




Article

# An Action Research for Improving the Sustainability Assessment Framework Instruments <sup>†</sup>

Nelly Condori-Fernandez <sup>1,2,\*</sup> , Patricia Lago <sup>2,\*</sup> , Miguel R. Luaces <sup>1</sup>  and Ángeles S. Places <sup>1</sup>

<sup>1</sup> Centro de Investigación CITIC, Universidade da Coruña, 15001 A Coruña, Spain; miguel.luaces@udc.es (M.R.L.); angeles.saavedra.places@udc.es (A.S.P.)

<sup>2</sup> Department of Computer Science, Faculty of Sciences, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands

\* Correspondence: n.condori.fernandez@udc.es (N.C.-F.); p.lago@vu.nl (P.L.); Tel.: +34-649834404 (N.C.-F.); +31-205987745 (P.L.)

<sup>†</sup> This paper is an extended version of our paper published in ICT4S 2019.

Received: 18 December 2019; Accepted: 13 February 2020; Published: 24 February 2020



**Abstract:** In the last years, software engineering researchers have defined sustainability as a quality requirement of software, but not enough effort has been devoted to develop new methods/techniques to support the analysis and assessment of software sustainability. In this study, we present the Sustainability Assessment Framework (SAF) that consists of two instruments: the software sustainability–quality model, and the architectural decision map. Then, we use participatory and technical action research in close collaboration with the software industry to validate the SAF regarding its applicability in specific cases. The unit of analysis of our study is a family of software products (Geographic Information System- and Mobile-based Workforce Management Systems) that aim to address sustainability goals (e.g., efficient collection of dead animals to mitigate social and environmental sustainability risks). The results show that the sustainability–quality model integrated with the architectural decision maps can be used to identify sustainability–quality requirements as design concerns because most of its quality attributes (QAs) have been either addressed in the software project or acknowledged as relevant (i.e., creating awareness on the relevance of the multidimensional sustainability nature of certain QAs). Moreover, the action–research method has been helpful to enrich the sustainability–quality model, by identifying missing QAs (e.g., regulation compliance, data privacy). Finally, the architectural decision maps have been found as useful to guide software architects/designers in their decision-making process.

**Keywords:** sustainability assessment; sustainability–quality model; decision map; action research; software-intensive systems

## 1. Introduction

Sustainability is defined as the “capacity to endure”. Accordingly, and from a purely technical perspective, it has been linked to the notion of software evolvability, or longevity, (e.g., [1]). Of course, software sustainability has a much broader scope.

The concept of sustainability in the context of software-intensive systems is growing in importance. Researchers have addressed sustainability as a key concern in various disciplines related to computer science, from artificial intelligence [2] and human–computer interaction [3] to software engineering [4,5] and networking [6]. In addition, sustainability is attracting increasing attention as a key driver for innovation in the software industry [7]. Thanks to this increasing interest in the interplay between software and sustainability, several efforts have been dedicated to understanding what software sustainability means (e.g., [8–11]) and how software engineering can help achieve sustainability goals (e.g., [12–15]).

### 1.1. Problem

Software quality assessment as such is not new. Assessment based on the notion of sustainability as a software quality property, however, is still emerging and poorly understood.

Both Lago et al. [10] and Venters et al. [9] have defined software sustainability in the context of so-called sustainability dimensions (e.g., economic, technical, social, environmental, individual). However, despite the agreed-upon multidimensional nature of sustainability, most research efforts have focused on (i) understanding how software specifically influences the environmental dimension of sustainability [16]; and (ii) measuring the greenness of software systems [17].

Consequently, how software should be assessed against sustainability concerns is still immature even though it is attracting increasing attention from both research and practice.

### 1.2. Contribution

The present research aims to cover the lack of methodological support on how to assess the multidimensional nature of software sustainability [18,19]. In previous work, Condori-Fernandez and Lago [11] proposed a sustainability–quality (SQ) model for software-intensive systems, which aims at characterizing the sustainability dimensions by (1) mapping how each quality attribute contributes to which dimension, and (2) if and how the quality attributes are inter-dependent. In this work, the authors consider four dimensions of sustainability (i.e., social, environmental, technical and economic) by using the definition provided in [20].

Despite some recent efforts to provide methodological support for designing sustainability in/by software systems (e.g., [15,21,22]), there is still a lack of suitable and validated instruments in industrial practice.

To fill this gap, in a previous work [20], we have presented an overview of our empirical research strategy based on action research, which was conducted to empirically validate the applicability of the SQ model. In this paper, the novel contribution to the work presented in [20] is threefold:

- First, it introduces a model of the Sustainability Assessment Framework (SAF) that consists of the SQ model [11,21,23] and the architectural decision map [15] proposed by the authors.
- Second, a replication of our original empirical study supported by action–research methodology [24,25] to validate/evaluate SAF in terms of applicability, and perceived efficacy (ease of use and usefulness). The SAF validation results as a consequence of applying the SQ Model and the architectural Decision Map notation in a software project that focuses on the achievement of sustainability goals.
- Third, we propose some improvements that should be considered for increasing the acceptance of the SAF instruments (Decision Map and SQ model) in practice.

The following sections provide a detailed account of our study. Section 2 presents the related work. Section 3 describes the SAF model on which our work is based. Section 4 describes the research method. Sections 5 and 6 report on our main findings regarding the validation of our framework. Further discussion is provided in Section 7. Section 8 discusses the validity threats and Section 9 concludes the paper.

## 2. Related Work

In this section, we discuss the related works that focus on the definition or application of assessment models.

Lago et al. [10] defined software sustainability based on a four-dimensional model that adds the technical dimension to the social, environmental and economic dimensions that already appear in the Brundtland report [26]. Venters et al. [27] conclude that the notion of software sustainability is a multifaceted concept and argue for a quantitative approach for measuring sustainability objectives.

Calero et al. [8], based on the ISO/IEC 25010 Standard, described the quality characteristics that should be considered in software sustainability. However, they define sustainability only in terms of energy consumption, resource optimization and perdurability (i.e., reusability, modifiability, and adaptability), and they do not consider the social, environmental and economic dimensions.

Regarding energy consumption, various studies have been conducted using assessment models. For instance, Yan et al. [28] developed a service-specific end-to-end energy consumption model based on the traffic characteristics and usage patterns of different mobile services. Pihkola et al. [29] analyzed the energy consumption of mobile data transfer and mobile networks in Finland. They found that energy efficiency per transferred gigabyte has significantly decreased, while the total energy consumption at a network level might increase in the future due to a significant increase in data usage.

Inspired by the CMM, Hankel and Lago [30] defined the SURF Green ICT Maturity Model (SGIMM (<https://goo.gl/f0ORLV>)) to assess the maturity of overall organizations with respect to Green ICT. To this aim, the SGIMM includes criteria in four areas, including *greening of ICT* and *greening of the primary processes*. In terms of software systems, these correspond to software energy efficiency and software energy awareness, respectively. In comparison to SGIMM, our approach is product-oriented, and it is orthogonal to the four sustainability dimensions defined in [10,11].

Recently, Lago introduced the architectural decision map (DM) method [15], which helps to scope quality requirements as design concerns along the four sustainability dimensions (i.e., technical, economic, social, environmental). In doing so, architects are supported during design decision making toward smart and sustainable software.

Becker et al. [31] added the individual as a fifth sustainability dimension in addition to the aforementioned sustainability dimensions. However, we argue that the social and individual dimensions share the same *social nature*. Furthermore, the first takes a broader perspective (e.g., organizations, society, stakeholder types), which is especially relevant in software architecture because it aims at capturing “the big picture”. Considering the individual as an additional dimension is only appropriate when their concerns must be addressed (e.g., in requirements engineering or human–computer interaction).

### 3. The SAF Model

To guide decision making from a software architect perspective, we propose a Software Sustainability Assessment Framework (SAF). It consists of the Sustainability–Quality Model (SQ model) [11,21] and the Architectural Decision Map (DM) [15]. In this section, we introduce the model of the framework that represents the most relevant elements and their relations (see Figure 1). The key elements of the Decision Map are classes colored in green (i.e., design concerns, sustainability expected impacts, and the effects among concerns), whereas the elements of the SQ model are colored in blue (e.g., Quality requirement, sustainability–quality dimension). However, the common elements relevant for both the SQ model and the Decision Map are colored in yellow.

Concerning the Decision Map, we can see three types of *expected impact*: (i) *Immediate impacts* refer to immediately observable changes. These are the concerns that are addressed within the current software project and are expected to be directly addressed by the architecture entities. (ii) *Enabling impacts* that arise from use over time. This includes the opportunity to consume more (or less) resources, but also shorten their useful life by obsolescence or substitution. (iii) *Systemic impacts* that refer to persistent changes observable at the macro-level (e.g., behavioral change, economic structural change).

We can also see that *quality requirements* and *sustainability-related requirements* are referred to as types of sustainability *design concerns*. The relationships among design concerns and software architecture entities are defined as *Effects*. We have three types of effects: *positive*, *negative*, and *undecided*. Regarding the SQ model, it is defined in terms of four *sustainability dimensions*: (i) *Technical dimension* addresses the long-term use of software-intensive systems and their appropriate evolution in an execution environment that continuously changes. (ii) *Economic dimension* focuses on preserving capital and (economic) value. (iii) *Social dimension* focuses on supporting current and future generations to have

the same or greater access to social resources by pursuing generational equity. For software-intensive systems, this dimension encompasses the direct support of social communities in any domain, as well as the support of activities or processes that indirectly create benefits for social communities. (iv) *Environmental dimension* aims at improving human welfare while protecting natural resources. For software-intensive systems, this dimension aims at addressing ecologic concerns, including energy efficiency and ecologic awareness creation.

As shown in Figure 1, each dimension is characterized by a set of *Quality requirements*, which can be inter-dependent. Such *dependency* is represented by an association class on the association relationship between quality requirements. Moreover, it can be of two types: (i) it is *inter-dimensional* if it relates a pair of quality requirements defined simultaneously in two different dimensions (e.g., security defined in the *technical* dimension can influence security in the *social* dimension), and (ii) it is *intra-dimensional* if a dependency exists between two different quality requirements defined within the same dimension (e.g., in the *technical* dimension, security may depend on reliability).

Since our SQ model provides support to both identify design concerns, and assess the qualities of the software architecture, a set of metrics are used for measuring the quality requirements, which should be measurable. Instances of metrics can be also defined for *sustainability-related requirements*, which is represented as a class that inherits from the design concern class.

It is important to remark that the SQ model can be used independently of the Decision Map. For instance, the SQ model might be used to aid the requirements prioritization process [13]. However, the Decision Map necessarily needs the SQ model for scoping the design concerns.

