Towards the Evaluation of Software Products from an Environmental Sustainability Perspective

Rui Dias Carvalho
Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal *ruimigcar2017@gmail.com*

Alberto Rodrigues da Silva
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal *alberto.silva@tecnico.ulisboa.pt*

Abstract

Sustainable development implies resource management that simultaneously guarantees the satisfaction of the present and future generations, considering the social, economic, and environmental dimensions. This paper proposes an approach to quantitatively assess software products' sustainability quality based on a library of requirements (i.e., general goals) considered as criteria in a multicriteria evaluation and analysis. To increase the environmental sustainability of software products, we argue that it is fundamental to comparatively evaluate them, identify the ones most in need of change, and quickly adapt existing products effectively and efficiently.

Keywords: Software Sustainability, Requirements Specification, Multicriteria Evaluation

1. Introduction

There is a growing concern regarding the environmental conservation throughout recent decades [5], [16], [26,27,28]. This concern goes hand in hand with the awareness that ecosystem services are fundamental and climate change and pollution significantly negatively impact those services [22], [29]. Pollution can have different forms, such as air pollution or acidification of the seas etc., many of them highly pervasive for any animal [23]. As a consequence of this reality, sustainable development was defined by the UN in 1987 as the development that meets the needs of the present without compromising the ability of future generations to meet their own needs [27].

This work acknowledges the three dimensions of sustainability, i.e., economic, social, and environmental, but its focus is on the environmental sustainability of software products (SPs). Like other products, SPs can contribute to pollution, e.g., due to high levels of energy consumption and disposal of electronic products at the end of life [6]. Considering these impacts, this paper proposes a multicriteria environmental sustainability evaluation of SPs from a GREEN in IT perspective, based on a library of general requirements (i.e., goals) and an evaluation indicator. This proposal contributes to the discussion of sustainability in information systems, possibly allowing a simple first understanding of resource usage by SPs and its impact.

"GREEN in IT" means that it shall be possible to make SPs more environmentally sustainable, and "GREEN through IT" means that it might be possible to have a more environmentally sustainable production and pollution mitigation or a greener world with IT products [18].

The impacts of IT products, e.g., allowing more intelligent and effective monitoring and mitigation of the environmental effects, are difficult to evaluate [8]. Regardless of this, software firms may consider the environmental impact of SPs more often if the auditing activity has a low cost and a well-defined process. This perspective of GREEN in IT is better accomplished with the tools we discuss because it might be possible to audit, design and develop SPs according to specific environmental sustainability qualities.

This paper is structured in 6 Sections. Section 2 introduces the sustainability dimensions of SP. Section 3 presents a library of requirements defined as general goals for environmental sustainability. Section 4 presents the multicriteria evaluation model for the evaluation of SPs. Section 5 supports the application of our model and the discussion with a simple but effective example. Finally, Section 6 presents the main conclusion.

2. Sustainability of Software Products

Sustainable software is an SP that is energy-efficient, has a positive impact on socio-economic activity, and minimizes the environmental impact of the process it supports [30]. The impact of sustainable software can be direct, through consumed resources, and indirect [10], through the activity of services and products produced with the support of the IT industry. This definition gathers the five dimensions of sustainability, as admitted by Venters, C. et al. [30], Becker et al. [2] and Lago et al. [14]: environmental, individual, economic, social and technical. We adopt this classification with five dimensions. From another perspective, Zaidan et al. [34] discussed six concerns that must be considered for the sustainability of software products: usability, functionality and features, security, developer support, customizability, and ease of installation.

The focus of this paper is the environmental sustainability of SPs. We try to contribute to increasing SPs' environmental sustainability without compromising other sustainability dimensions. Moreover, it is fundamental to acknowledge that IT is closely linked to industrial products with cyber-physical and IoT-based software. Also, digital twins is a technology that can be used to improve the environmental sustainability of industrial products by simulating, measuring and computing information obtained in real-time that can be used to generate optimization and design solutions [4]. For example, content management systems [1] is a class of software frameworks that simplifies and increases the production of SPs, especially of websites. However, their usage can only be considered environmentally friendly if it respects what is defined as being acceptable software practices, e.g., as defined by the researchers accepting the Karlskrona Manifesto [2].

