//UCIENCIA/23
CONVENCIÓN CIENTÍFICA INTERNACIONAL

Temática: X Taller Internacional de Gestión de Proyectos

Criteria for environmental feasibility analysis in software projects

Yadira García García [1*], Marieta Peña Abreu [2], Carlos Rafael Rodríguez Rodríguez [3], Pedro Arango Astorga [4]

[1] University of Informatics Sciences. Havana, Cuba. yadira.garcia9201@gmail.com
[2] University of Informatics Sciences. Havana, Cuba. mpabreu@uci.cu
3 University of Informatics Sciences. Havana, Cuba. crodriguezr@uci.cu
4 University of Informatics Sciences. Havana, Cuba. parango@uci.cu

[*] Autor para correspondencia: yadira.garcia9201@gmail.com

Abstract
The protection of the environment is a relevant factor in ensuring the availability of natural resources essential to achieve sustainable development. Information and Communication Technologies have the capacity to influence the reduction of global carbon emissions, which leads the industry to make important efforts in terms of technological innovation aimed at sustainability. For the development of a project to be sustainable, it must consider, in addition to the economic feasibility, and the social benefit, the reasonable use of natural resources, which entails an environmental feasibility analysis carried out. To contribute to the success of a project, in order to achieve sustainability, and contribute to environmental protection, it is imperative to include environmental criteria in the feasibility analysis. This research proposes a set of criteria to evaluate the environmental dimension in the feasibility analysis of software projects.

Keywords: green IT, environmental criteria, environmental feasibility, environmental sustainability, software sustainability

Introduction

Nowadays, due to the increase in environmental problems, the environment has gained special recognition and importance as a factor in guaranteeing progress. Therefore, environmental protection is a relevant factor in ensuring the availability of natural resources that are indispensable for sustainable development. The Information and Communication Technologies (ICT) have the capacity to influence the reduction of global carbon emissions, leading the industry to make significant efforts in technological innovation aimed at sustainability (Plasencia Soler, 2018). The incorporation of the concept of sustainability in software development implies a consideration of its impact on the environment throughout the stages of its life cycle (Carvalho & Jr, 2017) (García-Mireles et al., 2018) (Khalifeh et al., 2020).

The development of a sustainable software project must consider not only the economic feasibility and social benefit, but also the reasonable use of natural resources (García-Mireles et al., 2018), which entails carrying out environmental feasibility analyses. Environmental analysis has been more focused on investments (Carvalho & Jr, 2017) (Huemann & Silvius, 2017) than on software projects; however, in software project management, such analysis should be considered. In the particular case of software, its intangible character imposes additional challenges to establish its value.

The purpose of environmental feasibility analyses is to identify, predict, and interpret the environmental impact of a project, before its acceptance, modification or rejection. This process seeks to identify, quantify and assess the various impacts of a project on the environment and human health (Zulueta Véliz & Bello Pérez, 2016). In order to achieve sustainability and contribute to environmental protection, it is necessary to include environmental criteria in the feasibility analysis (Peña Abreu, 2017) (García et al., 2018). This research aims to propose a set of environmental criteria to be taken into account in the evaluation of the environmental dimension in the feasibility analysis of software projects.

Materials and methods o Computational methodology

Software and the environment

The negative impacts that may be generated by the computer systems to be developed, and the technology used for this purpose, must be identified and evaluated from the start. This activity should be oriented as far as possible to minimize or avoid the impacts caused by these technologies and to propose solutions from a global perspective. The following is a list of factors in software development that have an impact on the environment.

Electronic waste

Technological waste refers to any product, good or component that has an electronic device, which has reached the end of its useful life; it is associated with equipment used in the development of computer systems (computers, monitors, printers, photocopiers) (Martínez & Porcelli, 2015) (Mourão et al., 2018) (Khalifeh et al., 2020).

Energy requirements

The equipment used in software development requires electricity for its operation. The use of such devices leads to an increase in the production and consumption of electrical energy. With the growing demand for data centers to centralize Information Technology (IT) applications, energy consumption has shown an accelerated increase. All this technological production also generates large volumes of technological waste (Martínez & Porcelli, 2015) (GarcíaMireles et al., 2018) (Mourão et al., 2018) (Karita et al., 2019) (Khalifeh et al., 2020).

