quality requirements can be seen as the "How" of a software product.

Bearing in mind that software greenability is a means to improve a software product, we believe that it must be part of quality and, therefore, it is related to non-functional requirements. The first step should therefore be to include greenability in software quality.

According to ISO/IEC 25010 (figure 3), process quality influences product quality, which in turn influences quality in use. On the other hand, quality in use depends on the software product, which depends on process quality.

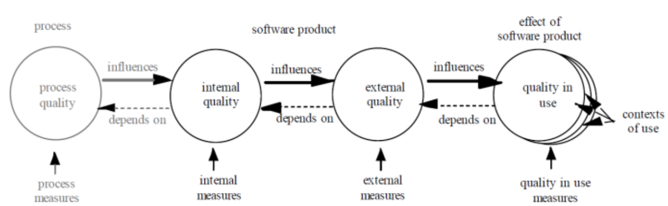

Fig. 3. Software quality life cycle [10]

The quality of a system is defined as the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value. These stated and implied needs are represented in the SQuaRE series of standards in various models (figure 4).

This international standard defines three quality models. Two of them, the product quality model and the quality in use model, are related to the product, while the third is related to the quality of data. Our work is focused on the first two.

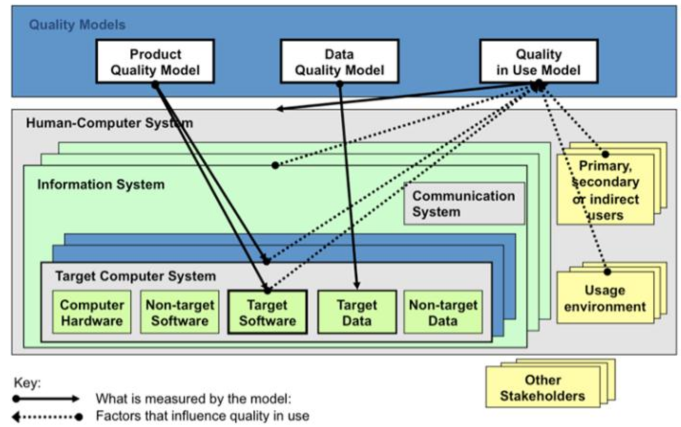

Fig. 4. Targets of quality model [10]

The software product quality model is composed of eight characteristics, which are further subdivided into sub-characteristics that can be measured internally or externally.

As is stated in the standard, "the product quality model is useful for specifying requirements, establishing measures, and performing quality evaluations" [10]. The quality characteristics defined can be used as a checklist in order to ensure a comprehensive treatment of quality requirements, thus providing a basis that can be used to estimate the consequent effort and activities that will be needed during systems development. The characteristics in the product quality model are intended to be used as a set when specifying or evaluating software product quality".

The system quality in use model is composed of five characteristics, which are further subdivided into sub-characteristics that can be measured when a product is used in a realistic context. These characteristics are thus a starting point for requirements, and can be used to measure the impact of the quality of the system in use and maintenance.

The software product quality characteristics can be used to specify and evaluate detailed characteristics of the software product that are prerequisites for achieving desired levels of quality in use.

A. Software Product Greenability

In [3], we have proposed the inclusion of sustainability in the software product model of the ISO/IEC 25010 standard [10]. However, naming the characteristic "sustainability" was not ideal because, as explained in section III, greenability deals with the technical aspects of the sustainability (see figure 1). Furthermore, the proposal did not consider capacity optimization as part of this greenability. We therefore refine that proposal as follows:

The proposal includes four sub-characteristics for the product greenability characteristic (figure 5):

- Energy efficiency. Degree of effectiveness and efficiency with which a software product consumes energy when performing its functions.
- Resource optimization. Degree to which the resources expended by a software product, when performing its functions, are used in an optimal manner. As in the standard, the authors consider that resources can include: other software products, the software and hardware configuration of the system, and materials (such as print paper, storage media).
- Capacity optimization. Degree to which the maximum limits of a product or system parameter meet requirements in an optimal manner, allocating only those which are necessary.
- Perdurability. Degree to which a software product can be used over a long period, being, therefore, easy to modify, adapt and reuse.

The next step in the process of integrating greenability into the software quality should be to include it in the quality in use model presented in ISO/IEC 25010.