To evaluate the environmental friendliness of SPs some authors consider the usage of conceptual frameworks like "Green Requirements Engineering" as fundamental [18]. In contrast, others think energy is the most crucial indicator of environmental sustainability of SPs, e.g., for websites or mobile apps [9] [32]. Penzenstadler et al. [18] and Kern et al. [13] assume that to define new requirements for environmental sustainability; we should start by looking for the ones that were defined in the context of other SPs sustainability dimensions, e.g., usability, security, and adapt them. Kern et al. [13] mention explicitly that SPs can be considered "relatively sustainable" when compared among them.

3. Environmental Sustainability Requirements for Software Products

Requirements Engineering (RE) intends to specify rigorous requirements in the early phase of software engineering, where the exact scope of the system is determined, and the stakeholders' needs and concerns are iteratively elaborated [18]. For example, requirements can be used to automate acceptance tests to verify if stakeholders' needs and concerns are satisfied [17], [36].

In this paper, we define how to audit and evaluate SPs regarding their environmental sustainability. The scope of our research is a class of requirements for environmental sustainability or green requirements that can be labelled as an example of Green Requirements Engineering, in which there is a focus on the environmental impacts of the system in consideration [18].

This research identifies high-level requirements, defined as goals, classified by general categories. To each goal, a number is attached to an environmental impact that might be negative or positive if they are considered a problem or a solution, respectively. Examples of categories for environmental sustainability of SP are the following: (C1) Color; (C2) Video; (C3) Elements; (C4) Energy; (C5) Aesthetics; (C6) Hardware; and (C7) SoftProduct.

All these requirements, if not satisfied, are considered problems because they may imply a higher level of electric energy consumption or more work to build the product. These were the chosen requirements because they help us understand the impact of the SPs elements that are usually responsible for an important volume of energy consumption [9] and complexity [15]. According to Maguel et al. [15], SP complexity is the measure of how difficult the program is to comprehend and work with. In this paper, we consider the number of elements an adequate measure of it.

These general categories can be applicable to every SP, being C1 to C5 and C7 straightforward examples. C6 assumes that any non-SP is designed and/or produced with the help of an SP that makes its environmental sustainability goals possible. Examples of such goals (and respective categories, indicated in square brackets) are presented as follows:

Goal 1. SP shall use lighter colours [Color]. SP shall use no lighter colours (lighter than a certain level) because darker colours mean lower electric energy consumption [2; 9]. Explains how to audit the lighter colours of a SP. It indicates the number of colours to change.

Goal 2. SP shall use static images instead of videos [Video]. SP shall use images instead of videos because moving images imply more electricity consumption [2; 9]. Indicates the number of videos to replace by static images.

Goal 3. SP shall use a minimum value of energy consumption [Energy]. SP shall use a maximum level of 18 Watts electric energy power because we use a similar value of mobile phones as a benchmark. It indicates areas in need of energy-saving.

Goal 4. SP shall have pleasant aesthetics [Aesthetics]. SP shall have a comparison between colours that does not imply a change in aesthetics, i.e., the contrast between colours allows easy reading.

Goal 5. Hardware shall meet its environmental purposes [Hardware]. Hardware shall be able to provide the desired output respecting its environmental purposes.

Goal 6. SP shall meet its environmental purposes [SoftProduct]. SP shall be able to provide the desired output respecting its environmental purposes [2].

Goal 7. SP shall have user-centric goals [SGoals]. SP shall have clearly defined goals for the end-user. Indicates the number of clearly defined user-centric goals.

Goal 8. SP shall be kept simpler as possible [Elements]. SP shall have a minimum of elements because the lowest number implies the minimum complexity. It indicates the number of elements, i.e., the system components seen by the end-user, e.g., text, static images and videos [31]. The SP with a higher number of elements is a smell that is the most complex, eventually needing simplification. The number of colours plus videos plus static images plus texts define the value of *Elements*.