Carbon footprint

Carbon footprint is the total greenhouse gases emitted directly or indirectly by an individual, organization, event or product throughout its life cycle. Some studies (Shehabi et al., 2018) (Zou et al., 2022) have reported that the overall impacts of software systems, such as their energy consumption and hardware disposal, results in a larger carbon footprint (García-Mireles et al., 2018) (Saputri & Lee, 2020). The current consumption of greenhouse gases related to the ICT sector corresponds to approximately 4%, which could reach 8% by 2025. By 2040, it will represent 14% of the global carbon footprint.

Impact on health

Technological advances can cause diseases, traumas, and physical or psychological ailments such as visual stress, cervical tension, carpal tunnel syndrome, and median nerve syndrome. In addition, electronic devices carry radiation that causes headaches, fatigue, stress, vision problems, nervousness, irritability, insomnia, cardiovascular, and gastric disorders. Software developers are constantly exposed to each of these ailments (Martínez & Porcelli, 2015). Considering the elements mentioned above, an environmentally friendly technology is necessary to reduce most of the existing pollution. Within this framework, we have Green IT and Green Computing, which focus on the efficient use of computational resources, minimizing the environmental impact through the implementation of sustainable development policies.

Green IT

Green IT is the study and practice of the design, construction and use of hardware, software, and information technologies with a positive impact on the environment (Patón-Romero et al., 2021). The implementation in organizations of Green IT effectively contributes to sustainability, reducing the negative impact of ICT on the environment due to the energy consumption and emissions that it produces. Globally, there are new trends in green technology, which will be analyzed below.

Dematerialization of documents

Electronic documents have become an ally of the environment, it promotes the use of text messages, data and e-mails in daily work, contributing to the reduction of the use of physical paper. Making use of the digital signature in documents, assuring the authorship and integrity of the same. The systems that allow the dematerialization of documents are called ecological systems (Martínez & Porcelli, 2015) (Patón-Romero et al., 2021).

Paper optimization

The paper industry is among the largest generators of greenhouse gases. One way to achieve savings and optimization in the use of paper is to use recycled paper (Khalifeh et al., 2020) (Patón-Romero et al., 2021). In some countries (PatónRomero et al., 2021) the concept of zero paper office or paperless office is applied as a good practice, which is related to the replacement of paper documents by digital documents and certificates.

Teleworking

Teleworking is a method of working remotely that brings great benefits in terms of the environment. This modality has been developed as a work alternative, growing with the use of new remote technologies. Adopting this method can contribute to the reduction of the emission of greenhouse gases, since every time we travel we increase fuel consumption (Karia & Asaari, 2016).

Virtualization

Virtualization is the process of running multiple independent operating systems on the same machine. This technology allows the separation of hardware and software and, in turn, consists of the use of software to enable a physical resource to run on multiple isolated virtual machines (Bermejo et al., 2019) (Bermejo & Juiz, 2020).

Cloud Computing

Cloud computing is a service model for accessing, allocating, controlling and optimizing resources, understood as making resources available for the user's enjoyment in various service modalities. This technology reduces paper consumption and contributes to CO2 reduction, being one of the most sustainable and efficient ICT from an environmental point of view.

Grid Computing

Grid technologies allow computers to share not only information but also computing power and storage capacity over the Internet or other telecommunications networks. In other words, in the grid not only content is shared, but also processing capacity, applications, and even completely heterogeneous devices (sensors, networks, computers, etc.).

Data center optimization

When optimizing a data center, the analysis aims at how good the operation is in terms of direct power consumption by powering the servers, and the total energy spent on other aspects, such as cooling and lighting (Patón-Romero et al., 2021). A better use of technology includes the optimization of energy use, the use of less polluting materials, the substantial reduction of physical space and the optimization of resource management.

Serverless computing

A form of cloud computing in which a cloud provider dynamically manages the hosting of computational resources. Allows greater control over capacity and power consumption. Efficiently shares infrastructure resources by running functions on demand.

