



Assessing the Sustainability Performance of Sustainability Management Software

Kern, Eva; Silva, Samantha Luisa; Guldner, Achim

Published in:
Technologies

DOI:
[10.3390/technologies6030088](https://doi.org/10.3390/technologies6030088)

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):
Kern, E., Silva, S. L., & Guldner, A. (2018). Assessing the Sustainability Performance of Sustainability Management Software. *Technologies*, 6(3), [88]. <https://doi.org/10.3390/technologies6030088>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Article

Assessing the Sustainability Performance of Sustainability Management Software

Eva Kern ^{1,2,*} , Samanthi Silva ³ and Achim Guldner ²¹ Faculty of Sustainability, Leuphana University Lüneburg, 21335 Lüneburg, Germany² Institute for Software Systems, Environmental Campus Birkenfeld, Trier University of Applied Sciences, 55768 Birkenfeld, Germany; a.guldner@umwelt-campus.de³ Centre for Sustainability Management (CSM), Leuphana University Lüneburg, 21335 Lüneburg, Germany; samanthi.silva@uni.leuphana.de

* Correspondence: mail@nachhaltige-medien.de; Tel.: +49-160-9686-6055

Received: 28 August 2018; Accepted: 16 September 2018; Published: 19 September 2018



Abstract: Companies have made considerable progress in assessing the sustainability of their processes and products, including the information and communication technology (ICT) sector. However, it is surprising that little attention has been given to the sustainability performance of software products. For this article, we chose a case study approach to explore the extent, to which software manufacturers have considered sustainability criteria for their products. We selected a manufacturer of sustainability management software on the assumption that they would be more likely to integrate elements of sustainability performance in their products. In the case study, we applied a previously developed set of criteria for sustainable software (SCSS) using a questionnaire and experiments, to assess a web-based sustainability management software product regarding its sustainability performance. The assessment finds that despite a sustainability conscious manufacturer, a systematic assessment of sustainability regarding software products is missing in the case study. This implies that sustainability assessment for software products is still novel, corresponding knowledge is missing and suitable tools are not yet being widely applied in the industry. The SCSS presents a suitable approach to close this gap, but it does require further refinement, for example regarding its applicability to web-based software on external servers.

Keywords: sustainability management; sustainable software; software energy measurements; sustainability criteria

1. Introduction and Motivation

Companies in the information and communication technology (ICT) sector increasingly consider sustainability issues in the way they conduct business [1–3]. Several companies jointly work towards sustainable developments in industry initiatives, such as the Global e-sustainability initiative [4]. Relevant topics include the effects of ICT throughout the life-cycle and supply chain of products and services [1,3,5], such as health and safety as well as energy issues.

Several analyses have investigated the effects that ICT has on society and the environment. The results were ambiguous, found to be positive, negative, or even to cancel themselves out [6,7]. In general, the effects of ICT on the environment were categorized into three effects [8–10]: First order (direct effects) derive from the physical existence of ICT and the services they provide. In the life-cycle phases of manufacturing or recycling, ICT hardware poses a risk by exposing humans and the environment to toxic materials, e.g., e-waste [1,7,11]. Second order (indirect effects) are the result of the constant use of ICT services, including transparency and increase of speed. Third order (also indirect) effects are aggregated long-term outcomes of using ICT. Yi and Thomas [12] summarized

the issues in their meta-analysis on the effects of ICT on the environment, concluding that the effect varies. Transport and travel offer both opportunities and risks, as emissions and energy consumption can be reduced through virtualization but the opportunity to collaborate might foster further business travel [7]. Table 1 provides a selection of issues and references. For social sustainability, ICT usage is perceived to have a positive impact on social developments [13], which includes both internal, for example employee-related, and external, for example community development-related, matters.

Table 1. Sustainability issues in the information and communication technology (ICT) sector.

Sustainability Issues	Source
Voluntary or regulatory requirements	Piotrowicz and Cuthbertson (2009) [1]
Effects of ICT in the supply chain: Health and safety, emissions, transport, etc.	Piotrowicz and Cuthbertson (2009) [1]; Prattipati (2010) [3]; Fuchs (2008) [7]; Silvius et al. (2009) [11]
Resource consumption and virtualization	Fuchs (2008) [7]
E-waste	Piotrowicz and Cuthbertson (2009); [1]; Silvius et al. (2009) [11]; Fuchs (2008) [7]
Use of ICT to foster sustainable development	Fuchs (2008) [7]; Galloway and Mochrie (2005) [14]; Mohamed et al. (2010) [15]

For software, a sub-section of ICT, a core difference in terms of sustainability is whether the software product itself can be called “sustainable” rather than providing the opportunity for more sustainable behavior and processes (“sustainable by software”). The latter refers to software that is sustainable due to its ability to create positive contributions to sustainable development, based on its use and application. For example, software for the sustainability management of organizations belongs to the aspect “sustainable by software”. The focus of this inquiry is whether approaches for sustainability by software (e.g., sustainability management software) is in itself a “sustainable” software. To explore this, the paper uses a case study approach focusing on one manufacturer of sustainability management software for enterprises. This paper analyzes, evaluates, and categorizes the different aspects of sustainable software by applying a previously published framework for assessing sustainability software [16], which is introduced next (Section 2). A summary of the case study methodology follows (Section 3) and leads to the results (Section 4). The results and subsequent implication for research and practice for the sustainability assessment of software are critically discussed (Section 5). The manuscript concludes with a summary and potential future research questions (Section 6).