The simplicity of this approach compares well with other proposals [21] and makes possible future modelling of goals easier with tools such as Pistar [19]. This set of goals can be used to audit a SP regarding its environmental impact and shade the light on new tools that can help to rebuild SPs to make them more environmentally sustainable and, at the same time maintain all the other requirements that make its usage sustainable, e.g., by combining usability and security requirements.

4. Multicriteria Evaluation for Environmental Sustainability of Software Products

Multicriteria evaluation or multicriteria analysis (MCA) is a technique used in several fields of science, such as computer science [3], [11], civil engineering [35], public policy [25], and environmental planning [29]. MCA can complement or be an alternative to cost-benefit analysis, being relevant at two of the stages of Impact Assessment, namely [25]: assessing the economic, social and environmental dimensions and comparing policy options. This analysis uses the following concepts: (i) objectives, i.e., the indication of the direction of change desired, (ii) evaluation criterion, i.e., the basis for evaluation in relation to a given objective, (iii) goal is a synonymous with a target, and (iv) attribute, i.e., a measure that indicates if goals have been met or not. There are various multicriteria analysis' methods, but the MCA method can be defined as an aggregate of all objectives, criteria (or attributes) and criterion scores [8].

The main objective of this evaluation is to minimize energy consumption and SP complexity. In this paper, the goals referred to in the previous section are used as attributes, and the concept objective is labelled as direction to avoid confusion with the mentioned goals. Table 1 shows the proposed model used to compare and evaluate SPs, with some concrete values (as an example), where:

- WP, the Weighted Performance, is equal to W * D * P;
- W the weight (between 0 and 1, the higher the value the higher the importance);
- D the direction (-1 if negative and 1 if positive); and
- P the performance (number of elements for the corresponding Goal).

Table 1. Multicriteria evaluation model for environmental sustainability of software products.

	Input Matrix			Software products SPX	
Goal Category	**Goal**	**Weight (W)**	**Direction (D)**	**Performance (P)**	**Weighted Performance (WP)**
C1. Color	G1	0.4	**-1**	3	-1.2
C2. Video	G2	0.4	**-1**	3	-1.2
C8. Elements	G8	0.2	-1	4	-0.8
C7. SGoals	*G7*	D	**1**	2	
				TOTAL	-1.5

For the values shown in Table 1, we may have the evaluation indicator of environmental sustainability defined by the following formula:

$$ESustainability = \frac{Number\ of\ Colours * 0.4 + Number\ of\ Videos * 0.4 + Number\ of\ Elements * 0.2}{Number\ of\ SP\ Goals}$$

The indicator uses the number of SP Goals dividing the total value of WP to determine the average environmental impact per sustainability goal. The values for the parameters *NumberofColours*, *NumberofVideos* and *NumberofElements* are the values for criteria *Colours*, *Videos* and *Elements*, respectively. The weight of 0.2 for the total number of elements is 50 % of the weight for the other two criteria, meaning that it has 50 % importance. It contains text and buttons but other elements already computed, such as images and videos.

The considered goals are relevant both to desktop and web-based SP. The scores used consider that images and colours, each with a 40 % score, are the main sources of energy consumption in standalone software [9][21]. Text elements, with a 20 % score, are less impactful for energy consumption but increase the complexity of an SP, i.e., any element added to an SP implies work and time to develop and maintain it.

5. Application Example and Discussion

This section discusses the application of the proposed evaluation model with three concrete websites: inaturalist.org, identify.plantnet.org, elixir-europe.org (see Table 2). These websites were chosen because they have environmental goals and represent an important community of end-users interested in biological and ecological issues. SP3 has significantly distinct goals as compared with those of SP1 and SP2. Other SP could have been chosen, but these are commonly used enough to explain the proposal.

Table 2. SPs used to apply the proposed model.

SP Id	Website	Description
SP1	inaturalist.org	Community of people interested in a database of species
SP2	identify.plantnet.org	Community of people interested in a database of plants species
SP3	elixir-europe.org	Community of life sciences institutions interested in developing a database of scientific knowledge

Table 3. Multicriteria evaluation model for environmental sustainability of software products.