Acquisition of sustainable technologies

Use of sustainable technologies that comply with certain norms, standards (ISO 14001, ISO 779/9296) and acceptable levels of energy consumption (software, specific computer equipment such as sensors) (Patón-Romero et al., 2021).

Transportation decrease

Application of travel policies and design ofsoftware project processes in a way that minimizes travel. Actively promote alternatives such as the use of e-mail, cell phones, and video conferencing (Marnewick, 2017) (Patón-Romero et al., 2021), which are also used in teleworking.

Energy consumption

Takes energy consumption into account in the design of project processes, promote green energy, energy-saving equipment, energy efficiency practices, and their intelligent use. Use programming approaches and tools that allow profiling energy consumption dynamically (García-Mireles et al., 2018) (Mourão et al., 2018) (Khalifeh et al., 2020). Reducing energy consumption is one of the main goals in sustainable software development.

Optimization of resource consumption

Takes into account, for the development of a system, the type of resource to be consumed, as well as the way in which it will be managed. Define measures and metrics focused on the optimization of these resources (García-Mireles et al., 2018) (Mourão et al., 2018). Reduce, reuse, recycle, and consider the waste of resources in the design of the project processes (Martens & Carvalho, 2016) (Yuan, 2017) (Khalifeh et al., 2020).

Green subcontracting

When selecting materials, products and equipment, environmental aspects such as energy consumption, waste and the pollution they cause, and reuse capabilities, should be taken into account. Select suppliers based on their environmental policies, expertise, locations (to minimize transportation) and use of natural resources (Aarseth et al., 2016) (Carvalho & Jr, 2017) (Khalifeh et al., 2020).

Environmental criteria

Having analyzed the above elements, corresponding to the software and the environment, as well as the use of green computing in pursuit of sustainability, a set of environmental criteria to be considered by software development companies is presented. The criteria to be evaluated in the environmental feasibility analysis to carry out or not the execution of a software project must take into account the type of system to be developed, regarding its future contribution or detriment to the environment, as well as the needs and requirements of the clients.

In general, the environmental dimension should be evaluated in conjunction with the economic, technical, commercial, and social dimensions, which contributes greatly to the decision-making process in order to initiate a software project with a higher probability of success. Table 1 shows the proposed criteria.

Table 1. Conceptualization of the proposed criteria.

Criteria: Dematerialization and document optimization	**Classification:** Qualitative
Conceptualization: materialization of documents, using digital documents. Paper recycling. The greater the result, the better contributions to the environment the project will have and the better feasibility.	
Scale of values for evaluation: •Very high •High •Medium •Low •Not occurring	
Criteria: Use of telework as a work alternative	**Classification:** Qualitative

Conceptualization: use teleworking as a work alternative, which benefits the environment by contributing to the reduction of greenhouse gases.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Use of virtualization **Classification:** Qualitative

Conceptualization: the possible use of virtualization in the development of the system is analyzed. A high use of this technology contributes to minimize the use of physical resources such as servers or storage devices.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Use of Cloud Computing **Classification:** Qualitative

Conceptualization: the possible use of cloud computing for system development is analyzed. A high use of this green technology contributes to the allocation, control, and optimization of resources.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Use of Grid Computing **Classification:** Qualitative

Conceptualization: the possible use of grid computing for software development is discussed. A high use of this green technology contributes to the sharing of content, processing power, applications, and heterogeneous devices.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Use Serverless Computing **Classification:** Qualitative

Conceptualization: the possible use of serverless computing in the development of the system is analyzed. A high use of this technology allows greater control over capacity and power consumption.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Data center optimization **Classification:** Quantitative

Conceptualization: the optimization of energy use is measured in terms of direct consumption for server power supply, and in aspects such as air conditioning and lighting. Use of fewer polluting materials, substantial reduction of physical space, and optimization of resource management.

Scale of values for evaluation: [1,5]

Criteria: Optimization of resource consumption **Classification:** Quantitative

Conceptualization: resource consumption and management analysis. Reduce, reuse, recycle, and consider resource waste for optimization in the design of project processes.

Scale of values for evaluation: [1,5]

Criteria: Energy efficient practices **Classification:** Qualitative

Conceptualization: promote efficient energy consumption through the use of green energy, energy saving equipment, energy efficiency practices, and their intelligent use in the design of project processes. The better the practices, the better the results of the project.