Therefore, software product greenability can be defined as: Degree to which a product lasts over time, optimizing the parameters, the amounts of energy and the resources used.

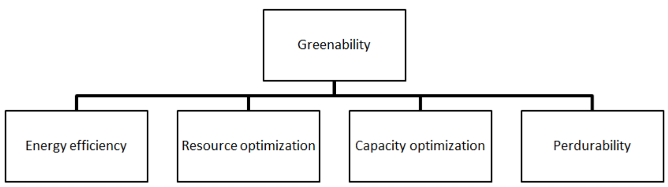

Fig. 5. Software product greenability

Greenability, however, is not only relevant to the software product. As shown in figure 3, software product quality influences quality in use and quality in use, depends on the software product quality. So, it is also necessary to study the inclusion for the greenability in the quality in use model.

V. Adding Greenability to the Quality in Use Model

The process of including a new characteristic in the standard requires a set of actions:

A. Working with the sub-characteristics: The identification and definition of the sub-characteristics is carried out.
B. Defining the new characteristics: new characteristics formally defined/refined, considering the sub-characteristics.
C. Reviewing quality in use characteristics: the model is reviewed in order to check whether it is affected by the inclusion of the new characteristic.
D. Redefining quality in use: the quality in use definition is reviewed in order to include the new characteristic in it.

Note that this is not a linear process, but an iterative one, as the completion of one action may lead to the review of the previous ones.

A. Working with the sub-characteristics

We identified the sub-characteristics in two steps. First, we studied the model characteristics in order to determine which existing characteristics would affect greenability. Secondly, we considered adding new sub-characteristics not derived from the previous step. We have done this by using the definition of the characteristics provided in the standard trying to determine its influence on the greenability in use. We have consequently obtained the following sub-characteristics:

- *Efficiency Optimization*. Optimization of resources expended in relation to the accuracy and completeness with which users achieve goals. Relevant resources can include time consumption, software resources, etc.
- *User's environmental perception*. Degree to which users are satisfied with their perception of the consequences that the use of software will have on the environment.
- *Minimization of environmental effects*. Degree to which a product or system reduces the effects on the environment in the intended contexts of use.

B. Defining new characteristics

New characteristics are formally defined and refined, considering the sub-characteristics identified above. In this case, only one new characteristic has been identified:

Greenability in use. *Degree to which a software product can be used by optimizing its efficiency, by minimizing environmental effects and by improving the environmental user perception.*

Because of the two first steps, we have obtained the new characteristic and sub-characteristics shown in figure 6.

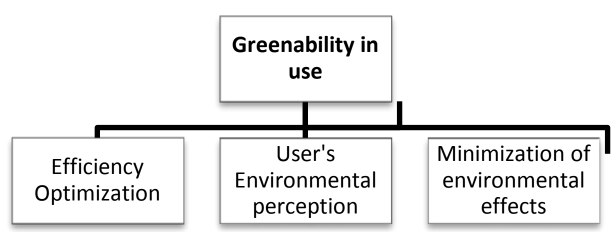

Fig. 6. Quality in use Greenability

C. Reviewing quality in use characteristics

It is now necessary to review the definitions of the other quality in use characteristics that could be affected by the inclusion of the new one. The following definitions have therefore been redefined:

Context coverage. *Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk,*

greenability *and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified*

Context completeness. Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk, ***greenability*** *and satisfaction in all the specified contexts of use*

Flexibility. Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk, ***greenability*** *and satisfaction in contexts beyond those initially specified in the requirements*

D. Redefining quality in use

Finally, the last step is to review the quality in use definition in such a manner that it takes into account the new added characteristic.

Quality in use *is the degree to which a product or system can be used by specific users to meet their needs in order to achieve specific goals with efficiency, freedom from risk,* ***greenability*** *and satisfaction in specific contexts of use*.

The result of this process is the new quality in use model shown in figure 7.