Input Matrix				Software products					
				SP1		SP2		SP3	
Category	Goal	Weight (W)	Direction (D)	Performance (P)	Weighted P (WP)	P	WP	P	WP
C1	G1	0.4	**-1**	2	-0.8	1	-0.4	1	-0.4
C2	G2	0.4	**-1**	2	-0.8	0	0	1	-0.4
C8	G8	0.2	-1	53	-10.6	247	-49.4	98	-19.6
C7	G7	D	**1**	2	n.a.	2	n.a.	3	n.a.
	Score				-4.01		-16.6		-6.8

Table 3 shows the application of the evaluation model with these SPs.

Criteria *Color*, *Video* and *Elements* are used in a continuum of integer values. A higher number for each criterion means a lower score value because lighter colours and moving images imply more energy consumption [2; 9]. A higher number of elements imply more development effort, meaning more energy and resource consumption. Elements in SP1 is 53 (2 *Color* plus 2 *Video* plus 25 text plus 24 image), in SP2 is 247 (1 *Color* plus 0 *Video* plus 201 text plus 45 image) and in SP3 is 98 (1 *Color* plus 1 *Video* plus 84 text plus 12 image). Both text and image elements are not computed for simplicity and because they are considered not as much relevant as others. *SGoals* is criteria that divides the total score of other not abstract criteria to obtain a total value that considers the complexity of different goals between software products. The *SGoals* identified, from the end-user perspective, and classified as suggested by Function Point Analysis, i.e., the system components as seen by end-user where [31]: (i) Build a database; (ii) Connect with users; (iii) Retrieve Inst. Info and (iv) Management Support.

This example showed that RE makes it possible: it may add rigour and efficiency when performing tasks. This is a reality, even though our MCA and library of environmental sustainability requirements include only a restricted number of elements to guarantee its simplicity. The result of using *ESustainability* is the score -4.01 for SP1, -16.6 for SP2 and -6.8 for SP3. SP2 is the one with the lowest value, i.e., the one whose change is the most relevant, because regardless of having no videos and only having one colour that need to be changed, it contains 247 elements regardless of having the same number of goals as SP1. This value is reached even if ignoring that this number of elements represents a very heterogenous group, such as algorithms, text, and images, being images a very relevant type of element in website evaluation due to eventual communication burdens they can be responsible for [9].

Nevertheless, for simplicity, the elements' distinction was also ignored that only

SP2 (*identify.plantnet.org*) do not use moving images and algorithms were not counted. We also acknowledge that the objective of *elixir-europe.org* is more complex and has more goals, and as a result, we understand that it is performing relatively better when compared with SP1 and SP2. Regardless of the score and goals, any of these websites could be further optimized for environmental sustainability, e.g., by eliminating or reducing automatic movement, simplifying the structure (fewer elements), avoiding external links, etc. Also, the criteria *Aesthetics* can complement our analysis showing that SP1 and SP2 need to change their background colours because they are too light. Nevertheless, measuring the area that the colour to be changed covers needs an automatic procedure with fine granularity.

Our model assumes that the website most in need of change, i.e. SP2, is the same that *http://www.websitecarbon.com/* [32] indicates as the least environmentally friendly, as measured on 21 March 2022: (i) *inaturalist.org* produces 1.42g of CO_2 per visit; (ii) *identify.plantnet.org* produces 2.30g of CO_2 per visit, and (iii) elixir-europe.org produces 0.87g of CO_2 per visit.

The *sustainablewebmanifesto.com* only produces 0.01g of CO_2 per visit, and it might be considered a benchmark for improving more environmentally sustainable websites. While our approach considers an SP as a white box, where every element is perfectly identifiable, *http://www.websitecarbon.com* is an example of a black-box approach where only the energy performance is measured. In our approach, first we audit the SP, identifying each element and showing the ones that can justify a given result and should be changed to obtain more environmentally friendly performances. In other words, this method produces information needed for the changes to be made automatically for adapting existing SP.