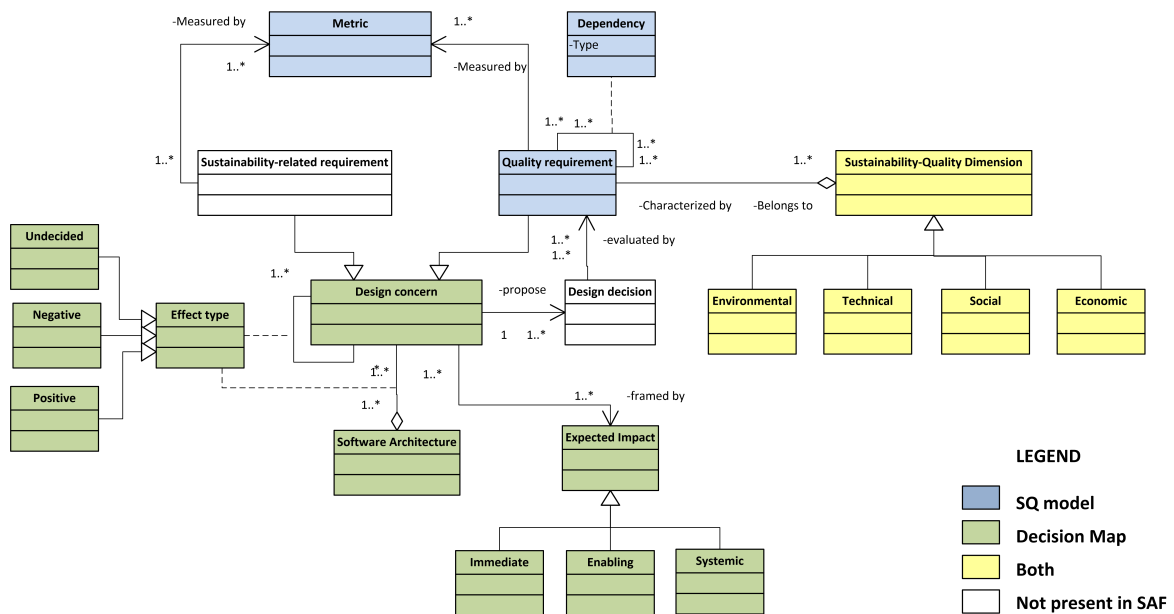


Figure 1. The Sustainability Assessment Framework (SAF) model.

#### 4. Research Method

The validation and evaluation of the SAF framework is conducted iteratively and incrementally by using two types of action research: (i) *technical action research* because we aim to scale up the action (treatment) to conditions of practice by using it in a particular problem [24]; and (ii) *participatory* because throughout the research process, the researchers are active in making decisions informed by the experiences the participating organization shares on applying the action [25,32]. Petersen et al. [32] defined the action as the treatment introduced by the researcher to induce a positive change in the company. In our work, the action is the Sustainability Assessment Framework that consists of two instruments (SQ model and DM).

As shown in Figure 2, the research strategy consists of four stages (grey-colored areas): diagnosis, action, feedback collection, and reflection. These stages are carried out at two different levels. We start the diagnosis at the *product-family level* to understand the common characteristics of the family of products and identifying sustainability-related issues. The output of the diagnosis is the selection of a product (software project), which is then used in the following three stages. At the *product level*, we plan the research to apply our framework to the selected software project, then we collect the feedback from the participants and reflect on it to refine the design of the framework to cover the identified needs and, as such, increase the SAF acceptance likelihood within the organization.

As shown in Figure 3, we follow iterative and incremental Research Cycles (RCs) at the product level, where each increment focuses on the instruments of the framework, namely the SQ Model and the Architectural Decision Map. It is important to notice that the interaction between the two RCs results from the synergistic nature of the two instruments: as the SQ Model is used to build the Decision Map, the two stages of Feedback collection and Reflection (in RC1) are carried out as part of the evaluation of the Decision Map (in RC2).

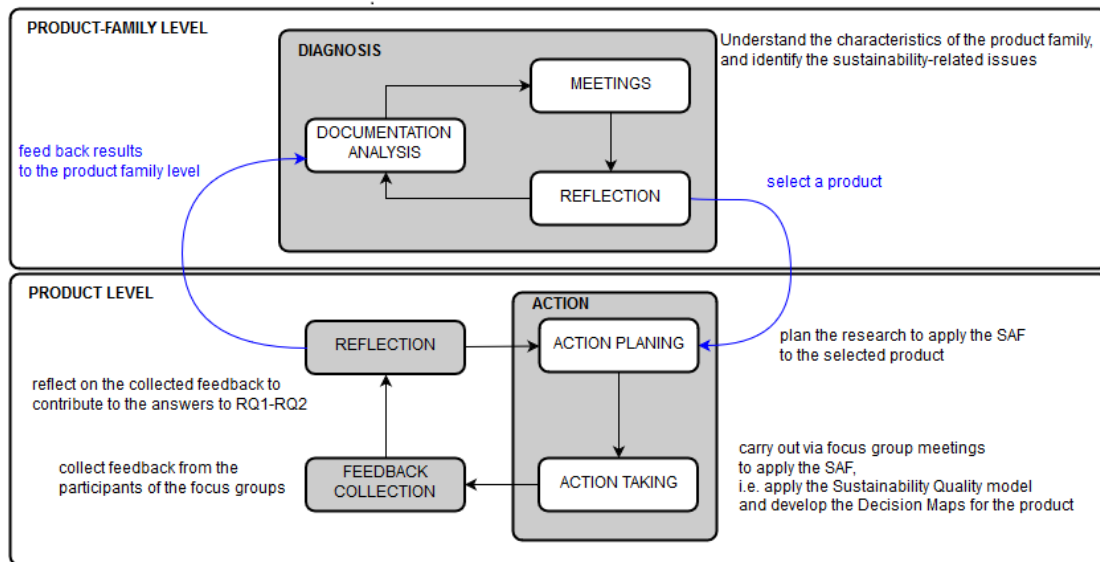


Figure 2. Participatory Technical-Action Research Process [20].

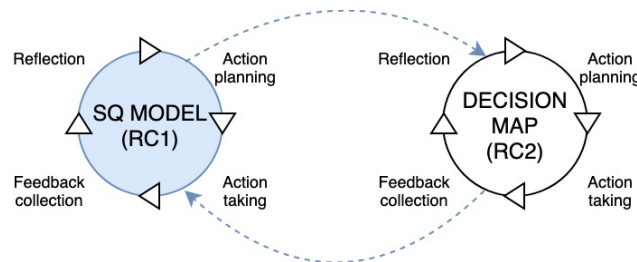


Figure 3. Research cycles conducted for validating/evaluating the SAF instruments at the product level.

Finally, the results from the Reflection stage (at the product level) are presented as feedback to the participating software company (at the product-family level). It is important to notice that other projects from the product-family can be considered in the subsequent iterations as long as the participants agree to it as part of the Diagnosis stage. The research questions that shape our study, the empirical research context (including the project and participants) and the unit of analysis are described in the following subsections.

#### 4.1. Goal and Research Questions, and Variables

The goal of our empirical research is to (i) validate the SAF instruments in terms of their applicability in practice, and consequently identify potential improvements to increase their acceptance in practice, and (ii) evaluate the framework with respect to the perceived ease of use and usefulness. From this goal we formulate the following research questions and sub-questions:

**RQ1:** How applicable is the framework to assess if the relevant sustainability–quality concerns have been covered in the software architecture at hand?

**RQ1.1:** How applicable is the SQ Model to identify requirements as sustainability–quality concerns from a software architecture perspective?

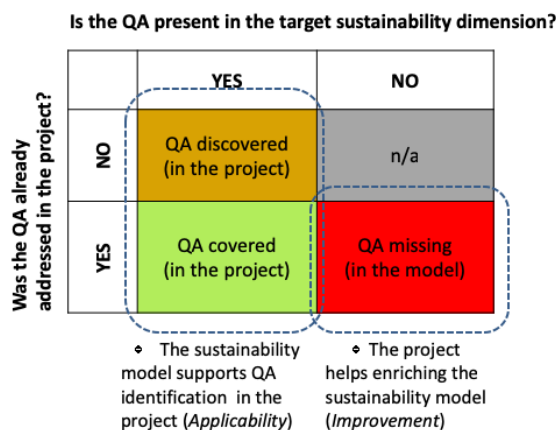
**RQ1.2:** How applicable is the Decision Map to frame the relevant sustainability–quality concerns of the software architecture at hand?

**RQ2:** How easy to use and useful is perceived the framework to assess if the relevant sustainability–quality concerns have been covered in the software architecture at hand?

The initial SQ model [11,21] and the Decision Map notation [15] were used as the starting point for answering RQ1. Naturally, the application of both instruments (action) to the selected products can also help to enrich the model with new sustainability–quality attributes, and the decision map notation with new insights. Then, user perceptions on ease of use and usefulness will be analyzed to answer RQ2. In the following, we define the **variables** identified from our research questions:

- **Applicability.** In Method Engineering research, applicability is investigated based on the situation in which it is applied [33]. According to Kitchenham et al. [34], applicability is defined as “*focusing on specific cases in which a method is used*”. Given that the applicability of a method can only be known when the method is used, action research has been used as a means to validate both SAF instruments to a specific case. Improvements to the framework that are realized as an outcome of implementing the action research were also considered as part of RQ1. Figure 4 shows the possible states that can be determined as a result of applying the SQ model (RQ1.1): Quality Attribute (QA) discovered (orange cell), QA covered (green cell), QA missing (red cell), and n/a when a QA is not observable (grey cell). These states are used in Section 5 and Tables 2 and 5 to report the results of the first research cycle (RC1).
- **Perceived ease of use.** This construct was originally defined by Fred Davis [35], which was instantiated to evaluate the framework as follows: the degree to which a person believes that using the SAF instruments would be free of effort.
- **Perceived usefulness.** It was also defined by Davis [35] and instantiated as follows: the degree to which a person believes that SAF will be effective in achieving its intended objective, which is *the ability to provide guidance for decision making*. To achieve this overall objective, we considered the following specific objectives:
  - *The SQ model should provide support for mapping the most relevant design concerns.* We consider that the usage of the SQ model as a tool for supporting early identification of sustainability–quality attributes should enable the achievement of this objective.
  - *A created DM should be able to be understood by people not involved in the process.* Considering that software sustainability is a broad topic that requires the participation of multiple stakeholders with different perspectives and backgrounds, the achievement of this objective is important to facilitate effective collaboration among the stakeholders.
  - *The SQ model and DM should help in analyzing the sustainability concerns of software architecture.*





**Figure 4.** The sustainability–quality (SQ) Model validation [20].

#### 4.2. Research Context

The research context consists of: (i) a family of software products developed by the Database Laboratory (LBD) of the University Of A Coruña. Enxenio (ENX), a spin-off software company, will maintain the software products in the future. The products have been developed as part of the GIRO Project (Acronym used for the project name: “Generating, Managing and Integrating Routes using OLAP”); and (ii) practitioners involved in the GIRO project and researchers, who participated in the action research.