2. Sustainability Assessment of Software

The sustainability assessment of the software, remains, despite several calls for research, a niche topic [17–20]. Indeed, based on the abilities and characteristics of software, the hardware capacities are determined, affecting the resource consumption during its usage phase (e.g., of end-user devices, networks and data centers [21]). In order to assess sustainability of software, the first step is to define and characterize sustainable software products [19,22–29]. This paper builds on the definition of sustainable software by Dick et al. [30] that sustainable software positively contributes to sustainable development including its economic, societal and environmental effects and is sustainable itself. Furthermore, research has identified a multitude of possible criteria to assess sustainable software products. Existing approaches include quality models for software, e.g., ISO 25010, or built on the life cycle assessment (LCA) approach. Calero et al. [26,31] propose an extended version of the ISO model, called ISO 25010+S that includes sustainability issues by defining an additional quality aspect “greenability”. They propose criteria such as energy efficiency, resource optimization, capacity optimization, and perdurability. Naumann et al. [5] present a life cycle of software products and map effects regarding sustainable development such as working conditions, download size, accessibility, hardware requirements, and backup size across the life cycle. The causal model published by Kern et al. [16] relates observable software properties (e.g., resource management functionality, user autonomy) to hardware aspects, issues of behavior and organization, and natural resources. In line with the causal model, a set of criteria for sustainable software (SCSS) by Kern et al. [16] is proposed,

building on an extensive review of the academic literature on relevant criteria for green and sustainable ICT. The SCSS follows a hierarchical order, presenting indicators to assess the impacts of software on sustainable development, focusing on the usage phase of software [5]. It encompasses three main aspects to assess the impact of software on natural resources from a life-cycle perspective: Resource efficiency, potential hardware operating life, and user autonomy. Each of them is further differentiated with corresponding sub-criteria (see Table 2).

Table 2. Set of criteria for sustainable software products (SCSS): main criteria and corresponding sub-criteria [16]. The SCSS is available online at: [32].

1	Resource Efficiency
1.1	Hardware efficiency: Which hardware capacities must be available for operating the software product and what is the degree of capacity utilization during operation?
1.2	Energy efficiency: How much electricity does the hardware consume when the software product is used to execute a standard usage scenario?
1.3	Resource management: Does the software product have an energy management feature, and how effective is it when using the product in a standardized context?
2	Potential hardware operating life
2.1	Backward compatibility: Does the manufacturer of the software product guarantee that the current release can be executed on a reference system that is n years old?
2.2	Platform independence and portability: Can the software product be executed on different currently prevalent productive system environments (hardware and software), and can users switch between them without disadvantages?
2.3	Hardware sufficiency: Does the amount of hardware capacity used remain constant over time as the software product is developed further and additional functions are added?
3	User autonomy
3.1	Transparency and interoperability: Can users understand resource-relevant aspects of the software product with a reasonable amount of time and effort? Are they free to re-use data they produced with this software product with other software products?
3.2	Uninstall ability: Can the software product be uninstalled easily, without leaving traces, and without avoidable disadvantages?
3.3	Maintenance functions: Does the software product provide easy-to-use functions permitting users to repair damage to data and programs?
3.4	Independence of outside resources: Can the software product be operated as independently as possible of resources not subject to the users' control?
3.5	Quality of product information: Does the information provided about the software product support its resource-efficient use?

For a better understanding if and how sustainability-conscious companies consider sustainability for their own product, this paper applies the SCSS to the case of a manufacturer of sustainability management software. The paper focuses on a web-based software, including experiments to assess the sustainability of the software (e.g., network traffic) and additionally a questionnaire and interview data. The analysis ensures reliability by reviewing verified product information and user manuals. With this approach, we applied the SCSS in a new context and, therefore, also critically reviewed it.

3. Methods

To answer the research question of the article, a single case study design [33] was applied. Based on the lack of theories and analyses of sustainability in software, a case study is a suitable approach for exploring this area [33,34]. Following the six-step process by Ying [33], the research question "How can sustainability management software itself be assessed in terms of sustainability" was developed (step 1). Regarding the design (step 2 [33]) a single case study approach was chosen, to explore the sustainability of a web-based software for sustainability management, which is the unit of analysis. For the exploration, the previously introduced SCSS [16] was chosen to investigate the selected sustainability management software: assessment of the sustainability software, based on the

SCSS and thus focusing on the usage phase of software. To do so, we critically reviewed the SCSS, published in [16], regarding its application to web-based software. Table 3 shows the selected criteria from the SCSS applied to assess the WeSustain Enterprise Sustainability Management (ESM) software. For this paper, we chose to focus on qualitative criteria as previous publications predominantly focused on measuring energy consumption of software [16,35–37].

Table 3. Selection of the criteria that are included in the sustainability assessment of the WeSustain Enterprise Sustainability Management (ESM) software against the SCSS (\checkmark = applied; (a)–(d) = named indicator applied; - = criterion not applied).

1	Resource efficiency	
1.1	Hardware efficiency	
1.1.1	Recommended system requirements and resulting hardware requirements (including peripheral devices)	\checkmark
1.1.2	Minimum system requirements and resulting hardware requirements (including peripheral devices)	\checkmark
1.1.3	Hardware utilization in idle mode assuming a standard configuration	(d)
1.1.4	Hardware utilization during normal use assuming a standard configuration and a standard usage scenario	(d)
1.1.5	Economical use of hardware through adaptability and support for users when adapting the software product	(d)
1.1.6	Online delivery	-
1.2	Energy efficiency	-
1.3	Resource management	
1.3.1	Adaptation of hardware capacities used to current demand	\checkmark
1.3.2	Adaptation of hardware capacities used to current supply	\checkmark
1.3.3	Default settings supporting resource conservation	-
1.3.4	Feedback on use of hardware capacities and energy	\checkmark
2	Potential hardware operating life	
2.1	Backward compatibility	\checkmark
2.2	Platform independence and portability	\checkmark
2.3	Hardware sufficiency	\checkmark
3	User autonomy	
3.1	Transparency and interoperability	
3.1.1	Transparency of data formats and data portability	\checkmark
3.1.2	Transparency and interoperability of the programs	\checkmark
3.1.3	Continuity of the software product	(a), (b)
3.1.4	Transparency of task management	(c)
3.2	Uninstall ability	
3.2.1	Uninstall ability of programs	-
3.2.2	Capability to erase data	(c)
3.3	Maintenance functions	
3.3.1	Recoverability of data	\checkmark
3.3.2	Self-recoverability	-
3.4	Independence of outside resources	
3.4.1	Offline capability	-
3.5	Quality of product information	
3.5.1	Comprehensibility and manageability of product documentation, licensing conditions, terms of use	\checkmark
3.5.2	Resource relevance of product information	\checkmark