It is assumed that an improvement is possible regardless that savings of resources do not always mean less environmental impact due to the impacts of the service provided, and the difficulties of recycling continue to exist.

6. Conclusion

This paper proposes and discusses a multicriteria analysis to evaluate software products regarding their environmental impact. This model is supported by a library of requirements for environmental sustainability, defined as general goals. Previous work that considers green requirements engineering should be developed within the context of general-purpose requirements engineering is acknowledged [18].

The new idea we present in this paper is the application of MCA with only a very limited number of criteria. This first step is building an SP to make others more environmentally sustainable. This proposal might use an algorithm that inputs the data provided by our model and outputs sequences of instructions of change for the SP. Furthermore, this number of criteria is not only for SP but also for industrial products. It needs more testing, especially in industrial design using digital twins, GREEN through IT testing. For instance, (1) a printer software that optimizes environmental sustainability of paper printed documents; (2) an SP that searches new usages for natural molecules or raw materials; or (3) software that tests a new machine to collect microplastics in a laboratory.

Future research would include extending and improving the rigorous specification of the discussed requirements library, maintaining simplicity, and designing a tool to better support the audit and evaluation processes. We also plan to conduct more case studies and tests to help validate and extend the model proposed in this article.

Acknowledgements

Research partially funded by FCT UIDB/50021/2020 and 02/SAICT/2017/29360.

References

1. Barker, D.: Web Content Management, O'Reilly, (2016)
2. Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S.: Sustainability design and software: The Karlskrona manifesto, in Proceedings of the 37th International Conference on Software Engineering vol. 2, ICSE'15, IEEE Press, Piscataway, NJ, USA, 467–476 (2015)
3. Blin, M., Tsoukiàs, A.: Multi-Criteria Methodology Contribution to the Software Quality Evaluation, Software Quality Journal, 9, pp. 113–132, (2001)
4. Carvalho, R., Silva, A.: Sustainability Requirements of Digital Twin-Based Systems: A Meta Systematic Literature Review. Appl. Sci. 2021, 11, 5519. https://doi.org/10.3390/app11125519. (2021)
5. Dastbaz, M., Pattinson, C., Akhagar, B.: Green Technology a Sustainable Approach, MK, XXV + 321 (2015)
6. Easterbrook, S.M.: Climate change: a grand software challenge. In: of the FSE/SDP WS on Future of Software Engineering Research, FoSER 2010, pp. 99–104. ACM (2010)
7. ELIXIR (2022), elixir-europe.org. Accessed March 21, 2022
8. Finland Ministry of Transport (FMT): The two faces of the ICT sector: consuming energy and materials while also paving the way to a carbon-neutral society, https://www.lvm.fi/-/the-two-faces-of-the-ict-sector-consuming-energy-and-materials-while-also-paving-the-way-to-a-carbon-neutral-society-1210283. Accessed 23 March 2022 (2022)
9. Greenwood, T., https://www.wholegraindigital.com/blog/website-energy-efficiency/. Accessed March 26, (2022)
10. Hilty, L., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., Wager, P.: The relevance of information and communication technologies for environmental sustainability. Environm. Modelling & Software, 21(11):1618 – 1629, (2006)
11. Ihor, B., Oleksii, D., Aleksandr, K., Nataliia, K., Oleksandr, M., Volodymyr, P.: Multicriteria Choice of Software Architecture Using Dynamic Correction of Quality Attributes. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham. (2020) https://doi.org/10.1007/978-3-030-16621-2_39
12. iNaturalist, https://www.inaturalist.org. Accessed March 21, (2022)
13. Kern, E., Hilty, L., Guldner, A., Maksimov, Y., Filler, A., Gröger, J., Naumann, S.: Sustainable software products—Towards assessment criteria for resource and energy efficiency, Future Generation Computer Systems, 86, 199-210 (2018) https://doi.org/10.1016/j.future.2018.02.044
14. Lago, P., Koçak, S., Crnkovic, I., Penzenstadler, B.: Framing sustainability as a property of software quality. Commun, ACM, 58 (10), 70-78, (2015)
15. Magel, K., Kluczny, R., Harrison, W., Dekock, A.: Applying Software Complexity Metrics to Program Maintenance, Missouri University of Science and Technology, 65-79 (1982), DOI: https://doi.org/10.1109/MC.1982.1654138
16. Malhotra, A., Melville, N., Watson, R.: Spurring Impactful Research on Information Systems and Environmental Sustainability, MIS Quarterly, 37, 1265-1274 (2013)
17. Paiva, A. C. R., Maciel, D., Silva, A. R.: From Requirements to Automated Acceptance Tests with the RSL Language, Communications in Computer and Information Science, 1172 (ENASE'2019 Selected Papers), Springer (2020)
18. Penzenstadler, B.: Infusing Green: Requirements Engineering for Green In and Through Software Systems, Third International Workshop on Requirements Engineering for Sustainable Systems, 44-53 (2014)
19. Pimentel, J., Castro, J.: piStar tool - a pluggable online tool for goal modeling. In: 26th International Requirements Engineering Conference (RE), pp. 498–499 (2018)
20. Pl@ntNet, identify.plantnet.org/pt. Accessed March 21, (2022)
21. Questionnaire Guide on the Sustainability Catalogue Using the Pistar Tool. Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa (2021)
22. Steffen, W., Richardson, K., Rockström, J., Cornell, S., Fetzer, I., Bennett, E., Biggs, R., Carpenter, S., Vries, W., Wit, C., Folke, C., Gerten, D., Heinke, J., Mace, G., Persson, L.,