Scale of values for evaluation: •Very high •High •Medium •Low •Not used

---

Criteria: Product carbon footprint — **Classification:** Quantitative

Conceptualization: estimation of the total greenhouse gases that may be emitted by direct or indirect effect in the development of the project's processes.

Scale of values for evaluation: • Numerical value calculated according to:

carbon footprint (CF)= activity data x emissions factor

Activity data: parameter that defines the degree or level of activity generating of greenhouse gases emissions.

Emission factor: amount of greenhouse gases emitted for each unit of the "activity data" parameter.

For each result, a given amount of carbon dioxide equivalent (CO_2 eq). The carbon footprint of the product is the total of the partially calculated emissions from the activities associated with its development. The lower the CF result, the better the results and feasibility of the project.

Criteria: Impact on the developers' health — **Classification:** Qualitative

Criteria: Software compatibility with different power-constrained hardware designs — **Classification:** Qualitative

Conceptualization: analysis of the compatibility of the technologies to be used for software development on different hardware. As well as the compatibility of the future software to be developed on this type of hardware.

Scale of values for evaluation: •Very high compatibility •High compatibility •Medium compatibility •Low compatibility •Doesn't exist

Criteria: Acquisition of sustainable technologies — **Classification:** Qualitative

Conceptualization: analysis of the acquisition of sustainable technologies that comply with norms, standards, and acceptable levels of energy consumption.

Scale of values for evaluation: •Very high acquisition •High acquisition •Medium acquisition •Low acquisition •No decrease • Does not occur

Criteria: Green subcontracting — **Classification:** Qualitative

Conceptualization: analysis of environmental aspects to be taken into account when selecting materials, products, and equipment. Select suppliers based on their environmental policies, knowledge, locations, and use of natural resources.

Scale of values for evaluation: •Very high •High •Medium •Low •Not occurring

Companies today can make the software an integral part of their efforts to achieve sustainability. Taking into account elements such as carbon footprint in the way the system is designed, developed, and implemented. Representing, for example, aspects of how data centers that provide cloud services operate. The software itself does not consume energy or emit harmful discharges. The problem lies in how it is developed for use, and subsequently how it is used. Creating green software develops a higher quality product that is leaner, cleaner, and fit for purpose. Today's software developers are increasingly considering the approach of a sustainable organization, having a commitment to green software being a compelling attraction to customers.

Results and discussion

Computation with words

Computation with words (CWW) is a technique based on fuzzy logic where words are used instead of numbers (Zadeh, 1996). The 2-tuple linguistic model is used to carry out CWW processes, where the use of linguistic information makes decisional models more reliable under uncertain environments.

To carry out the environmental feasibility analysis of software projects, the use of the 2-tuple linguistic model is proposed. This choice is given because the method allows assessments in different expression domains, assimilates uncertainty environments and there is no loss of information during its calculation.

Method for the analysis of environmental feasibility of software projects

For the instrumentation of the model, a series of steps are followed:

Step 1: Identify the projects to be evaluated by experts in project management and software development.
Step 2: Select experts to participate in the analysis process.
Step 3: Define the environmental criteria to be evaluated by the experts.
Step 4: Determine the weight, if applicable, of each of the criteria to be evaluated.
Step 5: Compile the evaluations of the experts of each criterion associated with the project.
Step 6: Add expert ratings.
Step 7: Interpret the results.

Each of these steps is briefly described below:

Step 1: Identify the projects to be evaluated by experts in project management and software development

A set of projects must be identified $P = \{p_j \mid j \in (1, \ldots, n)\}$ which constitute the input to the analysis process.

Step 2: Select experts to participate in the analysis process

A group of experts intervenes in the evaluation of these criteria $E = \{e_i \mid i \in (1, \ldots, m)\}$ who analyze the projects.

Step 3: Define the environmental criteria to be evaluated by the experts

The evaluation of the projects is carried out based on a set of environmental criteria identified $C = \{c_k \mid k \in (1, \dots, p)\}$.