For the preparation of the sustainability assessment (step 3 [33]) one web-based software was carefully selected. Rather than focusing on large scale software solutions for corporations, a focus was set on software suitable for small and medium-sized enterprises (SME) as the majority, 99.8 percent of companies in the EU, are SMEs [38]. The software products were selected from a list of web-based sustainability software for SMEs presented by Johnson et al. [39], with a focus on integrated sustainability aspects rather than a sole selection of ecological aspects, narrowing the list from ten to five. Furthermore, web-based sustainability management software products from the category “assessment and reporting” were chosen and the manufacturers were asked for collaboration, finally resulting in the selection of *CR Kompass*. After reviewing the *CR Kompass* with the manufacturer, WeSustain GmbH, the decision to rather focus on their software product *WeSustain ESM*, (ESM stands for Enterprise Sustainability Management) was made due to its higher flexibility and validity. To collect the data (step 4 [33]), the selected web-based sustainability management software was assessed, based on the SCSS [16]. With a focus on relevant criteria for web-based tools and to consider at least one criterion from each main level criteria (level 1, Table 2) the following categories were chosen:

- Criterion 1.1 Hardware efficiency and 1.2 Energy efficiency (the assessment is confined to indicators regarding network traffic)
- Criterion 1.3 Resource management
- Criterion 2. Potential hardware operating life
- Criterion 3.1 Transparency and interoperability

The case study furthermore included an experiment of 30 measurements for network traffic, following a typical use case (standard usage scenario; necessary for application of the SCSS by Kern et al. [16]):

- Start the WeSustain ESM software
- Select key performance indicators
- Generate diagram and download it as .png file
- Generate report and download it as .pdf and .doc file
- Delete report and diagram
- Insert additional key performance indicator
- Insert information on the company
- Format the report
- Close the software

A detailed version of the usage scenario is included in the replication package of this article: [40]

The scenario followed the measurement method presented in [35]. The KPIs were selected randomly from the GRI G4 indicators as these are mostly frequently used by the users of WeSustain EMS, according to WeSustain GmbH. The results of the experiment were complemented with interviews and questionnaires with representatives of the manufacturer.

To analyze the data (step 5 [33]), we reviewed the case in depth regarding strengths, limitations and further research for assessing sustainability in web-based software, in particular with the application of the SCSS. Finally, the last step [33], is to share the data, e.g., in this publication. The results were also shared with WeSustain GmbH, the manufacturer of the web-based software product that had been assessed.

4. Results

The application of the SCSS to the case of WeSustain GmbH and its ESM software reveals that the sustainability management software itself has not been systematically assessed with regards to its sustainability performance. Rather, selected sustainability initiatives, e.g., sustainable servers, demonstrate the relevance of sustainability in software for this case. This section presents key results

of the case study of applying the SCSS to the web-based sustainability management software by WeSustain GmbH.

The detailed results of the case study, including experiment and questionnaire, are attached in the Supplementary Material of the article and online [40].

4.1. Hardware and Energy Efficiency: Measuring Network Traffic

To assess network traffic, the SCSS suggests reviewing the recommended minimum (indicator 1.1.1 (e) and 1.1.2 (e)) *network bandwidth*, as well as measuring the *average bandwidth utilization for network access in idle mode* (indicator 1.1.3 (d)) and *when running the usage scenario* (indicator 1.1.4 (d)).

According to the manual and the answers of the manufacturer in the questionnaire, “there are no technical requirements” beyond those gathered in criterion 2.1 *backward compatibility* [41]. Hence, indicator 1.1.1 (e) and 1.1.2 (e) cannot be assessed. The average bandwidth utilization in idle mode is too small to be distinguished from the baseline measurements (indicator 1.1.3 (d)).

To assess the indicator 1.1.4 (d) Measurement of average bandwidth utilization for network access when running the standard usage scenario under the standard configuration, the average network traffic of a system under test (SUT) was measured, running the standard usage scenario under the standard configuration as described by Kern et al. [16] and Guldner et al. [35]. The measurement results are depicted in Figure 1.

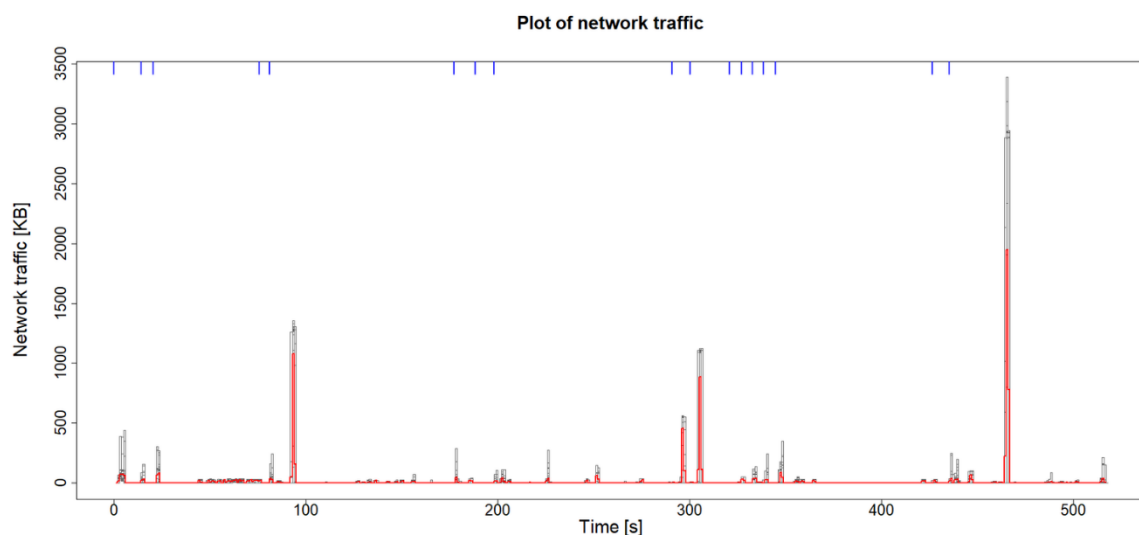


Figure 1. Average measured network traffic during standard usage scenario of WeSustain ESM.

The figure shows the 30 measurements of the standard usage scenario (grey) and the per-second average (red) of the standard usage scenario. The scenario took 516.6 s on average. During this time, one image of a plot, one PDF file and one Word document is downloaded. These downloads are also visible in the peaks of the network traffic; downloading the image (second 94), downloading the PDF report (second 297) and downloading the Word file (second 306). The last peak (second 466) is likely to result from searching all the available KPIs in WeSustain ESM. In order to receive the value for the average bandwidth utilization for network access when running the usage scenario (indicator 1.1.4 (d)), it is necessary to average the traffic for all measurements and to measure the baseline network traffic of the SUT. Figure 2 shows the measured averages for all measurements and baselines. Reducing the average measurement result (16.59 kB/s) by the average baseline measurement (0.42 kB/s), equals the net load for indicator 1.1.4 (d), which is 16.16 kB/s. In other words, during the 516.6 s of each standard usage scenario, 8.3502 MB (0.00835 GB) are transferred.