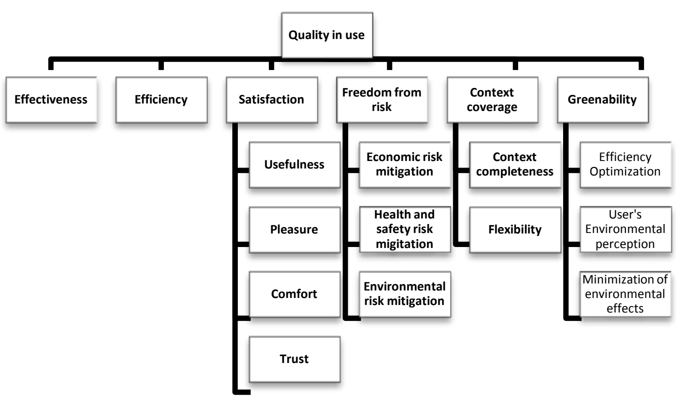

Fig. 7. Complete Quality in use model

VI. Studying the Impact Levels of ICTs with Regard to Greenability.

As previously indicated our work is focused on minimizing the first level impact of software, although it may also affect the second and third levels (see Section II and figure 2). It must be noted that effects at the third level are indirect and more difficult to consider during the software development.

In this section, we wish to provide an example of the relationship between a greenability in use sub-characteristics and the ICT impact levels.

A. Sub-characteristic: Efficiency Optimization

Domain application: Application with which to generate reports

First level impact

- Negative. The user wishes to obtain an annual report of the company's total expenditure grouped by department. However, the application does not provide this option and it is necessary to generate an expense report for each department. This implies that in order for the user to perform this task, s/he must use more resources.
- Positive. The application provides accurate information about the enterprise's light consumption expenditure.

Second level impact

- Negative: The data recovery processes are slower because data are scattered in several reports.
- Positive: Reports are stored in digital format, signifying that reports on paper are not necessary (e-materialization).

Third level impact

- A growth in IT-related services supposes a structural change and a new way of using resources at company level.

As will be noted, the development of a green software product impacts directly on the first level but also influences the second and may even reach the third.

Having defined the product quality (PQ) and the quality in use (QiU) models, the next step is to discover how to link them.

VII. Linking Product Quality and Quality in Use

By incorporating greenability to the software product quality and to the quality in use models of the ISO/IEC 25010 standard, we have created the foundations for assessing and achieving greenability in software. Therefore, we must consider the greenability of both the software product itself and of the software in use. However, as discussed in [12], the ISO/IEC 25010 standard (figure 3) states that there is a relationship between the product quality and the quality in use, but it does not say how to make this connection.

In fact, in the software quality field, most efforts have been made as regards product quality and it is difficult to find works on quality in use. The emphasis has principally been placed on assessing the quality of software products because they are more precisely defined in literature, can be more easily identified (and are thus easier to evaluate), and also because of the "dependency" of QiU on product quality.

This relationship between the product and the quality in-use of the software product would appear to be based on the assumption than having a product with high quality will guarantee a product with a high QiU. However, this is not necessarily true in many situations. The fact that a product has the best quality does not necessarily guarantee that the product will fulfill the user's needs in its context of use, especially when the overall quality (as perceived by the end-user) is composed of many conflicting factors. A Ferrari is not the best car to go to work in if you are a social worker in a deprived suburb in the outskirts of New York.

Our focus on the quality assessment is exactly the opposite: concentrating on the quality in use as the driving factor to consider when designing a software product, or when selecting the product that best fits a user's needs. There are several reasons why we feel the need to challenge the traditional approach used to evaluate the quality of a software product. Firstly, not all the product quality characteristics of a software product have the same influence on its QiU. This, together with the false assumption regarding the direct dependency between the product quality and the QiU mentioned above, frequently forces some of the product aspects (which are non-critical for the end user) to be over-specified for the sake of ensuring a certain level of QiU. This unnecessarily increases costs, development efforts and, resource usage, without a direct effect on the advantages that the end-user perceives.

We therefore focus on the QiU, and we analyze the relationship between the product quality and the QiU of a software product in the opposite direction to that which has traditionally occurred. This means applying a 'backward' analysis (we start with a given level of QiU and we wish to determine the minimum level of product quality that will guarantee such a desired quality in use), as opposed to the traditional "forward" analysis (by which we attempt to determine the level of QiU of a software product, given a measured level of product quality). In order to obtain a (good) level of quality in use, the goal would in fact be to be able to select the reduced set of really relevant product quality sub-characteristics that ensure the required level of quality. Focusing solely on them will avoid superfluous costs or irrelevant features which may unnecessarily increase the final impact on the environment and also the price of the product.