Step 4: Determine the weight, if applicable, of each of the criteria to be evaluated

These criteria, as appropriate, will have different weights assigned according to their importance in the analysis of the projects, for which a weight vector is defined. $W^c = (W^{c_1}, \dots, W^{c_p})$.

Step 5: Compile the evaluations of the experts of each criterion associated with the project.

To express the preferences of the experts, the utility vector is used $X = \left(x_j^{ki}, \dots, x_j^{ki}\right)$, where x_j^{ki} represents expert preference e_i about the project e_j according to criteria c_k. Experts will be able to issue their preferences through linguistic values (S): $x_j^{ki} = s_j^{ki} \in S = \{S_0, \dots, S_g\}$ being g+1 the cardinality of the Linguistic Term Set (CTL) S, that is, the number of terms in S. Each linguistic term does have an associated membership function [0; 1]. As CTL is proposed S= {Very Low, Low, Medium, High, Very High}.

Compilation of expert assessments

Experts provide their assessments through preference vectors: $X = \left(x_j^{ki}, \dots, x_j^{ki}\right)$ which can be collected as shown:

Table 2. Preferences of the experts

Proyects	Criteria	Experts		
		e_1	…	e_m
p_1	c_1	x_1^{11}	…	x_1^{1m}
	…	…	…	…
	c_k	x_1^{k1}	…	x_1^{km}
p_2	c_1	x_2^{11}	…	x_2^{1m}
	…	…	…	…
	c_k	x_2^{k1}	…	x_2^{km}
p_n	c_1	x_n^{11}	…	x_n^{1m}
	…	…	…	…
	c_k	x_n^{k1}	…	x_n^{km}

Project analysis

Transformation of the input into a fuzzy set

The fuzzy set that represents a linguistic term si will be 0 in all except in the value corresponding to the ordinal, i, of the linguistic label that will be one. For example, for the label Very High, in the CTL, the fuzzy set that represents it is (0, 0, 0, 0, 1).

Transformation of fuzzy sets into 2-tuples

The representation model based on 2-tuples is based on the concept of symbolic translation. The symbolic translation of a linguistic term is a valued number in the interval [-0.5, 0.5) which represents the "information difference" between a quantity of information expressed by the value β ∈ [0, g] obtained in a symbolic operation and the nearest integer value i ∈ {0,…, g} indicating the index of the linguistic label (S_i) closer in S. Starting from this concept (Martínez, 1999) develops a representation model for linguistic information that uses as a base the representation 2-toucas, (S_a, α_a), S_a ∈ S and α_a ∈ [-0.5,5,0.5), -where: S_a represents the linguistic label, and αa is a number that expresses the value of the distance from the original result to the index of the closest linguistic label in the linguistic label, its, symbolic translation.

Taking these concepts into account, the following function is used to transform the fuzzy sets already obtained into linguistic 2-tuples (Herrera et al., 2005).

$$x\left(F(S_t)\right) = x\left(\left\{\left(S_j, \gamma_j\right), j = 0, \dots, g\right\}\right) = \frac{\sum_{j=0}^{g} j\gamma_j}{\sum_{j=0}^{g} \gamma_j} = \Delta\beta = (s_i, \alpha)$$

Table 3. Expert evaluations expressed in 2-tuples

Proyects	Criteria	Experts		
		e_1	…	e_m
p_1	c_1	$(s_a, \alpha_a)_1^{11}$	…	$(s_a, \alpha_a)_1^{1m}$
	…	…	…	…
	c_k	$(s_a, \alpha_a)_1^{k1}$	…	$(s_a, \alpha_a)_1^{km}$
p_2	c_1	$(s_a, \alpha_a)_2^{11}$	…	$(s_a, \alpha_a)_2^{1m}$
	…	…	…	…
	c_k	$(s_a, \alpha_a)_2^{k1}$	…	$(s_a, \alpha_a)_2^{km}$
p_n	c_1	$(s_a, \alpha_a)_n^{11}$	…	$(s_a, \alpha_a)_n^{1m}$
	…	…	…	…
	c_k	$(s_a, \alpha_a)_n^{k1}$	…	$(s_a, \alpha_a)_n^{km}$

Step 6: Add experts ratings

Aggregation

The linguistic information representation model is supported by a computational model based on the functions Δ y Δ^{-1}, that transform numeric values into 2-tuples and vice versa without loss of information, so traditional numeric aggregation operators can be easily extended to 2-tuples in order to obtain accurate results and provide a representation that facilitates their interpretation. These operators will be used to obtain the collective value of each criterion for each project from the aggregation of the evaluations of all the experts the collective value of each criterion for each project from the aggregation of the evaluations of all the experts in 2-tuples and to obtain the feasibility of each project from the aggregation of the evaluations of all its criteria.