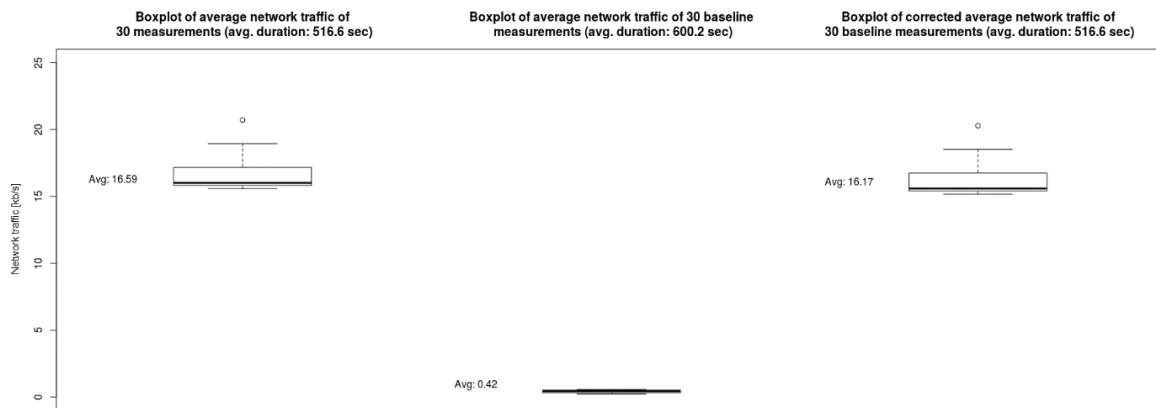


Figure 2. Average network traffic of scenario (l), baseline (m) and scenario reduced by baseline (net load for indicator 1.1.4 (d)) (r).

From these measurements, it is also possible to estimate values for indicators 1.2 (b) and 1.2 (c). Indicator 1.2 (a) *energy consumed on the local device for running the standard usage scenario* is not included in the analysis since we concentrated on qualitative criteria, except from criteria regarding network traffic. Indicator 1.2 (b) is the *estimation of the energy consumed in the network for the data traffic caused by running the standard usage scenario under the standard configuration (a current estimate of network energy intensity in kWh/GB from literature may be used, if necessary differentiated among types of access network)*. The following formula was used to estimate the energy for this scenario (E) [42]:

$$E = t(S) \times 52 \text{ W} + \text{GB}(S) \times 13 \text{ Wh/GB.}$$

$t(S)$ is the time it takes to transfer the data over the used internet connection and $\text{GB}(S)$ is the amount of transferred data in GB.

After consulting one of the original authors of the formula [37], it was modified to take the increased energy efficiency of modern networking devices in comparison to 2015 into account by reducing the energy consumption due to the transferred data by 4.

Transferring 0.00835 GB over the network connection used in the experiment (210 Mbit/s) takes approx. 0.318 s. Thus, the formula yields $0.318 \times 52 \text{ W} + 0.00835 \text{ GB} \times 13 \text{ Wh/GB} = 16.541 \text{ Ws} + 0.108552 \text{ Wh} = 16.541 \text{ Ws} + 390.787 \text{ Ws} = 407.329 \text{ Ws}$ for indicator 1.2 (b).

Indicator 1.2 (c) estimates the energy consumed by servers for the remote processing and storage for running the standard usage scenario under the standard configuration based on factors for average energy intensity of data center services from literature. For this estimation, the formula, developed by Valancius [42], was applied: 211.1 Joules per transferred Gbit, or

$$E = 211.1 \text{ Ws/GBit} \times \text{Traffic [GBit]}.$$

This results in an estimate of approximately 14.102 Ws for the scenario of this case (in which 0.066801265 Gbit were transferred).

Furthermore, for web-based software, especially during the usage phase, hosting aspects and data traffic are an issue. The manufacturer of WeSustain ESM informed us that their hosting partner uses energy from wind- and hydropower, demonstrating the relevance of sustainability considerations not only for their product in the direction of “sustainable by software” but also for the product itself as sustainable software.

About 19% of the network traffic is caused by the downloading process in the usage scenario (see Section 3) for the .png, .pdf and .doc files, resulting in 1.6 MB (out of overall 8.3502 MB, see indicator 1.1.4 (d)). Compared to measurement results of network traffic caused by using content management systems or database systems [16,35], the network traffic of this case study is low. The same is true for the indicators 1.2 (b) and (c). However, it depends on the usage scenario that is completely different for any kind of software products. It would be interesting to compare the results of different web-based

sustainability management software. Generally, the data traffic induced by the software could possibly be reduced by using application level cache, optimized images and compression, as shown by Dick et al. [43].

4.2. Resource Management

The criterion *resource management* addresses the question “To what extent does the software product contribute to efficient management of the resources it uses during operation?” Table 3 summarizes the four sub criteria and corresponding indicators.

The analyzed software product does not allow for the adaption of hardware capacities to the current supply (criterion 1.3.2). As the assessed software does not provide default settings, the corresponding indicator of criterion 1.3.3 *default settings supporting resource conservation* cannot be analyzed. This criterion reviews if the default settings of the software product take the goal of resource conservation into account [16]. The corresponding indicator 1.1.3 (a) is *reviewer’s assessment whether the default settings of the software product are selected in such a way that they also take the goal of resource conservation into account*. Furthermore, the software does not support energy-friendly modes by providing corresponding settings. Additionally, the user does not receive feedback on the usage of hardware capacities and energy (indicator 1.3.4 (a)). Thus, so far, the analyzed software product does not take resource management issues into account.