A. Use of Bayesian Networks

In order to determine the relationship between the quality in use (QiU) and the product quality (PQ), we need statistical methods and tools that can carry out backward analyses. However, the commonly used linear regression (LR) or principal component analysis (PCA) are not useful here because they conduct forward analysis and need initial numerical information at data level, which is in many cases difficult to obtain. However, Bayesian Belief Networks (or, simply, Bayesian Networks, BNs) can be very useful. A BN is a directed acyclic graph, whose nodes are the uncertain variables and whose edges are the casual or influential links between variables. A Conditional Probability Table (CPT) is associated with each node in order to denote such causal influence [11].

To define a BN it is necessary to: (1) provide the set of random variables (nodes) and the set of relationships (causal influence) among those variables; (2) build a graph structure with them; and (3) define conditional probability tables associated with the nodes. These tables determine the weight (strength) of the links of the graph and are used to calculate the probability distribution of each node in the BN.

The use of BNs thus allows us to model the different relationships among the characteristics and sub-characteristics of the product quality and the QiU, in addition to the degree of dependence or influence among them. This signifies that it is necessary to define the structure and conditional probability tables in which the uncertainty relationships among the BN nodes (characteristics) that we wish to build are reflected. Once the network has been defined, it is necessary to train it using a set of controlled experiments, so that it "learns".

The trained network can additionally be used to make inferences about the values of the variables in the network. Bayesian propagation algorithms use probability theory to make such inferences, using the information available (usually a set of observations or evidences). Such inferences can be abductive, and if we wish to determine the product quality sub-characteristics we must consider guaranteeing a required level of QiU (the cause that best explains the evidence); or predictive, if we wish to determine the probability of obtaining certain results in the future. All the variables in the network can therefore be used as either a source of information or an object of prediction, depending on the evidence available and on the goal of the diagnostic process.

B. Modelling the Bayesian Network

In this point, we show how to prepare a Bayesian Network in order to link product quality and quality in use. It should be noted that this is a general example as regards how to use this approach, and that it must be tailored to specific contexts.

Our working hypothesis is that the PQ has an influence on the QiU and that this influence can be modeled and studied by using a BN, in such a way that we can conduct a backward analysis of the required level of PQ to ensure a given level of QiU. We have successfully applied this approach previously [12], and other authors have also used the BN for the assessment of software quality [13].

In order to model the Bayesian Network, it is first necessary to identify the relationships between the PQ characteristics and the Quality in use. The relationship between PQ and QiU can be modeled by determining the characteristics of the former that affect the characteristics of the latter. As our present focus is on greenability, we shall focus the process solely on the greenability in use characteristic (although the same process can be applied to the other QiU characteristics).

It is then necessary to identify the PQ characteristics that have a significant influence on the greenability in use sub-characteristics. This is done by using the definitions provided in the standard along with those defined in this paper for the new sub-characteristics related to greenability. This process is made by a set of experts in quality. Firstly, they established the relationship among the characteristics independently. Next, they made a meeting to interchange opinions and agree on values. Table 1 shows these relationships by employing a matrix, in which the "X" indicates a relationship between these characteristics.

The next step is to determine the relationship between the sub-characteristics of the QP characteristics and the sub-characteristics of the greenability (using the information in Table I as a starting point). This step is necessary because if it were not taken we would be assuming that, for example, all the reliability sub-characteristics have the same influence on the efficiency optimization and this might not be true (it could perhaps be more closely related to fault tolerance than to maturity).

We have chosen only one of the greenability in use sub-characteristics in order to continue with the example. More specifically, and based on the results shown in Table I, we decided to work with the environmental user perception because it is the one with most interactions and because it appears to be the closest to the user.

		Greenability in Use		
		Efficiency optimization	User's environmental perception.	Minimization of environmental effects.
Product quality	Functional Suitability	X	X	
	Performance Efficiency	X	X	X
	Compatibility	X	X	X
	Usability	X	X	
	Reliability		X	
	Security			
	Maintainability	X	X	X
	Portability		X	X
	Greenability	X	X	X
		6	8	5

TABLE I. RELATIONSHIPS BETWEEN CHARACTERISTICS OF SOFTWARE PRODUCT QUALITY AND GREENABILITY IN USE

We have therefore identified the relationships shown in Table II. The "X" again indicates a relationship between these characteristics.