Aggregation operators

Extended Arithmetic Mean

This operator allows to determine the balance point or center of the set of values. For a set of 2-tuples $x = \{(s_1, \alpha_1), \dots, (s_n, \alpha_n)\}$, The extension of this operator is obtained as follows:

$$x^{-e}(x) = \Delta\left(\frac{1}{n}\sum_{i=1}^{n}\Delta^{-1}\big((s_i, \alpha_i)\big)\right) = \Delta\left(\frac{1}{n}\sum_{i=1}^{n}\beta_i\right)$$

OWA (Ordered Weighted Averaging)

This is a weighted aggregation operator, in which the weights are not associated with a predetermined value but with a determined position. So if we have a set of 2-tuples $x = \{(s_1, \alpha_1), \dots, (s_n, \alpha_n)\}$ y $W = (w_1, \dots, w_n)$ is its associated weight vector such that $w_i \in [0,1]$ and $\sum w_i = 1$, the operator extension is obtained as shown:

$$OWA(x) = \Delta\left(\sum_{i=1}^{n} w_i * \beta_i\right)$$

where β_i is he i-ésimo highest value of $\Delta^{-1}(s_i, \alpha_i)$.

Calculate the collective value of each criterion for each project:

As it is intended to handle multi-expert evaluations, it is necessary to determine the collective value of the criteria for each project. It is assumed that all experts have the same weight in the evaluation. This collective value can be obtained as follows:

$$(s_b, \alpha_b)_j^k = \bar{x}^e\left((s_a, \alpha_a)_j^{k1}, \dots, (s_a, \alpha_a)_j^{km}\right)$$

Table 4. Collective values of the criteria for each project

Proyectos	Criterios	Expertos			Valores colectivos x criterios
		e_1	...	e_m	
p_1	c_1	$(s_a,\alpha_a)_1^{11}$	...	$(s_a,\alpha_a)_1^{1m}$	$(s_b,\alpha_b)_1^1$
	...	...	...	...	...
	c_k	$(s_a,\alpha_a)_1^{k1}$	...	$(s_a,\alpha_a)_1^{km}$	$(s_b,\alpha_b)_1^k$
p_2	c_1	$(s_a,\alpha_a)_2^{11}$	...	$(s_a,\alpha_a)_2^{1m}$	$(s_b,\alpha_b)_2^1$
	...	...	...	...	...
	c_k	$(s_a,\alpha_a)_2^{k1}$	...	$(s_a,\alpha_a)_2^{km}$	$(s_b,\alpha_b)_2^k$
p_n	c_1	$(s_a,\alpha_a)_n^{11}$	...	$(s_a,\alpha_a)_n^{1m}$	$(s_b,\alpha_b)_n^1$
	...	...	...	...	...
	c_k	$(s_a,\alpha_a)_n^{k1}$	...	$(s_a,\alpha_a)_n^{km}$	$(s_b,\alpha_b)_n^k$

Calculate social feasibility of each project

Once the collective value of the criteria for each project has been obtained, the environmental feasibility of each project is determined. Taking into account the weight of each criterion (w^{c_k}), this feasibility can be obtained as shown:

$$(s_c,\alpha_c)_j = \bar{x}^e\left((s_b,\alpha_b)_j^1,\ldots,(s_b,\alpha_b)_j^n\right)$$

Table 5. Collective values of the projects

Proyects	Collective values of the projects
p_1	$(s_c,\alpha_c)_1$
p_2	$(s_c,\alpha_c)_2$
...	...
p_n	$(s_c,\alpha_c)_n$

Step 7: Interpret the results

Once the values of the collective evaluations of each of the criteria and of each project are obtained, they must be ordered in order to obtain an adequate interpretation of the results, so that specialists can easily determine which project has the greatest environmental feasibility. To perform this ordering it is necessary to use comparison operators for 2-tuples such as those presented in (Herrera et.al., 2000). These operators allow obtaining ordered sets as part of the solution of the problem.