##### 4.2.1. The GIRO Project

The GIRO Project ([http://lbd.udc.es/ProjectInformation.do?lang=es\\_ES&slug=GIRO](http://lbd.udc.es/ProjectInformation.do?lang=es_ES&slug=GIRO)), funded by the Spanish Center for Industrial Technological Development (CDTI) under the FEDER Innterconecta program (project reference ITC-20151247), aims to develop a Geographic Information System-based Mobile Workforce Management systems for six different customer companies (here anonymized), who are partners of the GIRO consortium and leaders in the following market areas:

- Company A: meat by-products treatment not intended for human consumption; in particular, dead animals collection in the northwest of Spain.
- Company B: management and valorization of organic waste transforming it in biogas.
- Company C: management of alarm systems in companies and homes.
- Company D: well-being services for elderly people residing in their own homes.
- Companies E and F: job safety analysis.

The GIRO project has a reference architecture that can be reused and adapted to address the specific requirements of the individual customer companies.

##### 4.2.2. Participants

The action–research team involved five participants. Two of them are researchers, who played three different roles in the action research [24]:

- Designer: Designing the instruments of the SAF framework.
- Helper: Using the instruments of the SAF framework to help the software company in getting awareness of the sustainability–quality requirements that were addressed in the project.
- Researcher: Drawing lessons learned about the instruments of the SAF framework (e.g., sustainability–quality model)

Therefore, both researchers acted as responsible for planning and executing the application of SAF in the selected case.

The remaining two participants were involved in the execution of the four stages of action research (diagnosis, action, feedback collection, and reflection). For the validation and evaluation of the SAF instruments, two research cycles were carried out as shown in Figure 3. The involvement of these participants was as follows:

- In the first action research cycle (RC1), two participants were practitioners from the Database Laboratory and the software company (Enxenio), who used the *SQ model* to identify relevant QAs that were not originally considered in the project or missing QAs that were not present in the Model. The roles of these participants were software analyst and software project manager. Besides, a software architect from Enxenio contributed to solving some doubts in the QAs identification process.
- In the second action research cycle (RC2), one of the three participants involved RC1 participated in the construction of the *Decision Map* for the selected product. This participant was the most senior software analyst, with 21 experience years in the development of Geographic Information Systems (GIS) and 5 years in GIS-Mobile Workforce Management Systems.

#### 4.3. Unit of Analysis

The unit of analysis of the study is the product family developed under the GIRO project by the software company, which is interested in knowing how sustainable is the existing reference software architecture. In particular, we focus on the Mobile Workforce Management (MWM) System developed for GESUGA (<https://www.gesuga.com/en/>), which is a company devoted to the treatment of meat by-products not intended for human consumption. GESUGA receives every day approximately 500 requests for dead animal collection and it must collect them in the shortest possible time.

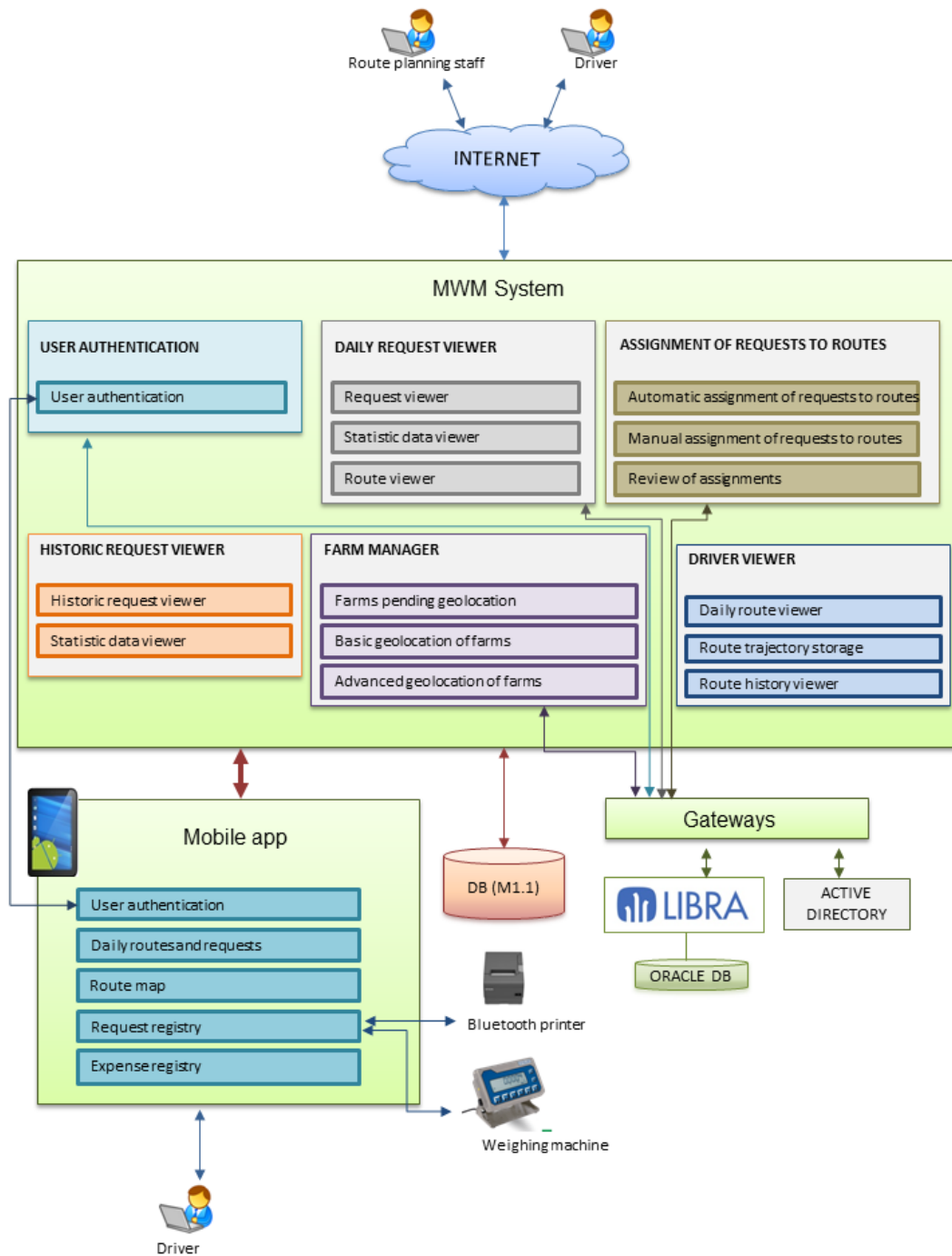
Figure 5 shows the software architecture of the system, which is composed of a web application and a mobile app. Two user profiles are identified, staff from the route planning department and drivers that are responsible for following the routes to collect the requests on the farms. The first profile uses the *daily request viewer* to browse through the collection requests received each day, to review the routes assigned, or to generate reports on the statistics for the day. This staff may also assign requests to routes manually (using the *assignment of requests to routes* module), even though most of the process is performed automatically and the staff only has to review the result. The staff may also view the historic information regarding requests (using the *historic request viewer* module), and manage the farms in the system, particularly their geolocation (using the *farm manager* module).

Drivers can view information about the requests they have assigned in the MWM System (using the *driver viewer* module). They also use the mobile app to review the routes and requests. During the collection at the farm, the driver has to record some information, namely the weight of the collected animal, and (if applicable) the reason the collection could not take place and any expenses that he or she might have incurred.

The system has to exchange information with existing software sub-systems. One of them is the ERP of the company, (*LIBRA*), which is used for all the management tasks in the company. *LIBRA* is involved to send requests and store results. Finally, the authentication is managed by an *Active Directory* server that is used as a single authentication system in the company.

To answer our research questions, the research activities (*action planning* and *action taking*, *feedback collection* and *reflection*) were carried out at the product level: the architecture of the Mobile Workforce Management (MWM) System (Figure 5).





**Figure 5.** Unit of Analysis: Software architecture of the GESUGA Mobile Workforce Management (MWM) System.

4.4. Procedure

In the following, we describe the specific procedure planned for validating and evaluating each instrument of the SAF.

#### 4.4.1. Procedure for the Sustainability–Quality Model

Through an *action plan* defined for applying our SQ model [21] (action/treatment) to the selected product, we carried out *action taking* (i.e., technical-documentation analysis and four focus group meetings [36]). Each focus-group meeting was planned by the researchers (the first author played the moderator’s role). The purpose of the corresponding focus groups and participants are shown in Table 1. Although these focus groups were small, the participants reflected on the analyzed QAs, by explaining their relevance for the selected project and giving examples of how some of the QAs were addressed. Because of this active discussion among participants (practitioners and researchers), we considered the four sessions as a focus group and not an interview. Each focus group had an average duration time of 60 min, and notes were taken during the focus group meetings by the moderator (*feedback collection*).

**Table 1.** Focus groups (FGs) conducted during the first research cycle.

FG	Purpose	LBD/ENX	ENX
FG1	Social sustainability	2	
FG2	Technical sustainability	2	1
FG3	Economic-Environmental sustainability	2	
FG4	Generic	1	1

#### 4.4.2. Procedure for the Decision Map

Our action plan was defined for evaluating the Decision Map through a single-case experiment. As shown in Figure 6 (right-hand side), the procedure starts with a *training session* aimed to introduce the main concepts and notation elements necessary to build appropriately a Decision Map [15].

Then, the participant was asked to use the improved version of the SQ model V1.1 (the outcome of the first research cycle [20], see the left-hand side of Figure 6) for *building a Decision Map* of the selected product. To carry out this second activity, some specific instructions and a template for creating the decision map were distributed by the researcher. The training material could be used freely. Moreover, the researcher acted as a helper during this activity to clarify possible doubts regarding the notation or the procedure.

Once a first version of the decision map was built, the participant was asked to *answer some questions* regarding his/her first impression on ease of use and usefulness of both SAF instruments. Difficulties that might have been experienced in building the decision maps, as well as suggestions for improving the framework, were also collected in the stage (*Feedback collection from session 1*).

To reinforce the learning of the decision maps, the researcher provided feedback to the participant using a reference decision map created for the same project (version A). *By comparing both versions of the decision maps*, the participant was then asked to highlight the main differences, answer for the second time questions regarding his/her perceptions on ease of use and usefulness, as well as additional comments or suggestions for improvements (*feedback collection from session 2*).