The level of influence (indicated by the relationships in Table I and Table II) may vary between different application domains. The relationships shown in both tables might then need to be tailored to other domains, but the method indicated in this paper is still valid.

It is necessary to use the information provided in Table II to model the Bayesian Network that reflects the relationships identified, taking into account the hierarchical structure of the models and the construction rules for Bayesian Networks. figure 8 shows the results.

As can be observed, there is a node for each characteristic and sub-characteristic, and arcs represent the relationships between the nodes (there is one arc for each X in Table II and another with which to connect a PQ characteristic with its sub-characteristics).

		User's environmental perception.
Portability	Adaptability	X
	Installability	
	Replaceability	X
Compatibility	Co-existence	
	Interoperability	X
Usability	Appropriateness recognisability	X
	Operability	
	User error protection	
	User interface aesthetics	
	Learnability	X
	Accessibility	
Functional Suitability	Functional Completeness	
	Functional Correctness	
	Functional Appropriateness	X

		User's environmental perception
Reliability	Maturity	X
	Availability	
	Fault tolerance	X
	Recoverability	X
Maintainability	Modularity	X
	Reusability	X
	Analysability	
	Modifiability	X
	Testeability	
Greenability	Energy efficiency	X
	Resource optimization	X
	Perdurability	X
Performance Efficiency	Time- behaviour	X
	Resource utilization	X
	Capacity	X

TABLE II. RELATIONSHIPS BETWEEN PRODUCT QUALITY SUB-CHARACTERISTICS AND GREENABILITY IN USE SUB-CHARACTERISTIC

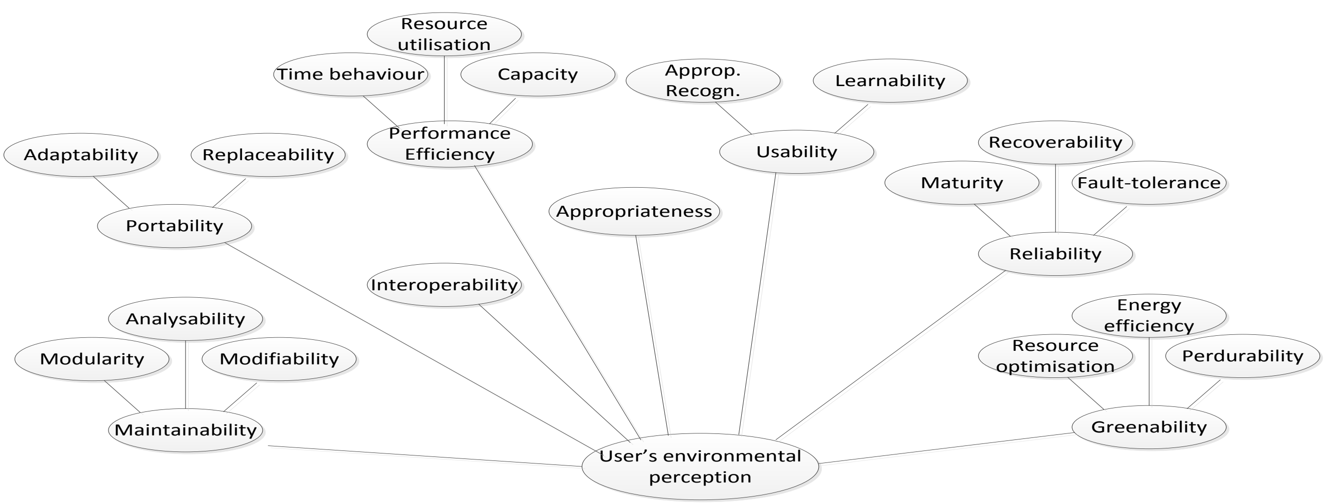

Fig. 8. BN for environmental user perception

Although this BN reflects the relationships identified, it produces a very high number of entries on the final node (that of the environmental user perception). The definition of the probability tables is therefore very laborious and cumbersome.