2-tuple comparison criterion

For the 2-tuples (s_k,α_1) and (s_l,α_2) representing two valuations:

- If $k > l$ then $(s_k, \alpha_1) > (s_l, \alpha_2)$
- If $k < l$ then $(s_k, \alpha_1) < (s_l, \alpha_2)$
- If $k = l$ then:
  - If $\alpha_1 = \alpha_2$ then $(s_k, \alpha_1) = (s_l, \alpha_2)$
  - If $\alpha_1 < \alpha_2$ then $(s_k, \alpha_1) < (s_l, \alpha_2)$
  - If $\alpha_1 > \alpha_2$ then $(s_k, \alpha_1) > (s_l, \alpha_2)$

As a result of the comparison, a list of the projects ordered according to the evaluation of their environmental feasibility should be obtained, after having applied the 2-tuple linguistic model.

Conclusions

This article presents as a proposal a set of environmental criteria to be taken into account in the feasibility analysis of software projects before starting their development. The environmental feasibility analysis in this area is a novel field of research, which constitutes a source of incentives to improve the efficiency and quality of software and contribute to the economic growth of a country. By evaluating the environmental dimension within the feasibility, it contributes to the further development of a more sustainable, environmentally friendly software. Green IT is one of the most relevant fields in terms of sustainability, and one of the main assets with the greatest impact for organizations today.

Regarding environmental issues, the notable contributions of the current literature are more associated with the incorporation of sustainability in software projects, focusing mainly on the product of the project or the process throughout the project, not being so concerned with the feasibility analysis before starting a project. Therefore, the proposed criteria cover this stage. Software companies can incorporate the evaluation of these environmental criteria in the feasibility analyses, in addition to the traditional economic, technical, commercial, and social criteria, paying attention to the incorporation of sustainability in project management.

Finally, the results of this study will add a contribution to the existing knowledge related to software project feasibility analyses.

References

Aarseth, W., Ahola, T., Aaltonen, K., Økland, A., & Andersen, B. (2016). Project sustainability strategies: A systematic literature review. *International Journal of Project Management, 35*. https://doi.org/10.1016/j.ijproman.2016.11.006

Bermejo, B., & Juiz, C. (2020). Virtual machine consolidation: A systematic review of its overhead influencing factors. *The Journal of Supercomputing*, *76*(1), 324–361. https://doi.org/10.1007/s11227- 019-03025-y

Bermejo, B., Juiz, C., & Guerrero, C. (2019). Virtualization and consolidation: A systematic review of the past 10 years of research on energy and performance. *The Journal of Supercomputing*, *75*(2), 808– 836. https://doi.org/10.1007/s11227-018-2613-1

Carvalho, M., & Jr, R. (2017). Can project sustainability management impact project success? An empirical study applying a contingent approach. *International Journal of Project Management*, *35*. https://doi.org/10.1016/j.ijproman.2017.02.018

García, Y. G., Abreu, M. P., Várcacel, Y. P., & Rodríguez, C. R. R. (2018). Factibilidad social de proyectos de software en la toma de decisiones mediante la computación con palabras. *Serie Científica de la Universidad de las Ciencias Informáticas*, 11(3), 56–70.