#### 4.5. Experiment Instrumentation

As shown in Figure 6, we prepared some training material to help the participants learn how to create a DM. Also, to conduct our single-case experiment we designed a set of instruments (namely, a Questionnaire, DM template, and Reference DM). More details about each instrument can be found in Appendix A.

In the following we present the results obtained from the *reflection* on the collected feedback to answer RQ1 and RQ2.

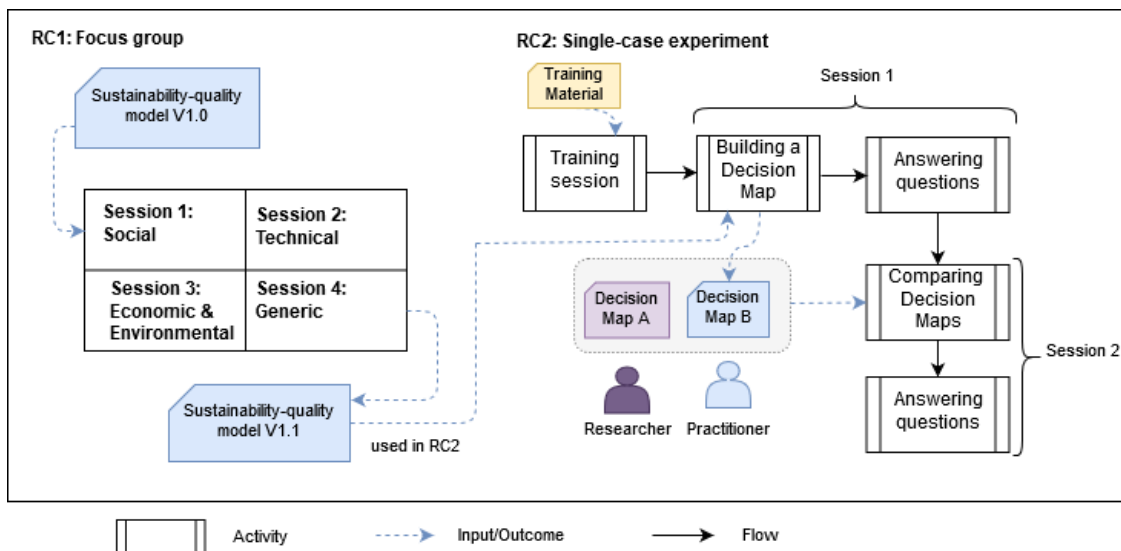


Figure 6. Action Planning applied to validate/evaluate SAF: SQ Model (Left) and Decision Map (Right).

## 5. Validation of the Sustainability–Quality Model

### 5.1. Applicability of the Sustainability–Quality Model

The following presents our study results. The QAs covered by the SQ Model that has been addressed in the project or that are relevant but have not been addressed yet are shown in Table 2. The new QAs that were missing in the SQ Model are shown in Table 5.

#### 5.1.1. Quality Attributes Covered in the Project

Given that there was a need in the project to share information with some existing systems (i.e., Libra, Active Directory) used by the customer company, **compatibility** was relevant to the project. First, w.r.t. the social sustainability dimension, its relevance is because users are required to use information from different systems. It is also relevant to the technical dimension because as long as the MWM system performs efficiently, even sharing with other software systems, its use will last longer.

The **effectiveness** of the software product was proved during the one month test period. Developers found that the MWM system enables the stakeholders to achieve goals such as scheduling new requests from the insurance company by the managers, or storing and tracking requests until they are fulfilled by drivers. This QA is relevant to the technical dimension because effectiveness contributes to the long-term usage of the MWM system. It is also relevant to the social dimension because it contributes to the well-being of the stakeholders (e.g., drivers are less stressed when the MWM system is used to track new requests). Finally, it is also relevant to the economic dimension because achieving the goals of the stakeholders effectively contributes to the long-term business objectives of the software company (i.e., customer satisfaction), as well as the business objectives of the stakeholders (e.g., saving costs).

The relevance of **efficiency** was confirmed by monitoring certain resources when users perform their tasks (e.g., the average time used to complete the tasks). This QA is relevant to the economic dimension because of the costs saved. It is also relevant to the environment dimension because of the reduced use of the resources. However, this is only partially confirmed because the usage of another type of resources, like the amount of energy used by the MWM system, has not been yet evaluated.

**Freedom from risk** is addressed in terms of *environmental risk mitigation* because the MWM system supports collecting dead animals under European regulations (CE 999/2001). Hence, the MWM system (i) avoids exposing people to potential disease-causing pathogens (social dimension), and (ii) reduces environmental concerns like potential contamination of air, soil, surface and sub-surface water (environmental dimension). The QA is also addressed in terms of *health and safety risk mitigation* because health risks to people (e.g., farmers and potential meat consumers) are mitigated using the MWM system (social dimension). However, the safety risk mitigation attribute has not been addressed for certain types of stakeholders. For instance, drivers are exposed to road safety risks. The system could implement new technologies such as vehicle-to-vehicle communication, to show the safer route to reach a farm.

The **functional suitability** attribute was confirmed by the stakeholders through a one month test. In terms of *functional appropriateness and functional correctness*, the stakeholders confirmed that the MWM system helped them to achieve their tasks. Functional suitability is relevant to the technical dimension because it contributes to the long-term use of the system. It is also relevant to the economic dimension because the software company does not have to invest much effort in corrective maintenance actions.

**Maintainability** was considered extremely relevant by the software company. The software company defined a common architecture for the six software systems of the GIRO project. Hence, the *modifiability and modularity* are key to facilitate the adaptation of the reference software architecture to be used in the product family and its evolution (technical dimension). Moreover, *modifiability and reusability* will help the company to address all relevant requirements of each software system of the product family reducing redesign costs and allowing a quicker response to company customers. Hence, these two QAs are relevant to the economic dimension, a fact that was not identified in the original model, in Table 2 we mark both contributions with a “+”.

The system shows an efficient performance (e.g., processing and response time) in delivering its main functionalities (i.e., finding an optimum route, allocating new requests in the planning). This implies that the system will be used longer, and therefore the relevance of **performance efficiency** in terms of *time behaviour* to both the technical and environmental dimensions was confirmed.

There was a consensus that the MWM system must be reliable performing under normal operations (*maturity*), and whenever it is required by end-users (*availability*). Therefore, **reliability** was confirmed as relevant to the technical, economic and environmental dimensions. Moreover, we found that this QA is also relevant to the social dimension because the social sustainability goals of these kind of systems can only be achieved if the software is available. Hence, we marked the social contribution with a “+” since it was not considered in the original model.

**Satisfaction**, defined in terms of *trust and usefulness*, is relevant to address both social and economic sustainability dimensions. The direct relation between user satisfaction and technology acceptance [37,38] has a positive impact on the social sustainability dimension because satisfied users will be in a much better position for getting access to social resources provided by the corresponding software system. Furthermore, both QAs are relevant to the economic dimension because the satisfaction of the customers is one of the primary business objectives of the software company. Finally, as long as the usefulness and trust of the system are positively valued by the end-users, the acceptance of the system will be prolonged, and hence both QAs contribute to the technical dimension as well.

**Security** was addressed in terms of *accountability, authenticity, confidentiality, and integrity*. Its contribution was confirmed to the social sustainability dimension. The role-based access control implementation addresses the *confidentiality* attribute because roles are assigned to a group of users that are authorized to access the system. *Authenticity* was also addressed, by using the login of each user as an identifier. *Accountability* was addressed tracing the actions of users for certain actions (i.e., updates of new routes). Finally, *integrity* was also considered as a good contributor to the technical dimension because the MWM system can prevent unauthorized access to the system. This new contribution was also marked with a “+”.

**Usability** was considered as relevant in terms of *appropriateness recognizability, operability, and user error protection*. The relevance of these attributes was to the social sustainability dimension. *User error protection* was addressed using field-validation, mandatory fields, and action confirmation. Protecting users against making errors is relevant to the social dimension because it is strongly related to the quality of user experience [39]. Furthermore, this QA was also considered relevant both to the technical and economic dimensions, which were also confirmed by two case studies. As both contributions were not considered in the original model, they are marked with a “+”. During the testing period, some issues related to *operability* had to be fixed (and therefore, the attribute was considered relevant). Finally, the *appropriateness recognizability* was also addressed because end-users recognized that the system is appropriate for meeting their needs.

**Table 2.** Sustainability–quality analysis of the MWM system (Green cell = QA is addressed, orange cell = QA is discovered as relevant, light-grey cell = QA is in the model but not relevant for the project, + = new contribution).

Characteristics	Attributes	Definition According to [13]	TECH	SOC	ENV	ECON
Compatibility	Co-existence	product can perform its functions efficiently while sharing environment and resources with other products.				
	Interoperability	a system can exchange information with other systems and use the information that has been exchanged.				
Context coverage	Context completeness	system can be used in all the specified contexts of use				
	Flexibility	system can be used in contexts beyond those initially specified in the requirements.				
Effectiveness	Effectiveness	accuracy and completeness with which users achieve specified goals.				
Efficiency	Efficiency	resources expended in relation to the accuracy and completeness with which users achieve goals.				
Freedom from risk	Economic risk mitigation	system mitigates the potential risk to financial status in the intended contexts of use.				
	Environmental risk mitigation	system mitigates the potential risk to property or the environment in the intended contexts of use.				
	Health and safety risk mitigation	system mitigates the potential risk to people in the intended contexts of use.				
Functional suitability	Functional appropriateness	the functions facilitate the accomplishment of specified tasks and objectives.				
	Functional correctness	system provides the correct results with the needed degree of precision.				
	Functional completeness	degree to which the set of functions covers all the specified tasks and user objectives.				
Maintainability	Modifiability	system can be effectively and efficiently modified without introducing defects or degrading existing product quality				
	Modularity	system is composed of components such that a change to one component has minimal impact on other components.				
	Reusability	an asset can be used in more than one system, or in building other assets				
	Testability	effectiveness and efficiency with which test criteria can be established for a system.				
Performance efficiency	Capacity	the maximum limits of a product or system parameter meet requirements.				
	Resource utilization	the amounts and types of resources used by a system, when performing its functions, meet requirements.				
	Time behaviour	response, processing times and throughput rates of a system, when performing its functions, meet requirements.				
Portability	Adaptability	system can effectively and efficiently be adapted for different or evolving hardware, software or usage environments.				
	Replaceability	product can be replaced by another specified software product for the same purpose in the same environment.				
Reliability	Availability	system is operational and accessible when required for use.				
	Fault tolerance	system operates as intended despite the presence of hardware or software faults.				
	Maturity	system meets needs for reliability under normal operation.				
	Recoverability	system can recover data affected and re-establish the desired state of the system in case of an interruption or a failure.				