One practice that is commonly used to simplify the relationships in BNs is based on the introduction of synthetic nodes. In this case, we decided to introduce a synthetic node among the PQ characteristics that are conceptually related. The BN obtained (figure 9) drastically reduces the number of entries in the probability tables. The BN obtained could easily be used to create three individual BNs (one for each of the sub-trees that comprise the BN), work independently with them and then combine them to form the complete Bayesian Network.

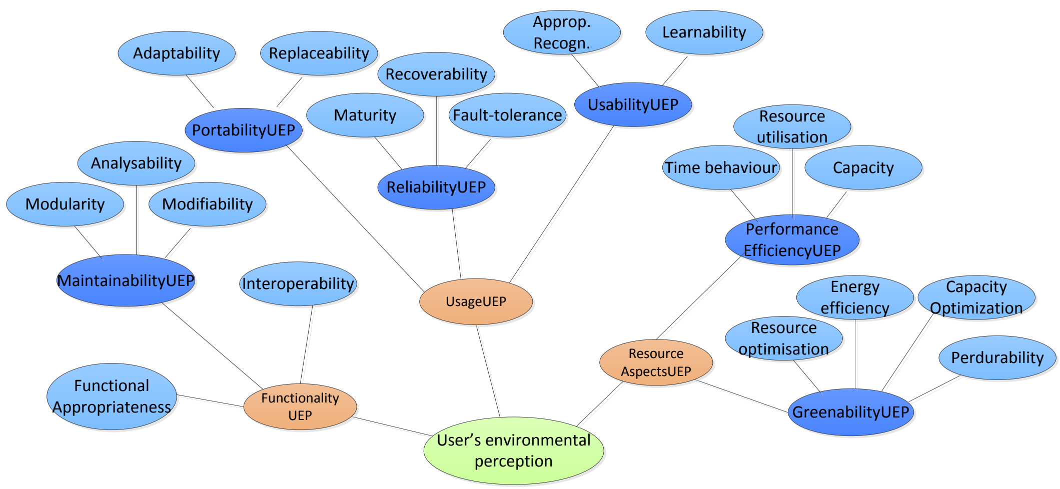

Fig. 9. The BN from Fig 7, with synthetic nodes

C. Adapt the Bayesian Network

In the previous point, we have shown an example of how to use Bayesian Networks to determine the influence of the greenability in use on the product quality. However, it is necessary to define the rest of BNs and adapt them to the specific context to which it is to be applied.

In order to carry out this adaptation, we must ensure that all the characteristics of the standard are applicable to this context and that no further characteristics are going to be needed. In addition, it should be determined whether to include new characteristics of the context by studying the state of the art, looking for other proposals, consulting experts, etc.

After these actions have been taken, we will be able to build the structure of the Bayesian network.

The next step is to create the probability tables. The influences of some given characteristics will obviously depend on the domain. That means it is vital to create tables to reflect the specific reality of a particular domain.

To do that, we must carry out experiments or surveys that allow us to obtain a series of data that serve as input to the network validation process. This is part of our future work in order to finish the Bayesian network and use it for working on the greenability assessment of a software product.

The final step in being able to use this network will be the definition of specific measurements for the software product we wish to measure. These measurements should be able to be calculated for the product; this will preferably be automatic, though that is not always possible. These measurements will be the ones which will serve as input to the external nodes of the network; their values should be changed into valid inputs to the network. The values will be propagated though the BN, via the nodes and by applying the probability tables, until the lower node is reached (the one about quality in use, or some of its characteristics)

D. Use of the Bayesian Networks

Once the BN's have been adapted to the specific context, they are ready to be used. These BN's can be used to carry out a forward or backward analysis.

In the forward analysis, we can determine the quality in use of a product once it has been created. In order to do we should define measurements for the external nodes that make up the input to the network.

In the backward analysis, we can determine the minimum values of external quality that the product needs to reach a desired level of quality in use. In that way, we can ascertain what values (for the measurements defined for each characteristic) the product should have for the input nodes.

VIII. Conclusions and Future work

Greenability is a means to improve the software product and should therefore be integrated into quality models, such as the ISO/IEC25010 standard. By doing so, greenability can be assessed and achieved during the software development process, just like other software qualities.