García-Mireles, G. A., Moraga, M. Á., García, F., Calero, C., & Piattini, M. (2018). Interactions between environmental sustainability goals and software product quality: A mapping study. *Information and Software Technology*, 95, 108–129. https://doi.org/10.1016/j.infsof.2017.10.002

Herrera Francisco, M.L., & Sánchez P. (2005), Managing non-homogeneous information in group decision making. European Journal of Operational Research. 116: p. 115-132

Herrera, F. y Herrera-Viedma, E. (2000), Linguistic decision analysis: Steps for solving decision problems under linguistic information: Fuzzy Sets and Systems

Huemann, M., & Silvius, G. (2017). Projects to create the future: Managing projects meets sustainable development. *International Journal of Project Management,* 35(6), 1066–1070. https://doi.org/10.1016/j.ijproman.2017.04.014

Karia, N., & Asaari, M. (2016). Innovation Capability: The Impact of Teleworking on Sustainable Competitive Advantage. *International Journal of Technology, Policy and Management*, 16. https://doi.org/10.1504/IJTPM.2016.076318

Karita, L., Mourão, B., & Machado, I. (2019). Software industry awareness on green and sustainable software engineering: A state-of-the-practice survey. 501–510. https://doi.org/10.1145/3350768.3350770

Khalifeh, A., Farrell, P., Alrousan, M., Alwardat, S., & Faisal, M. (2020). Incorporating sustainability into software projects: A conceptual framework. *International Journal of Managing Projects in Business*, 13(6), 1339–1361. https://doi.org/10.1108/IJMPB-12-2019-0289

Marnewick, C. (2017). Information system project's sustainability capabality levels. *International Journal of Project Management*, 35. https://doi.org/10.1016/j.ijproman.2017.02.014

Martens, M., & Carvalho, M. (2016). Key factors of sustainability in project management context: A survey exploring the project managers' perspective. *International Journal of Project Management*, 35. https://doi.org/10.1016/j.ijproman.2016.04.004

Martínez, A. N., & Porcelli, A. M. (2015). Implicancias de las tecnologías informáticas en el ambiente y nuevas tendencias en el desarrollo de la informática verde como aporte al desarrollo sustentable. *Actualidad Jurídica Ambiental*, 50 (Octubre), 8–36.

Martínez, L. (1999), Un nuevo modelo de representación de información lingüística basado en 2-tuplas para la agregación de preferencias lingüísticas. Universidad de Granada: España.

Mourão, B. C., Karita, L., & Do Carmo Machado, I. (2018). Green and Sustainable Software Engineering—A Systematic Mapping Study. *Proceedings of the XVII Brazilian Symposium on Software Quality*, 121–130. https://doi.org/10.1145/3275245.3275258

Patón-Romero, J. D., Baldassarre, M. T., Rodríguez, M., Runeson, P., Höst, M., & Piattini, M. (2021). Governance and Management of Green IT: A Multi-Case Study. *Information and Software Technology*, 129, 106414. https://doi.org/10.1016/j.infsof.2020.106414

Peña Abreu, M. (2017). Modelo para el análisis de factibilidad de proyectos de software en entornos de incertidumbre. https://repositorio.uci.cu/jspui/handle/123456789/7938

Plasencia Soler, J. (2018). Modelo para contribuir a la sostenibilidad de entidades de las tecnologías de la información y las comunicaciones desde la gestión estratégica. https://doi.org/10.13140/RG.2.2.34361.83044

Saputri, T. R. D., & Lee, S. (2020). Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition. *Journal of Software: Evolution and Process*, 32(8). https://doi.org/10.1002/smr.2254

Shehabi, A., Smith, S., Masanet, E., & Koomey, J. (2018). Data center growth in the United States: Decoupling the demand for services from electricity use. *Environmental Research Letters*, 13. https://doi.org/10.1088/1748-9326/aaec9c

Yuan, H. (2017). Achieving Sustainability in Railway Projects: Major Stakeholder Concerns. *Project Management Journal*, 48, 115–132. https://doi.org/10.1177/875697281704800508

Zadeh, L. A. (1996), Nacimiento y evolución de la lógica borrosa, el soft computing y la computación con palabras. Vol. 8.

Zou, Y., Huang, M., Xiang, W., Lu, L., Lu, Y., Gao, J., & Cheng, Y. (2022). The impact of high-tech industry development on energy efficiency and its influencing mechanisms. *Frontiers in Environmental Science*, 10. https://www.frontiersin.org/articles/10.3389/fenvs.2022.962627

Zulueta Véliz, Y., & Bello Pérez, R. E. (2016). Modelos de Evaluación de la Importancia del Impacto Ambiental en contextos complejos bajo incertidumbre. https://repositorio.uci.cu/jspui/handle/ident/9065