Table 2. Cont.

Characteristics	Attributes	Definition According to [11]	TECH	SOC	ENV	ECON
Satisfaction	Trust	stakeholders has confidence that a product or system will behave as intended.				
	Usefulness	user is satisfied with their perceived achievement of pragmatic goals.				
Security	Accountability	actions of an entity can be traced uniquely to the entity.				
	Authenticity	the identity of a subject or resource can be proved to be the one claimed.				
	Confidentiality	system ensures that data are accessible only to those authorized to have access.				
	Integrity	system prevents unauthorized access to, or modification of, computer programs or data.				
Usability	Appropriateness recognizability	users can recognize whether a system is appropriate for their needs, even before it is implemented.				
	Learnability	system can be used to achieve specified goals of learning to use the system.				+
	Operability	system has attributes that make it easy to operate and control.				
	User error protection	system protects users against making errors.	+			+
Accessibility	Accessibility	system can be used by people with the widest range of characteristics and capabilities.				
Robustness	Robustness	Refers to the capability of the system to behave in an acceptable way in unexpected situations				
Survivability	Survivability	The degree to which a system continues to fulfil its mission by providing essential services in a timely manner in spite of the presence of attacks				

### 5.1.2. Quality Attributes Discovered in the Project

In this section, we present the QAs that were not addressed but were considered relevant for the project as a result of using the SQ model. Practitioners agreed on the relevance of addressing the **context coverage** requirement in terms of *context completeness and flexibility*. The one-month test period was used to verify many issues related to effectiveness and efficiency. However, some usage contexts were not covered by the testing process. An explicit specification of the different contexts of use was considered a relevant requirement that would contribute both to the technical and economic dimensions, because if the system is not able to work at any other potential context of use (not explicitly specified), higher effort and costs would be needed to improve the flexibility of the MWM system.

Although the *modifiability and reusability* attributes were considered for **maintainability** to contribute to the environmental dimension, they were not addressed in the project because of the lack of tools to determine to which extent the modifiable or reusable software artifacts of the system contribute to reducing the environmental impact.

**Resource utilization** was considered relevant to the environmental dimension of the project because the planner module has an impact on the optimization of resource utilization (e.g., trucks needed for collecting dead animals, gasoline consumed by trucks). However, the QA was not fully addressed because the actual amount of resources used by the system was not precisely determined.

Regarding **testability**, even though it was considered relevant for the technical sustainability dimension, it was not addressed because we could not find any evidence on how testable the software artifacts are. Similarly, the **capacity** of the system for the technical dimension was also not addressed because load or stress tests were not performed.

Although the company has knowledge of existing standard **accessibility** guidelines (e.g., ISO/IEC 40500:2012), this QA was not addressed as it was not considered important for the project. However, the participants agreed that removing interaction/communication barriers implementing some accessibility features to help certain users (e.g., deaf and hearing-impaired drivers) could be beneficial to our society in the long term.



Although **learnability** was not implemented, it was considered relevant for the social and economic dimensions. If features that help novel users to learn the system quickly, and features to help users progress were implemented, training costs would be saved by the company. A “+” is used to mark this new contribution to the economic dimension.

Finally, given that the MWM system could be affected by several unexpected situations (e.g., the GPS signal is lost), **robustness** is considered as relevant to address the technical sustainability dimension.

### 5.2. Improvements in the Model

In this section we report the missing QAs that were identified as relevant to be included in the SQ Model as well as their corresponding direct dependencies among sustainability dimensions (see Table 5):

**Data Privacy.** Given MWM system needs contextual data to be stored and shared for enabling (1) the management of work assignments (collection of dead animals from one farm to another) and (2) the tracking of real-time field workers, data privacy arises as a key requirement that should be considered as part of the model, in particular to the social dimension.

**Timeliness.** Given the route planning of the MWM system needs to be continuously adapted with the last requests, timeliness is relevant for allowing field workers (drivers) to continue with their job successfully (rapid collection of dead animals). As this requirement concerns the favorable and opportune time of having the right information(next pickup location), it has been also included as part of the social sustainability dimension.

**Regulation compliance.** Considering the main purpose of the MWM system, dead animal collection within a reasonable time, the software designers/developers should be aware of the existent European regulations to meet this requirement for contributing to (1) social sustainability since health risks are minimized, and (2) environmental sustainability since potential contamination of natural resources (e.g., water, air) is reduced.

**Scalability.** Given that currently collection services are offered only in the Galician region, there is an interest in scaling up the number of clients. Thus, scalability should be also considered as a contributor to the economic sustainability dimension because of the significant cost saving.

**Tailorability.** Given that the MWM system should allow users (drivers) to create a new configuration of functionality, tailorability is relevant for the SQ model as well. For instance, a driver can report some causes of delays using photos instead of text. This QA contributes to the technical and social sustainability dimensions. It is social because giving users the tailoring capability in their context of use can contribute to getting better access to the information provided by the system. It is also technical because the tailoring of the MWM system can contribute to longer-term usage.

## 6. Validation of the Decision Map and SAF Evaluation

In this section, we firstly present the analysis on the applicability of the DMs (RQ1). Then, we report on the evaluation results regarding the perceived ease of use and usefulness of SAF (RQ2). This data about the perceptions experienced by the practitioner was collected in two different moments, before and after the comparison with a reference DM, as shown in Figure 6.

### 6.1. Applicability of the Decision Map

By analyzing the created DM (see Figure A3 in Appendix B) as well as the difficulties experienced by the practitioner through the DM creation process, we observed the following:

- **Scoping sustainability design concerns:** Only quality attributes were considered as part of the scope. This might be because the practitioner is led to focus only on the attributes that were present in the SQ model, and in doing so is not stimulated to explore other sustainability concerns that might be relevant for the project at hand. Moreover, we observed that the created DM included

several quality attributes that were not directly related to the software architecture of the selected product (e.g., learnability, testability). This observation was also confirmed by the practitioner, who indicated “It is difficult to decide which attributes should be included. There are many, with four dimensions each”. “Sometimes I felt like I was giving structure to the attributes in general instead of describing how the attributes relate to the system”. This feedback confirms that we need to extend the SAF with an instrument that makes explicit the link between the concerns illustrated in a DM and the architecture design elements addressing such concerns.

- **Framing the expected impact:** Looking at the first version of the DM created by the practitioner, we found that most of the relevant design concerns were not appropriately located in the areas that correspond to the type of impact (namely, immediate, enabling, systemic). For instance, *maturity* (defined as reliability under normal operation) was considered as a design concern with enabling impact, while it could be measured immediately during the project. Similarly, *usefulness* was considered as a concern with an immediate impact instead of an enabling impact. In this case, the practitioner did not notice that the achievement of some pragmatic goals (i.e., in the project at hand, optimizing the routes planning) could require a longer time and also additional resources (e.g., the actual number of kilometers used by each truck with a minimum weight). These deficiencies found in the created DM were also confirmed by the practitioner, who indicated: “It was difficult to decide whether an attribute belongs to immediate, enabling or systemic scope of impact”. He also suggested adding some criteria that could be considered to help to frame the right type of expected impact.
- **Identifying the type of effects:** We observed that only positive effects were considered when some negative effects could have been also identified. Missing other types of effects could be a consequence of the lack of inclusion of other sustainability design concerns in the scope. Identifying a richer set of design concerns can potentially help to identify negative or undecided effects, too. For example, driver tracking is one of these sustainability design concerns that could have harmed the data privacy concern (quality attribute).

To complement this first analysis on the applicability of the DMs, in the next sub-section, we report the perceived ease of use and usefulness evaluation of SAF.

## 6.2. SAF Evaluation Based on Practitioner Perceptions

**Perceived ease of use (PEOU)** The practitioner’s answers were generally satisfied with the usage of DM. Specifically, the DM notation was evaluated as clear and understandable (see PEO1 in Table 3, 5 points); and the time required for building the DM of the project was not too long (PEO2, 4 points). As shown in the Table 3, the practitioner did not experience any changes in his perception of ease of use. These observations may support the hypothesis that the simplicity and clarity of the DM notation has a positive effect on the effort required for building a DM, and consequently also on the perceived ease of use.

**Table 3.** Graded feedback on the perceived ease of use of the Decision Map.

Items	Session 1					Session 2					
	1	2	3	4	5	1	2	3	4	5	
PEOU1	I found the notation of the decision maps as clear and understandable.									x	x
PEOU2	I have the feeling that the time expended in creating the decision map was too long.								x		x

**Perceived usefulness (PU)** We observed that the practitioner was more positive in the first session than in the second session. For instance, he found that the DM of any project should be well-suited to also be understood by people not involved in the creation process as an important characteristic of usefulness (see PU1 in Table 4). As the practitioner was not able to easily understand the DM created by the researcher (Version A), during the second session the rating for this item (PU1) was

changed from 4 to 3 points. As shown in the Table 4, this perception was also reflected when the practitioner was asked to answer the question related to the DM's capability to facilitate the analysis of the sustainability concerns (PU3). The observed decrease in PU3 (from 4 to 2 points) could be attributed to the existing relationship between understandability and analyzability. Thus, according to these observations, it seems that the practitioner got more aware of the usefulness of the DM during the second session. The exercise of comparing different versions of decision maps resulted to be harder than building a DM, and this hurt the perceived usefulness.

It is also interesting to note that the SQ model was perceived as useful to provide good support for mapping design concerns (PU2). This observation is consistent with our previous study [11], hence confirming the usefulness of our model for providing support to design and assessment activities.

**Table 4.** Graded feedback on the perceived usefulness of SAF.

Items	Session 1					Session 2				
	1	2	3	4	5	1	2	3	4	5
PU1	I believe that the decision maps are well-suited to be understood by people not involved in the process.			x				x		
PU2	I think the sustainability–quality model provides a good support for mapping the most important concerns.			x					x	
PU3	I think that the sustainability model and the decision maps help analyzing the sustainability concerns of the software architecture.			x				x		

Together, these observations provide important insights in both the applicability of the DM on a real case and how DMs can be improved. In the following section, we present the improvements that we plan to implement for facilitating the creation of DMs.

### 6.3. Improvements in the Decision Maps

The received feedback does not highlight the need for specific improvements in the DM notation. It does, however, point to the need for more guidance in drawing/creating DMs and in using the instruments of the SAF. In this direction, we have identified the following points for future improvement:

**Recommendation support for navigating the SQ model.** The practitioner needs support in navigating the SQ model and, when a QA is considered/selected, be proposed with possible related QAs (characteristics and/or sub-characteristics) for further consideration, as well as the ability to compare and contrast similar QAs. A simple **recommender** could provide the needed support.