In this paper, we have proposed a extension of the quality in use model of the ISO/IEC25010 standard, includes a new characteristic: greenability. This is composed of three sub-characteristics: Efficiency Optimization, User's environmental perception, and Minimization of environmental effects.

Moreover, we consider that there is a direct influence between the product quality (which includes a greenability characteristic) and the quality in use (which already integrates greenability). We have therefore proposed a Bayesian Network that shows the relationships between both. As a future work we plan to work on this Bayesian Network, apply it to a specific domain and construct the probability tables in order to assess the greenability level of a given software product. We also plan to use Bayesian Networks for the greenability evaluation by means of measures and indicators. This is therefore another of our future works.

We also wish to continue with our work by studying the other aspects of sustainability, i.e., the economic and mainly the social aspects to which we believe special attention should be paid in order to indicate and mitigate some labor situations that currently occur in the software industry and that should be rejected immediately.

Acknowledgment

This work has been funded by the GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

References

[1] Berkhout, F. and J. Hertin, Impact of Information and Communication Technologies on Environmetal Sustainability: speculations and evidence. 2001

[2] Calero, C., M.F. Bertoa, and M.A. Moraga A Systematic Literature Review for Software Sustainability Measures in GREENS 2013 : Second International Workshop on Green and Sustainable Software. 2013. pp. 46-53.

[3] Calero, C., M.A. Moraga, and M.F. Bertoa Towards a software product sustainability model in Workshop on Sustainable Software for Science: Practice and Experiences 2013. arXiv:1309.1640 2013.

[4] Dick, M. and S. Naumann Enhancing software engineering processes towards sustainable software product design in EnviroInfo 2010, Integration of Environmental Information in Europe, Proceedings of the 24th International Conference on Informatics for Environmental Protection. 2010. pp. 706–715.

[5] Dick, M., S. Naumann, and N. Kuhn A Model and Selected Instances of Green and Sustainable Software. What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience in 9th IFIP TC 9 International Conference, HCC9 2010 and 1st IFIP TC 11 International Conference, CIP 2010, Held as Part of WCC 2010. 2010. pp. 248–259. Available from: http://dx.doi.org/10.1007/978-3-642-15479-9_24.

[6] Du, W.D., S.L. Pan, and M. Zuo, "How to Balance Sustainability and Profitability in Technology Organizations: An Ambidextrous Perspective.". IEEE Transactions on engineering management. Vol. 60(2): pp. 366-385 10.1109/TEM.2012.2206113, 2013

[7] Glinz, M., On non-functional requirements. "International Conference on Requirements Engineering": pp. 21-26, 2007

[8] IEEE 830, IEEE Recommended Practice for Software Requirements Specifications. . 1998

[9] ISO 26000 Guidance on Social Responsibility. 2010 Available from: https://www.iso.org/obp/ui/#iso:std:iso:26000:ed-1:v1:en

[10] ISO/IEC 25010, Systems and software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software product quality and system quality in use models. 2010

[11] Jensen, F.V., Bayesian Networks and Decisions Graphs.: Springer-Verlag, 2001.

[12] Moraga, M.A., M.F. Bertoa, C. Morcillo, C. Calero, and A. Vallecillo Evaluating Quality-in-Use Using Bayesian Networks. in 12th ECOOP Workshop Quantitative Approaches on Object Oriented Software Engineering (QAOOSE 2008). 2008.

[13] Neil, M., P. Krause, and N.E. Fenton, Software Quality Prediction Using Bayesian Networks: Software Engineering with Computational Intelligence, Chapter 6, Kluwer, 2003.

[14] Penzenstadler, B., A. Raturi, D. Richardson, C. Calero, H. Femmer, and X. Franch Systematic Mapping Study on Software Engineering for Sustainability (SE4S) in 18th Intl. Conf on Evaluation and Assessment in Software Engineering. 2014.

[15] Sroufe, R. and J. Sarkis, Strategic Sustainability: The State of theArt in Corporate Environmental Management Systems: Sheffield, UK: Greenleef Publishing, 2007.

[16] SWEBOK, "SWEBOK V3.0. Guide to the Software Engineering Body of Knowledge". Bourque, P. and Fairley, R.E. (eds.), NJ. IEEE Computer Society. . Vol., 2014

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