This offers great potential for improving the software product regarding its resource management. These results are common and resemble the assessment results of other software products, e.g., internet browsers, word processing, content management systems and databases [16,35]: Apart from a sleep mode that is started after a maximum of about 60 s, settings for resource management are not included in the previously tested software products, either. Web browsers provide feedback on use of hardware capacities and energy. For databases, feedback on the use of hardware capacities and energy, is, in two-out-of-three products, to some extent, presented [16,35]. Using energy management functions does not influence the usability of the analyzed web browsers, content management systems and databases.

These previous results arose from the assessment of non-sustainability-oriented software. Therefore, a different outcome was expected based on a more sustainability-oriented software manufacturer. Similar results across sustainability and non-sustainability-oriented manufacturers, however, show the novelty of this field, both in research and practice.

4.3. Potential Hardware Operating Life

The second criterion addresses the question “To what extent are hardware replacement cycles decoupled from software replacement cycles?”, encompassing four sub-criteria and corresponding indicators.

To analyze the backward compatibility (criterion 2.1), we asked the manufacturer which hardware—old(er) operating systems and old(er) frameworks—are supported by the software product. The software manufacturer states that 95% of the customers use the software product in the cloud. Hence, using WeSustain ESM requires, according to the manual and the manufacturer, a current web browser, i.e., Internet Explorer version 8.0 or later, Firefox version 7.0 or later, Chrome version 15.0 or later, or Safari version 5.0 or later. For a comparison of the software product, the SCSS [16] suggests testing the product on different standard configurations: on a current version and an earlier one. Since this is the first application of the SCSS on this software product, this additional indicator was neglected.

As specified by the manufacturer, the software can be executed on the operating systems Linux and Windows. The product is a web-based tool and was thus not installed nor executed on the server of the testing environment during the assessment. Therefore, the standard usage scenario was not run on different server systems. However, regarding the client side, WeSustain ESM was successfully run on Windows using the internet browsers Google Chrome (version 67.0.3396.99 (64-Bit)), Microsoft Edge (version 42.17134.1.0) and Firefox (version Quantum 61.0.1 (64-Bit)), on Linux, i.e., Ubuntu 16.4.5 LTS using Firefox (version Quantum 61.0.1) as well as on Android 6.0 using Chrome (version Chrome App 68.0.3440.85).

Based on the results of the questionnaire, the manufacturer does not compare the amount of hardware capacity used when updating and extending the software product. Thus, *hardware of sufficiency* (criterion 2.3) was not rated during the case study. Similarly, the criterion 2.3 had to be excluded from the assessment [16,35] since it was the first analysis.

Summarizing, the software product is, since it is web-based, highly platform independent and portable. Indeed, so far, the manufacturer was not aware of issues regarding backward compatibility and hardware sufficiency. Compared to previously assessed desktop software [16,35], it is more complex to analyze the amount of used hardware for web-based software, which is in line with previous results, e.g., Kreten et al. [44].

4.4. Transparency and Interoperability

The criterion *transparency and interoperability* addresses the questions “Can users understand resource-relevant aspects the software product with a reasonable amount of time and effort? Are they free to re-use data they produced with this software product with other software products?” It encompasses four sub-criteria and corresponding indicators. The criteria, their corresponding questions and indicators as well as the results of the assessment of the WeSustain ESM software are presented in Table 4 in detail.

Table 4. Results of the assessment of the criterion “3.1 Transparency and interoperability” to the sustainability-oriented web-based software product “WeSustain ESM” (* information found in manuals and suchlike, added to the answers of the WeSustain GmbH on the questionnaire).

Criterion and Corresponding Question	Indicator	WeSustain ESM
3.1.1. <i>Transparency of data formats and data portability</i> Is sufficient documentation provided for the data formats (file or data stream formats) used by the software product to enable interoperability? Do the data formats comply with open standards enabling further use of the data with another software product?	(a) Review of manuals and technical data sheets, comparison with known open standards	xls, csv, REST interface, docx *, html *, png *, pdf *
	(b) Check of compliance with known and open standards.	pdf1.4, Document does NOT conform to PDF/A
3.1.2. <i>Transparency and interoperability of the programs</i> Are application programming interfaces (APIs) clearly documented, and are dissemination and further development of the program supported? Do the interfaces comply to open standards to enable interoperability?	(a) If APIs exist: Review of the documentation of the interfaces on the basis of the documentation of the software product and its APIs	Yes, APIs are available
	(b) Is the source code open?	No
	(c) Is the software released under a license that allows it to further develop it?	No
3.1.3. <i>Continuity of the Software Product</i> Can the software product be used for longer periods of time without serious negatives (in particular IT security problems) occurring, and does the user have the option to avoid unnecessary updates?	(a) How long is the time period for which the supplier guarantees future support for the product, including security updates?	10 years
	(b) Does the manufacturer respond promptly when security gaps (vulnerabilities) become known?	Yes, 24 h in case of critical issues
	(c) Can the user influence the frequency of updates by configuring the software product and when doing so differentiate between security updates and other updates?	Not included in the analysis
	(d) Is it possible to receive differential updates only?	Not included in the analysis

Table 4. Cont.

Criterion and Corresponding Question	Indicator	WeSustain ESM
3.1.4. <i>Transparency of Task Management</i> Does the software product inform users that it is automatically launching or running tasks in the background that are possibly not being used?	(a) On the basis of the installation and the execution of standard usage patterns, test which processes are automatically launched by the software product and whether it informs users of this (Scale: informs users of all such processes/informs users of some such processes/does not inform users)	Not included in the analysis
	(b) If the software product is automatically launched at system start (“autostart”): does it inform users that this is the case?	Not included in the analysis
	(c) If the user carries out an action that can be understood as ending the program, but at least one of the tasks remains active: does the software product inform the user that this is the case?	Automatic logout when closing the browser; software informs user if there is unsaved data

First, the criterion regarding transparency assesses supported data formats (criterion 3.1.1). The manufacturer states that the data can be exported in Excel and CSV format as well as over a REST interface. Additionally, the software offers the possibility to export .pdf, .png, .docx, and .html files. According to the answers of the manufacturer, APIs are available for the software product (criterion 3.1.2). However, the documentation of these was not reviewed, since it was not available for the case study. The software product is neither open source nor released under a license that allows its further development.

The manufacturer guarantees ten years of support for the product, including security updates. Regarding the protocol security gaps (vulnerabilities), indicator 3.1.3 (b), the manufacturer did not provide specific information, but highlighted that users receive an answer within 24 h for critical issues. The indicator (c) and (d) of the criterion 3.1.3 (*influence of users regarding frequency and differential updates*) are not applicable to the case of web-based software and are thus excluded from the analysis.