Besides, so far the SQ model still carries a single definition per QA. Naturally, when a QA is classified in different sustainability dimensions, it is also defined differently. A simple example is *resource efficiency*, which in the technical dimension can be defined as the efficient consumption of, e.g., CPU time, while in the environmental dimension can be defined as the efficient consumption of energy. Accordingly, we plan to populate the SQ model with the **definitions** emerging of our study. In this respect, action research will help us refining extending the definitions emerging over time.

**Guidance to correctly position the design concerns in the expected impact.** As suggested during our study, we plan to make explicit the criteria for a QA to belong to the immediate-, enabling- or systemic-level of impact. Framing such criteria in a **checklist**, for example, will make it simple for the practitioner to check that each identified concern belongs to the impact level it is expected to have. Also, framing the criteria in an equivalent **decision graph** could support the identification of the most appropriate level of impact should the practitioner need support in deciding in the first place. Figure 7 shows an example of a decision graph for guiding in the identification of the type of impact (Immediate, Enabling, Systemic—represented as *nodes* of the decision graph).

In this example, the decision graph was built by considering three main criteria—represented as questions in the corresponding *diamond-shaped nodes*. Answers to these questions are represented as *decision edges* of the graph.

**Guidance for design space exploration:** The use of the SQ model seems to steer the attention of the practitioner away from design concerns other than QAs. We need to complement the SAF with an instrument that helps practitioners explore the design space more thoroughly, or systematically. To this aim, we plan to identify triggers for problem space (the *design space* is defined as the combination of the *problem space* including the design elements framing the problem, like design concerns, and the *solution space* including the elements of the design solution, like design decisions.) exploration similar to Tang et al. [40] (using decision patterns as triggers) and Razavian et al. [41] (using reflective questions). Our **triggers (maybe in the form of a checklist)** should stimulate the identification of the dependencies between concerns with positive or negative effects.

Lastly, an open research question is how we can support the practitioner in making explicit the link between DMs (e.g., a design concern) and the software architecture elements (e.g., addressing such concern). Future research will focus on defining an **architecture viewpoint** addressing this important new SAF instrument, which factually will support navigating between problem space and solution space.

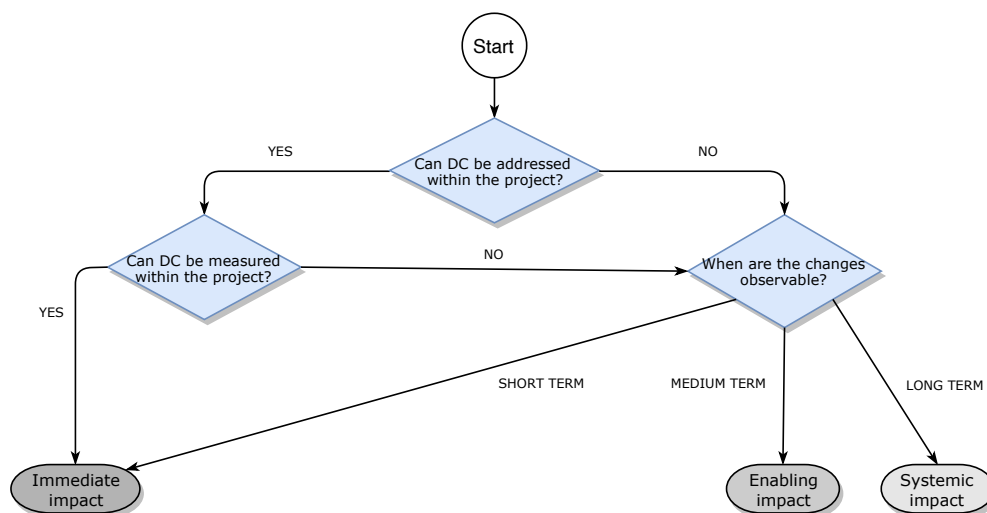


Figure 7. An example of Decision Graph guiding the selection of impact type for a Design Concern (DC).

## 7. Discussion

The present study was designed for validating the SAF instruments regarding its applicability in an existing software product (RQ1), and evaluating the perceptions experienced by the practitioner on ease of use and usefulness of the SAF (RQ2).

According to the results summarized in Table 2, we found that the applicability of our SQ model was effective (RQ1.1) due to the following facts: (i) most of the QAs present in the SQ model (cells colored in green) were corroborated as key requirements of the selected project; (ii) the practitioners were able to become aware of the relevance of certain QAs of the SQ model but that had not been addressed in the selected project (cells colored in orange).

Another interesting finding was that many QAs (e.g., co-existence, efficiency, availability, reusability, modifiability, trust, usefulness) contribute to more than one sustainability dimension. This confirms the multi-dimensional nature of sustainability (one of the principles of the Karlskrona Manifesto [42]). Moreover, this multidimensional characteristic can be useful to determine the relevance level of the QAs, which can be considered by software engineers when performing certain activities like design, assessment, and prioritization [11].

Further, the application of SAF instruments to the selected project (i.e., MWM system) helped us improve our SQ model in two ways:

- New QAs not been considered in the model were added. From Table 5, it can be seen that many of these new QAs contribute to the social sustainability dimension (i.e., data privacy, timeliness, regulation compliance, tailorability). However, only one QA (tailorability) was identified as a new contributor to the technical sustainability dimension. Overall, this result shows the importance of making explicit which sustainability dimension is relevant for which QA, so that significant metrics can be identified and monitored.
- New direct dependency relations (a direct dependency is defined as a finite set of ordered pairs of QAs, which is reflexive, symmetric and transitive [11]) were added as a consequence of identifying new contributions to different sustainability dimensions. Table 2 shows the QAs that were considered as part of this type of dependencies (cells are marked with a “+”). For example, the direct dependency between environmental and economic dimensions consists of four ordered pairs, whose QAs are efficiency, availability, modifiability, and reusability. This is because modifiability and reusability were discovered as new contributors to the economic dimension (+).

Concerning RQ1.2, the application of DMs to the MWM system did not uncover the need for any further improvements to the visual notation, but it helped us identify the need to support the practitioner in the reasoning process of drawing the DM, e.g., by adding checklists and recommendation systems to explore the design space and characterize the DM elements correctly (see Section 6.3).

Another improvement that contributes to addressing specific questions on the SAF applicability: RQ1.1 (SQ model applicability) and RQ1.2 (DM applicability) is that if a QA is found to contribute to multiple sustainability dimensions, its definition must be specialized for each dimension.

**Table 5.** New quality attributes and corresponding contributions to the sustainability dimensions.

Characteristics	Attributes	Definitions	TECH	SOC	ENV	ECON
Data Privacy	Data Privacy	privacy concerns arise wherever personally identifiable information is collected, stored, or used.				
Timeliness	Timeliness	the fact or quality of being done or occurring at a favourable or useful time.				
Regulation compliance	Regulation compliance	allows to draw conclusions about how well software adheres to application related regulations in laws.				
Scalability	Scalability	the ability of a computing process to be used or produced in a range of capabilities				
Tailorability	Tailorability	system’s capability to allow users to create or enable new configuration of functionality as well as control information provision.				

## 8. Threats to Validity

In this section we summarize the threats to the validity of the action research study [32], which were mostly identified during the first research cycle of the study [20].

**Internal validity.** As action research is highly context-dependent, the internal validity is low due to the different factors that might have influenced the results. Some of these factors are related to the specific type of software project, different expertise of stakeholders (e.g., experience of software designers/architects in dealing with quality requirements).

**Construct validity.** Action research is subjective as the results highly depend on the reflection of the researcher. In our study, some threats may occur due to the participation of:

- two participants, who are also part of the software company and they could not have provided an objective view of the situation.



- researchers as the designers of the action, who could have interpreted the results positively (selective bias) when reporting the results.

Both issues were partially solved involving multiple practitioners in the iterative discussions. For example, in the first research cycle, a third participant from the software company was involved. Furthermore, the researchers reviewed carefully the existing technical documentation to validate the data collected from the focus-group meetings.

Regarding the second research cycle, as we used only one senior participant from the company, we considered additional data sources to validate the Decision Map (i.e., perceptions, feedback, and two different versions of the decision map created by the practitioner and the researchers independently).

**External validity.** The specific social setting where the action is implemented can hinder the generalization of the results. In this paper, we have focused on organic waste management as a type of application domain. However, our findings could be applied to other projects with similar characteristics if the context is similar. Moreover, we consider that our model can be easily adaptable to other domains because it is based on the ISO/IEC 25010 standard [43].

## 9. Conclusions

The present research aimed to identify potential improvements to the Sustainability Assessment Framework (SAF) instruments (SQ model and Decision Map) to increase their acceptance likelihood in practice. The study was carried out through an action–research setting, where the SAF instruments were applied in one software product (a Mobile Workforce Management system developed for collecting dead animals in the northwest of Spain). As a result of this application (RQ1), from a practitioner perspective, we firstly found that the **SQ model** has shown to be supportive for the identification of (i) QAs that contribute to different sustainability dimensions (e.g., trust, modifiability, efficiency), whose information can be useful in the requirements prioritization process; and (ii) QAs that had not been yet addressed in the project (e.g., context-completeness, flexibility, testability, capacity), which might trigger the evolution of the software product. From a researcher’s perspective, our research has also helped to uncover new missing QAs that were identified as relevant to be included in the SQ model (i.e., regulation compliance, timeliness, data privacy, scalability, and tailorability).

Secondly, regarding the validation of the **Decision Map** with the same software product, the study helped us to uncover some difficulties experienced by the practitioner through the DM creation (i.e., scoping sustainability concerns, framing expected impact, and identifying type of effects). Our study also focused on understanding the practitioner perceptions on ease of use and usefulness of SAF (RQ2). In general, although the DM notation was perceived as clear and understandable, the perceived usefulness of the framework was affected due to the difficulty in (i) understanding a DM created by others (PU1), and (ii) analyzing the sustainability concerns of the software architecture (PU3).

Finally, improvements that we plan to implement and integrate as part of the framework were identified within the action–research environment (e.g., recommender for navigating the SQ model, decision graph for identifying correctly the type of expected impact). This action–research based work contributes to the existing effort on providing methodological support for the sustainability assessment of software-intensive systems. Further research needs to examine more closely the links between the DMs and software architecture elements. To do this, we plan to conduct an empirical study that involves practitioners and researchers interested in software sustainability design.