Ramanathan, V., Reyers, B., Sörlin, S.: Planetary boundaries: Guiding human development on a changing planet, 347 (6223), 1259855-1-1259855-10 (2015)
23. Subcommission on Quaternary Stratigraphy: Working Group on the 'Anthropocene', http://quaternary.stratigraphy.org/working-groups/anthropocene/. Accessed March 22, (2022)
24. Sustainable Web Manifesto, https://www.sustainablewebmanifesto.com/. Accessed March 21, (2022)
25. United Nations (EU) (2022), "TOOL #63. MULTI-CRITERIA ANALYSIS", https://ec.europa.eu/info/sites/default/files/file_import/better-regulation-toolbox-63_en_0.pdf. Accessed March 26 2022.
26. United Nations (EU): European Green Deal, (2019)
27. United Nations (UN): Our Common Future: Report of the World Commission on Environment and Development (un.org). (1987) Available online https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf
28. United Nations (UN): Transforming our World: The 2030 Agenda for Sustainable Development, 1-41, 25 to 27 September 2019
29. USDA, More About Ecosystem Services, 7 October 2016
30. Venters, C., Jay, C., Lau, L., Griffiths, M. Holmes, V., Ward, R., Austin, J., Dibsdale, C., Xu, J.: Software Sustainability: The Modern Tower of Babel, Position Paper: RE4SuSy: 3rd International Workshop on Requirements Engineering for Sustainable Systems (2014)
31. Vickers, P.: An Introduction to Function Point Analysis. Northumbria University, UK, (2003)
32. Website Carbon Calculator (2022), https://www.websitecarbon.com/. Accessed March 21, (2022)
33. Wu, F., Webster, C.: Simulation of land development through the integration of cellular automata and multicriteria evaluation. Environment and Planning B, 25:103– 126, (1998)
34. Zaidan, A., Zaidan, B., Hussain, M., Haiqi, A., Kiah, M., Abdulnabi, M.: Multi-criteria analysis for OS-EMR software selection problem: A comparative study, Decision Support Systems, 15-27 (2015)
35. Zavadskas, E., Antuchevičienė, J., Kapliński, O.: Multi-criteria decision making in civil engineering: Part I – a state-of-the-art survey, Engineering Structures and Technologies, 7:3, 103-113 (2015), DOI: 10.3846/2029882X.2015.1143204
36. Gomes, A. C., Paiva, A. C. R., Silva, A. R.: Generating Test Cases from Use Cases and Structured Scenarios: Experiences with the RSL Language, Proceedings of ISD'2021, AIS (2021)