The indicators (a) and (b) of the criterion *transparency of task management* refers to products that are directly installed on the client system. Hence, they were neglected in the current sustainability assessment. The task management of the software WeSustain ESM is transparent in case of unplanned user action that might be followed by data loss: When users leave the web page, they receive a pop-up stating that that the data is not yet saved. Users are logged out when they close the window of the browser.

In summary, transparency issues are partly taken into consideration in case of the analyzed software product. This complies with previously tested commercial products [16,35]. Generally, the application of the SCSS demonstrates that some transparency criteria are only applicable to installed desktop products.

5. Discussion

Sustainability management software can be seen as a facilitator to more sustainable behavior, but it can also be seen as an example for sustainability efforts by being sustainable itself. The application of the SCSS to the web-based software “WeSustain ESM” offers one approach for how sustainability management software itself can be assessed in terms of sustainability (research question). The questionnaire used to assess the sustainability of the software identified information on how the software manufacturer addresses sustainability in their products. Additionally, the article presents a critical reflection of the SCSS regarding web-based software executed on an external server. Both aspects are discussed in the following section.

5.1. Consideration of “Sustainable Software” by the Software Manufacturer

The case study confirms results from previous studies, that the sustainability assessment of sustainable software seems to still be in its infancy [45], because even software manufacturers with products that focus on sustainability, seem to lack the abilities to include these considerations in the software development process. A sensitivity of the manufacturer in our case study for sustainability is demonstrated by the consideration of data centers that use renewable energy (see Section 4.1). Nevertheless, other sustainability consideration for the software product itself seem to be neglected. Therefore, the sustainability assessment of sustainable software seems to be a novel approach and future research should investigate how these considerations can broadly be made available.

Generally, software developers are challenged to more strongly integrate the ideas of the SCSS, e.g., options to reduce the resource and energy consumption, into their software development processes.

Future research should complement the case study with broader empirical results. A focus could be on the market for sustainability management software, interrogating the question “How are aspects of sustainable software considered by software manufacturers providing sustainability management software?” This case study indicates a low focus and lack of appropriate tools such as the SCSS in practice. The research is not only beneficial in terms of understanding the status in practice, but could, with the help of accompanying workshops, show possible sustainability assessment approaches and demonstrate the relevance and benefits in practice, as well. However, the market for sustainability management software is just one possibility to extend the research regarding sustainability issues of software products. Further examples include public authorities, software companies offering tools for earth observation or for analyzing environmental data.

Practical implications from this research include ideas for the optimization of the software product, complimentary to the technical recommendations published by Dick et al. [43] (see Section 4.1). In order to provide specific indications for improving the sustainability of web-based tools, it is necessary to extend the research by accessing the server on which the software is hosted.

5.2. Application of SCSS to Web-Based Software

A novel focus of this case study is the application of SCSS to a web-based software. Therefore, this section critically reviews and discusses the applicability of the SCSS to web-based software, suggesting implications for further research.

In general, web-based software indicates a reduction of environmental impacts. Dick et al. [46] highlight that “Green Web Engineering describes the art of developing, designing, maintaining, administrating, and using a website in such a manner, that direct and indirect energy consumption within the complete life cycle of a website is reduced”. The SCSS includes green web aspects by addressing network traffic in different criteria and corresponding indicators (see section 0 of this article). While assessing these aspects in the case study, the SCSS was critically reviewed regarding the application to web-based software that is hosted on external servers (black box assessment) and, thus, direct measurements of the server side is impossible.

This results in the elimination of the following criteria from the assessment, because they have no or limited applicability to web-based software:

- 1.3.3. Default settings supporting resource conservation
- 3.1.4. Transparency of task management
- 3.2.1. Uninstall ability of programs
- 3.2.2. Capability to erase data

While the criteria regarding data traffic are especially interesting for the case of web-based software, some criteria of the SCSS are inapplicable for this kind of software products as no access to the servers is possible:

- 1.1.6. Online delivery
- 3.4.1. Offline capability

Additionally, some criteria need to be more strongly specified. The criterion 2.2 *platform independence and portability* asked for the possibility to execute the software on different operating systems and runtime environments. However, it does not specify if the indicator relates to the client or the server system. Similarly, the term “known and open standards” in indicator 3.1.1 (b), belonging to the criterion *transparency of data formats and data portability*, should be defined more clearly. For example, the WeSustain EMS software supports the .pdf, .png, .html, .docx, .xlsx, and .csv file format. The files can be opened without any problems. However, when validating it by using the web tool “www.pdf-online.com/osa/validate.aspx” the result is “Compliance: pdf1.4, Document does NOT conform to PDF/A”. The SCSS provides no information for the interpretation of this result. Thus, it is inconclusive whether this indicator is achieved or not.

Based on the findings in this paper, it is advisable (i) to extend the application of the SCSS on web-based software and consequently improve it, (ii) to assess further sustainability management software, and (iii) to review the SCSS in cooperation with software developers. Future activities require collaboration of research and practice to draw attention of software users in professional contexts (like sustainability management, analysis of environmental data, etc.) to the sustainability issues summarized in the SCSS. Following this path leads to a comprehensive approach: using sustainable software to support sustainability management processes.

6. Conclusions and Outlook

This paper extends the research by assessing sustainability management software by means of the set of criteria for sustainable software (SCSS), thereby applying the SCSS to web-based software. Thus, it provides innovative results by connecting the aspects of “sustainable by software” and “sustainable software”. This case study confirms previous publications that sustainability aspects of software are rarely considered. For this case of a sustainability management software manufacturer, first approaches to sustainable software, e.g., by using sustainability-oriented data centers, are indicated, pointing to the willingness to include sustainability considerations in their software products but missing the necessary tools for assessment and potential for improvement.

The SCSS presents such a tool. By applying the SCSS to a novel environment, a web-based software run on external servers, the SCSS was critically evaluated regarding its applicability to this context. This pointed to several areas for improvement, including more precise definitions in order to more easily apply the SCSS to web-based software, e.g., by providing information about which criteria is suitable for what kind of software products: local software application, software application with external data storage, software application with external execution, or server application.