**Author Contributions:** Investigation, N.C.-F., P.L.; Methodology, N.C.-F., P.L.; Validation, M.R.L., A.S.P., N.C.-F., P.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has received partial support by Xunta de Galicia/FEDER-UE CSI (ED431G/01, ED431C 2017/58); Xunta de Galicia/FEDER-UE, ConectaPeme, GEMA (IN852A 2018/14); MINECO-AEI/FEDER-UE with the projects: Datos 4.0 (TIN2016-78011-C4-1-R), BIZDEVOPS (RTI2018-098309-B-C32), FLATCITY (TIN2016-77158-C4-3-R.); and FONDECYT with the Kuisisqa project (014-2019-FONDECYTBM-INC.INV)

**Conflicts of Interest:** The authors declare no conflict of interest.



## Appendix A

We designed the following material and instruments:

- **Training material.** It consists of slides that introduce the basic concepts and elements of the Decision Map (DM) notation. Some examples were also included as part of the training material. This material can be downloaded from the following link: <https://bit.ly/2tAtugy>.
- **Questionnaire.** It aims to gather practitioner's perceptions on ease of use and usefulness of the DM and the SQ model. Our questionnaire focuses on exploring these last two constructs (PEOU, PU), which were instantiated through two types of questions: five closed questions in a 5-points Likert scale (5 strongly agree, 4 agree, 3 neutral, 2 disagree, 1 strongly disagree), and four open-ended questions. Table A1 shows the items that were formulated for understanding the practitioner's perceptions of ease of use and usefulness of the SAF instruments (SQM and DM).
- **DM template.** It was created with the Draw.io editor tool, which was used by the practitioner in the first session of the experiment (see Figure A1).
- **Reference DM.** It was created by the researchers, and used in the second session by the practitioner. Figure A2 shows the reference DM used in the study.

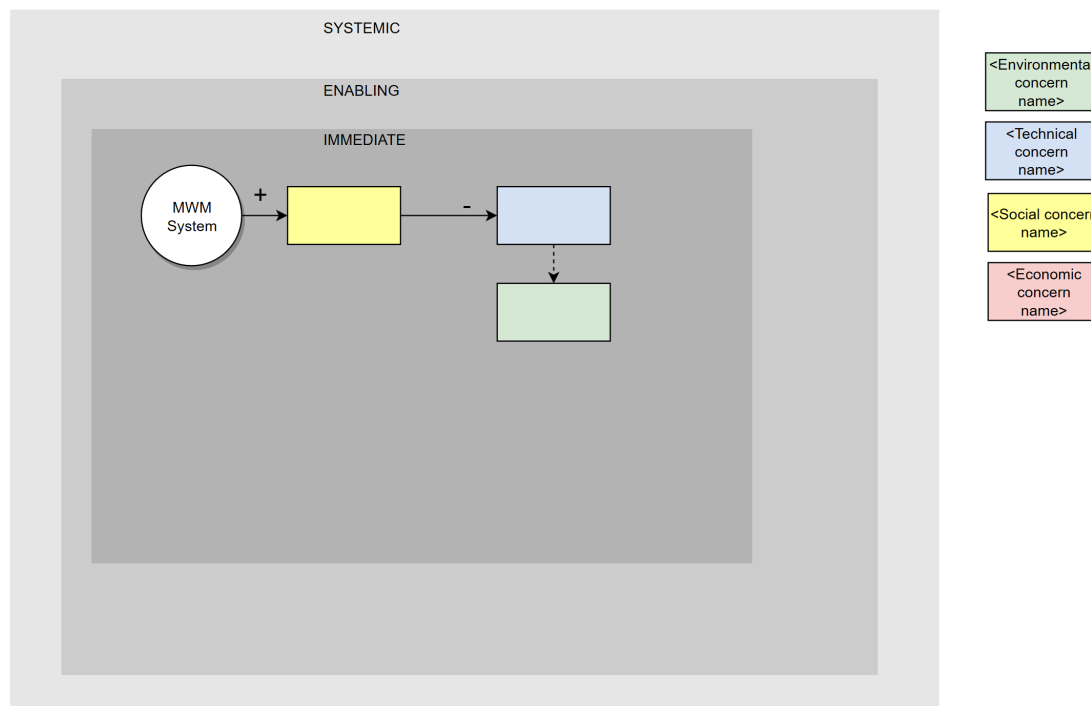


Figure A1. Decision Map template.

Table A1. Items of the questionnaire on perceived ease of use and usefulness.

ID	Items	SQ Model	DM	Perceptions
I1	I found the notation of the decision maps as clear and understandable		x	Ease of use
I2	I believe that the decision maps are well-suited to be understood by people not involved in the process		x	Usefulness
I3	I think the SQ model provides a good support for mapping the most important concerns	x		Usefulness
I4	I have the feeling that the time expended in creating the decision map was too long		x	Ease of use
I5	I think that the sustainability model and the decision maps help analyzing the sustainability concerns of the software architecture	x	x	Usefulness

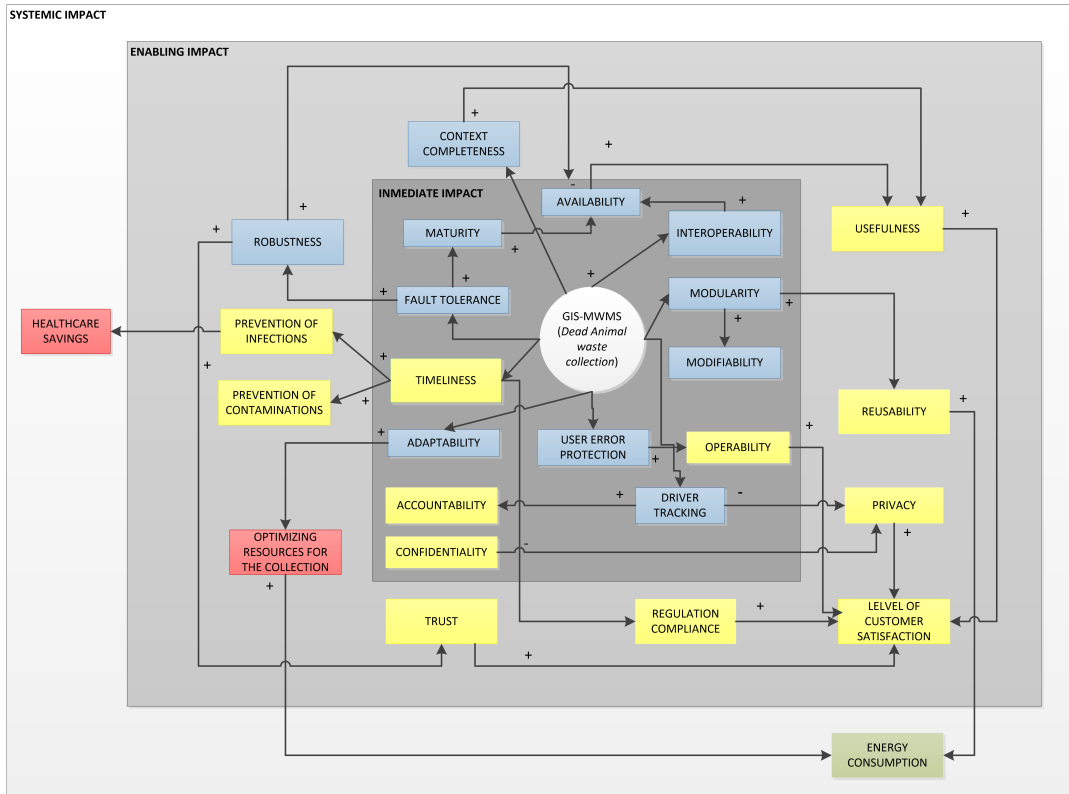


Figure A2. Reference Decision Map created by the researchers.

Appendix B

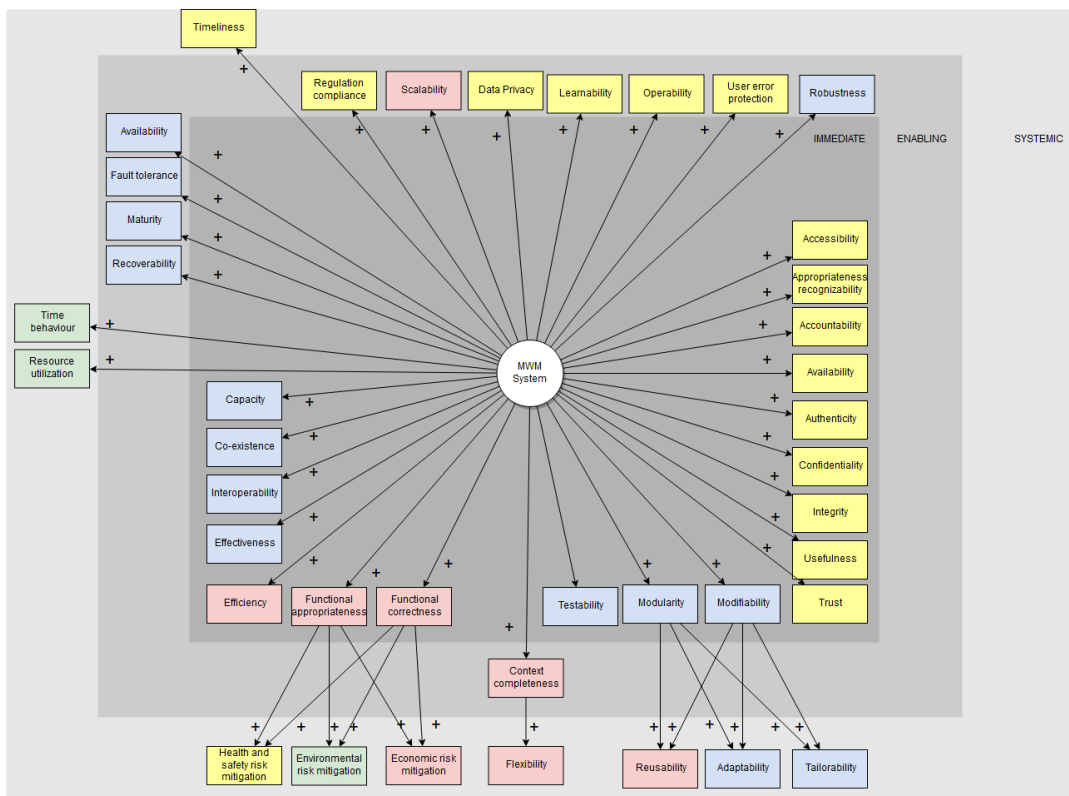


Figure A3. First version of Decision Map created by the practitioner.