Supplementary Materials: Externally hosted material: standard usage scenario, results of the assessment of the set of criteria for sustainable software (SCSS) to the WeSustain ESM software, measurement data: <https://doi.org/10.5281/zenodo.1344314>.

Author Contributions: Conceptualization, E.K., S.S.; Data curation, A.G.; Investigation, E.K., A.G.; Methodology, E.K., S.S., A.G.; Project administration, E.K.; Validation, E.K., S.S., A.G.; Writing—original draft, E.K., S.S., A.G.; Writing—review and editing, E.K., S.S., A.G.

Funding: This research received no external funding.

Acknowledgments: We thank the Research Group Informatics and Sustainability at the University of Zurich, the Öko-Institut e.V. in Berlin, the Institute for Software Systems at the Environmental Campus Birkenfeld and the Centre of Sustainability Management at Leuphana University Lüneburg. The case study would not have been possible without the help of WeSustain GmbH, we thank Niclas Schwarz and Jessica Warrings for access to the ESM tool and answering the questionnaires.

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

References

1. Piotrowicz, W.; Cuthbertson, R. Sustainability—A new dimension in information systems evaluation. *J. Enterp. Inf. Manag.* **2009**, *22*, 492–503. [[CrossRef](#)]
2. Hürtgen, S. *Von Silicon Valley nach Shenzhen. Globale Produktion und Arbeit in der IT-Industrie*; VSA-Verl.: Hamburg, Germany, 2009.
3. Prattipati, S.N. Sustainability and the role of information and communications technologies. *Bus. Renaiss. Q.* **2010**, *5*, 23–40.
4. Global E-Sustainability Initiative. About GESI. Available online: <https://gesi.org/about/mission-vision-ict-sustainability> (accessed on 15 August 2018).
5. Naumann, S.; Dick, M.; Kern, E.; Johann, T. The GREENSOFT Model: A Reference Model for Green and Sustainable Software and its Engineering. *Sustain. Comput. Inf. Syst.* **2011**, *1*, 294–304. [[CrossRef](#)]
6. Añón Higón, D.; Gholami, R.; Shirazi, F. ICT and environmental sustainability: A global perspective. *Telemat. Inf.* **2017**, *34*, 85–95. [[CrossRef](#)]
7. Fuchs, C. The Implications of New Information and Communication Technologies for Sustainability. *Environ. Dev. Sustain.* **2008**, *10*, 291–309. [[CrossRef](#)]
8. Forge, S. Powering down: Remedies for unsustainable ICT. *Foresight* **2007**, *9*, 3–21. [[CrossRef](#)]
9. Hilty, L.M.; Arnfalk, P.; Erdmann, L.; Goodman, J.; Lehmann, M.; Wäger, P.A. The relevance of information and communication technologies for environmental sustainability—A prospective simulation study. *Environ. Model. Softw.* **2006**, *21*, 1618–1629. [[CrossRef](#)]
10. Dedrick, J.; Kraemer, K.L.; Linden, G. Who profits from innovation in global value chains? A study of the iPod and notebook PCs. *Ind. Corpor. Chang.* **2010**, *19*, 81–116. [[CrossRef](#)]
11. Silvius, A.J.; van den Brink, J.; Smit, J. Sustainability in information and communications technology (ICT) projects. *Commun. IIMA* **2009**, *9*, 33.
12. Yi, L.; Thomas, H.R. A review of research on the environmental impact of e-business and ICT. *Environ. Int.* **2007**, *33*, 841–849. [[CrossRef](#)] [[PubMed](#)]
13. Young, A.G. Using ICT for social good: Cultural identity restoration through emancipatory pedagogy. *Inf. Syst. J.* **2018**, *28*, 340–358. [[CrossRef](#)]
14. Galloway, L.; Mochrie, R. The use of ICT in rural firms: A policy-orientated literature review. *J. Inf.* **2005**, *7*, 33–46. [[CrossRef](#)]
15. Mohamed, M.; Murray, A.; Mohamed, M. The role of information and communication technology (ICT) in mobilization of sustainable development knowledge: A quantitative evaluation. *J. Knowl. Manag.* **2010**, *14*, 744–758. [[CrossRef](#)]
16. Kern, E.; Hilty, L.M.; Guldner, A.; Maksimov, Y.V.; Filler, A.; Gröger, J.; Naumann, S. Sustainable software products—Towards assessment criteria for resource and energy efficiency. *Future Gener. Comput. Syst.* **2018**, *86*, 199–210. [[CrossRef](#)]
17. Penzenstadler, B.; Bauer, V.; Calero, C.; Franch, X. Sustainability in software engineering: A systematic literature review. In Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), Ciudad Real, Spain, 14–15 May 2012.
18. Penzenstadler, B.; Raturi, A.; Richardson, D.; Calero, C.; Femmer, H.; Franch, X. Systematic Mapping Study on Software Engineering for Sustainability (SE4S)—Protocol and Results ICS2 221: Irvine. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014.
19. Lago, P.; Koçak, S.A.; Crnkovic, I.; Penzenstadler, B. Framing sustainability as a property of software quality. *Commun. ACM* **2015**, *58*, 70–78. [[CrossRef](#)]
20. Verdecchia, R.; Ricchiuti, F.; Hankel, A.; Lago, P.; Procaccianti, G. Green ICT Research and Challenges. In *Advances and New Trends in Environmental Informatics: Stability, Continuity, Innovation*; Wohlgemuth, V., Fuchs-Kittowski, F., Wittmann, J., Eds.; Springer: Berlin, Germany, 2017; pp. 37–48.
21. Hilty, L.; Lohmann, W.; Behrendt, S.; Evers-Wölk, M.; Fichter, K.; Hintemann, R. *Green Software: Final Report of the Project: Establishing and Exploiting Potentials for Environmental Protection in Information and Communication Technology (Green IT)*; Report Commissioned by the Federal Environment Agency; Förderkennzeichen 3710 95 302/3; Federal Environment Agency: Berlin, Germany, 2015.