## References

1. Avgeriou, P.; Stal, M.; Hilliard, R. Architecture Sustainability [Guest editors' introduction]. *IEEE Softw.* **2013**, *30*, 40–44. [[CrossRef](#)]
2. De-Arteaga, M.; Herlands, W.; Neill, D.B.; Dubrawski, A. Machine Learning for the Developing World. *ACM Trans. Manag. Inf. Syst.* **2018**, *9*, 9:1–9:14. [[CrossRef](#)]
3. Bates, O.; Thomas, V.; Remy, C.; Nathan, L.; Mann, S.; Friday, A. The future of HCI and sustainability: Championing environmental and social justice. In Proceedings of the CHI EA '18 Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; Volume 2018. [[CrossRef](#)]
4. Penzenstadler, B.; Duboc, L.; Venters, C.C.; Betz, S.; Seyff, N.; Wnuk, K.; Chitchyan, R.; Easterbrook, S.M.; Becker, C. Software Engineering for Sustainability: Find the Leverage Points! *IEEE Softw.* **2018**, *35*, 22–33. [[CrossRef](#)]
5. Chitchyan, R.; Becker, C.; Betz, S.; Duboc, L.; Penzenstadler, B.; Seyff, N.; Venters, C. Sustainability Design in Requirements Engineering: State of Practice. In Proceedings of the 38th International Conference on Software Engineering Companion, Austin, TX, USA, 14–22 May 2016; pp. 533–542. [[CrossRef](#)]
6. Moghaddam, F.A.; Lago, P.; Grosso, P. Energy-Efficient Networking Solutions in Cloud-Based Environments: A Systematic Literature Review. *ACM Comput. Surv.* **2015**, *47*, 64–96. [[CrossRef](#)]
7. Nidumolu, R.; Prahalad, C.; Rangaswami, M. Why sustainability is now the key driver of innovation. *IEEE Eng. Manag. Rev.* **2015**, *43*, 85–91. [[CrossRef](#)]
8. Calero, C.; Moraga, M.Á.; Bertoa, M.F. Towards a Software Product Sustainability Model. *arXiv* **2013**, arXiv:1309.1640.
9. Venters, C.; Lau, L.; Griffiths, M.; Holmes, V.; Ward, R.; Jay, C.; Dibsdaile, C.; Xu, J. The Blind Men and the Elephant: Towards an Empirical Evaluation Framework for Software Sustainability. *J. Open Res. Softw.* **2014**, *2*. [[CrossRef](#)]
10. Lago, P.; Koçak, S.A.; Crnkovic, I.; Penzenstadler, B. Framing sustainability as a property of software quality. *Commun. ACM* **2015**, *58*, 70–78. [[CrossRef](#)]
11. Condori-Fernandez, N.; Lago, P. Characterizing the Contribution of Quality Requirements to Software Sustainability. *J. Syst. Softw.* **2018**, *137*, 289–305. [[CrossRef](#)]
12. Penzenstadler, B. Infusing Green: Requirements Engineering for Green in and through Software Systems. In Proceedings of the Third International Workshop on Requirements Engineering for Sustainable Systems, RE4SuSy 2014, Co-Located with 22nd International Conference on Requirements Engineering, Karlskrona, Sweden, 26 August 2014; pp. 44–53.
13. Condori-Fernández, N.; Lago, P. The Influence of Green Strategies Design onto Quality Requirements Prioritization. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Utrecht, The Netherlands, 19–22 March 2018; pp. 189–205.
14. Chitchyan, R.; Noppen, J.; Groher, I. What Can Software Engineering Do for Sustainability: Case of Software Product Lines. In Proceedings of the International Workshop on Product Line Approaches in Software Engineering, Florence, Italy, 19 May 2015; pp. 11–14. [[CrossRef](#)]
15. Lago, P. Architecture Design Decision Maps for Software Sustainability. In Proceedings of the International Conference on Software Engineering, Software Engineering in Society Track (ICSE SEIS), Montreal, QC, Canada, 29–31 May 2019.
16. García-Mireles, G.A.; Moraga, M.Á.; García, F.; Calero, C.; Piattini, M. Interactions between environmental sustainability goals and software product quality: A mapping study. *Inf. Softw. Technol.* **2018**, *95*, 108–129. [[CrossRef](#)]
17. Bagnato, A.; Rocheteau, J. Towards green metrics integration in the MEASURE platform. In Proceedings of the 4th International Workshop on Measurement and Metrics for Green and Sustainable Software Systems Co-Located with Empirical Software Engineering International Week (ESEIW 2018), Oulu, Finland, 9 October 2018; p. 39.
18. Rodríguez, A.; Penzenstadler, B. An Assessment Technique for Sustainability: Applying the IMAGINE Approach to Software Systems. In Proceedings of the 2nd International Workshop on Requirements Engineering for Sustainable Systems, Rio, Brazil, 15 July 2013; Volume 995.

19. Condori-Fernandez, N.; Procaccianti, G.; Ali, N. Metrics for Green and Sustainable Software: MeGSuS 2014. In Proceedings of the 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, Rotterdam, The Netherlands, 6–8 October 2014; pp. 62–63. [\[CrossRef\]](#)
20. Condori-Fernández, N.; Lago, P.; Luaces, M.R.; Saavedra, A.; González, L. Using Participatory Technical-action-research to Validate a Software Sustainability Model. In Proceedings of the 6th International Conference on ICT for Sustainability (ICT4S), Lappeenranta, Finland, 10–14 June 2019.
21. Condori Fernandez, N.; Lago, P. *A Sustainability-Quality Model: (Version 1.0)*; VU Technical Report: Amsterdam, The Netherlands, 2018.
22. Oyediji, S.; Penzenstadler, B.; Adisa, M.O.; Wolff, A. Validation Study of a Framework for Sustainable Software System Design and Development. In Proceedings of the 6th International Conference on ICT for Sustainability (ICT4S), Lappeenranta, Finland, 10–14 June 2019.
23. Condori-Fernández, N.; Bagnato, A.; Kern, E. A focus group for operationalizing software sustainability with the MEASURE platform. In Proceedings of the 4th International Workshop on Measurement and Metrics for Green and Sustainable Software Systems (MeGSuS), Co-Located at ESEM, Oulu, Finland, 9 October 2018.
24. Wieringa, R.; Morali, A. Technical Action Research as a Validation Method in Information Systems Design Science. In *Design Science Research in Information Systems; Advances in Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 220–238.
25. MacDonald, C. Understanding Participatory Action Research: A Qualitative Research Methodology Option. *Can. J. Action Res.* **2012**, *13*, 34–50.
26. Brundtland, G.; Khalid, M.; Agnelli, S.; Al-Athel, S.; Chidzero, B.; Fadika, L.; Hauff, V.; Lang, I.; Shijun, M.; Morino de Botero, M.; et al. *Our Common Future (Brundtland Report)*; Technical Report; World Commission on Environment and Development: Cape Town, South Africa, 1987.
27. Venters, C.; Jay, C.; Lau, L.; Griffiths, M.K.; Holmes, V.; Ward, R.; Austin, J.; Dibsdale, C.E.; Xu, J. Software Sustainability: The Modern Tower of Babel. In Proceedings of the RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems, Karlskrona, Sweden, 26 August 2014.
28. Yan, M.; Chan, C.; Gygax, A.; Yan, J.; Campbell, L.; Nirmalathas, A.; Leckie, C. Modeling the Total Energy Consumption of Mobile Network Services and Applications. *Energies* **2019**, *12*, 184. [\[CrossRef\]](#)
29. Pihkola, H.; Hongisto, M.; Apilo, O.; Lasanen, M. Evaluating the Energy Consumption of Mobile Data Transfer—From Technology Development to Consumer Behaviour and Life Cycle Thinking. *Sustainability* **2018**, *10*, 2494. [\[CrossRef\]](#)
30. Hankel, A.; Oud, L.; Saan, M.; Lago, P. A Maturity Model for Green ICT: The Case of the SURF Green ICT Maturity Model. In Proceedings of the 28th International Conference on Informatics for Environmental Protection (EnviroInfo), Oldenburg, Germany, 10–12 September 2014; pp. 33–40.
31. Becker, C.; Betz, S.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Penzenstadler, B.; Seyff, N.; Venters, C. Requirements: The Key to Sustainability. *IEEE Softw.* **2016**, *33*, 56–65. [\[CrossRef\]](#)
32. Petersen, K.; Gencel, Ç.; Asghari, N.; Baca, D.; Betz, S. Action research as a model for industry-academia collaboration in the software engineering context. In Proceedings of the ACM International Workshop on Long-Term Industrial collaboration on Software Engineering (WISE), Vasteras, Sweden, 15–19 September 2014; pp. 55–62. [\[CrossRef\]](#)
33. Tolvanen, J.P. *Incremental Method Engineering with Modeling Tools*; Springer: Berlin/Heidelberg, Germany, 1998.
34. Kitchenham, B.; Pickard, L.; Pfleeger, S.L. Case studies for method and tool evaluation. *IEEE Softw.* **1995**, *12*, 52–62. [\[CrossRef\]](#)
35. Davis, F.D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.* **1989**, *13*, 319–340. [\[CrossRef\]](#)
36. Krueger, R. *Focus Groups: A Practical Guide for Applied Research*; Sage Publications: Thousand Oaks, CA, USA, 1994.
37. Wixom, B.H.; Todd, P.A. A Theoretical Integration of User Satisfaction and Technology Acceptance. *Inf. Syst. Res.* **2005**, *16*, 85–102. [\[CrossRef\]](#)
38. Assefa, G.; Frostell, B. Social sustainability and social acceptance in technology assessment: A case study of energy technologies. *Technol. Soc.* **2007**, *29*. [\[CrossRef\]](#)

39. Ormeno, Y.I.; Panach, J.I.; Condori-Fernandez, N.; Pastor, O. Towards a proposal to capture usability requirements through guidelines. In Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, 29–31 May 2013; pp. 1–12. [[CrossRef](#)]
40. Tang, A.; Antony, T.; Lau, M.F. Software architecture review by association. *J. Syst. Softw.* **2014**, *88*, 87–101. [[CrossRef](#)]
41. Razavian, M.; Tang, A.; Capilla, R.; Lago, P. In two minds: How reflections influence software design thinking. *J. Softw. Evol. Process* **2016**, *28*, 394–426. [[CrossRef](#)]
42. Becker, C.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Penzenstadler, B.; Seyff, N.; Venters, C.C. Sustainability Design and Software: The Karlskrona Manifesto. In Proceedings of the 37th International Conference on Software Engineering, Florence, Italy, 16–24 May 2015; Volume 2, pp. 467–476.
43. ISO/IEC. *ISO/IEC 25010—Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*; Technical Report; ISO: Geneva, Switzerland, 2010.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).