22. Penzenstadler, B. Towards a Definition of Sustainability in and for Software Engineering. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 1183–1185.
23. Taina, J. Good, Bad, and Beautiful Software—In Search of Green Software Quality Factors. In *Green ICT: Trends and Challenges*; Lopez-Lopez, J.-C., Sissa, G., Natvig, L., Eds.; e-journal created by CEPIS—Council of European Professional Informatics Societies: Saint Peter, Belgium, 2011; pp. 22–27.
24. Penzenstadler, B.; Raturi, A.; Richardson, D.; Tomlinson, B. Safety, security, now sustainability: The non-functional requirement for the 21st century 2014. *IEEE Softw.* **2014**, *31*, 40–47. [[CrossRef](#)]
25. Bozzelli, P.; Gu, Q.; Lago, P. *A Systematic Literature Review on Green Software Metrics*; VRIJE Universiteit Amsterdam: Amsterdam, The Netherlands, 2013.
26. Calero, C.; Moraga, M.; Bertoa, M.F. Towards a software product sustainability model. *arXiv* **2013**, arXiv:1309.1640.
27. Ahmad, R.; Baharom, F.; Hussain, A. A Systematic Literature Review on Sustainability Studies in Software Engineering. In Proceedings of the KMICe Knowledge Management International Conference (KMICe) 2014, Langkawi, Malaysia, 12–15 August 2014.
28. Betz, S.; Caporale, T. Sustainable Software System Engineering. In Proceedings of the 2014 IEEE Fourth International Conference on Big Data and Cloud Computing (BdCloud), Sydney, NSW, Australia, 3–5 December 2014; pp. 612–619.
29. Schmidt, B. Sustainability Knowledge about Software Parts in Software Engineering Processes. In Proceedings of the 4th International Conference on ICT for Sustainability (ICT4S 2016), Amsterdam, The Netherlands, 29 August–1 September 2016.
30. Dick, M.; Naumann, S. Enhancing Software Engineering Processes towards Sustainable Software Product Design. In *EnviroInfo 2010: Integration of Environmental Information in Europe, Proceedings of the 24th International Conference on Informatics for Environmental Protection, Cologne/Bonn, Germany, 6–8 October 2010*; Greve, K., Cremers, A.B., Eds.; Shaker: Aachen, Germany, 2010; pp. 706–715.
31. Calero, C.; Moraga, M.Á.; Bertoa, M.F.; Duboc, L. Green Software and Software Quality. In *Green in Software Engineering*; Calero, C., Piattini, M., Eds.; Springer: Berlin, Germany, 2015; pp. 231–260.
32. Hilty, L.M.; Naumann, S.; Maksimov, Y.; Kern, E.; Filler, A.; Guldner, A.; Gröger, J. Set of Criteria for Sustainable Software. Available online: <http://green-software-engineering.de/criteria-catalog> (accessed on 18 September 2018).
33. Yin, R.K. Case study research. In *Design and Methods*, 5th ed.; Sage: Los Angeles, CA, USA; London, UK; New Delhi, India; Singapore; Washington, DC, USA, 2009.
34. Eisenhardt, K.M. Building theories from case study research. *Acad. Manag. Rev.* **1989**, *14*, 532–550. [[CrossRef](#)]
35. Guldner, A.; Garling, M.; Morgen, M.; Naumann, S.; Kern, E.; Hilty, L.M. *Energy Consumption and Hardware Utilization of Standard Software: Methods and Measurements for Software Sustainability; From Science to Society*; Springer: Berlin, Germany, 2018; pp. 251–261.
36. Drangmeister, J.; Kern, E.; Dick, M.; Naumann, S.; Sparmann, G.; Guldner, A. Greening Software with Continuous Energy Efficiency Measurement. In Proceedings of the Workshop Umweltinformatik zwischen Nachhaltigkeit und Wandel, Koblenz, Germany, 19 September 2013; pp. 940–951.
37. Dick, M.; Kern, E.; Drangmeister, J.; Naumann, S.; Johann, T. Measurement and Rating of Software-induced Energy Consumption of Desktop PCs and Servers. In *Innovations in Sharing Environmental Observations and Information, Proceedings of the 25th International Conference EnviroInfo, Ispra, Italy, 5–7 October 2011*; Pillmann, W., Schade, S., Smits, P., Eds.; Shaker: Aachen, Germany, 2011; pp. 290–299.
38. Papadopoulos, G.; Rikama, S.; Alajöökö, P.; Salah-Eddine, Z.; Airaksinen, A.; Luomaranta, H. Statistics on Small and Medium-Sized Enterprises. Available online: http://ec.europa.eu/eurostat/statistics-explained/index.php/Statistics_on_small_and_medium-sized_enterprises#Further_Eurostat_information (accessed on 25 July 2018).
39. Johnson, M.; Halberstadt, J.; Schaltegger, S.; Viere, T. Software and Web-Based Tools for Sustainability Management in Micro-, Small-and Medium-Sized Enterprises. In *Advances and New Trends in Environmental and Energy Informatics*; Marx Gomez, J., Sonnenschein, M., Vogel, U., Winter, A., Rapp, B., Giesen, N., Eds.; Springer International Publishing: Cham, Germany, 2016; pp. 259–274.

40. Kern, E.; Silva, S.; Guldner, A. Assessing the Sustainability Performance of Sustainability Management Software—Replication Package. Available online: <https://doi.org/10.5281/zenodo.1344314> (accessed on 18 September 2018).
41. WeSustain GmbH. *WeSustain User Manual*; Version 14.08; WeSustain GmbH: Buxtehude, Germany, 2014.
42. Schien, D.; Coroama, V.C.; Hilty, L.M.; Preist, C. The Energy Intensity of the Internet: Edge and Core Networks. In *ICT Innovations for Sustainability: Advances in Intelligent Systems and Computing*; Hilty, L.M., Aebischer, B., Eds.; Springer: Basel, Switzerland, 2015; pp. 157–170.
43. Dick, M.; Kern, E.; Johann, T.; Naumann, S.; Gülden, C. Green Web Engineering—Measurements and Findings. In *EnviroInfo 2012: Man Environment Bauhaus: Light up the Ideas of Environmental Informatics, Proceedings of the 26th International Conference on Informatics on Informatics—Informatics for Environmental Protection, Sustainable Development and Risk Management; Federal Environment Agency, Dessau, Germany, 29–31 August 2012*, 1st ed.; Arndt, H.-K., Knetsch, G., Pillmann, W., Eds.; Shaker Verlag: Aachen, Germany, 2012; pp. 599–606.
44. Kreten, S.; Guldner, A.; Naumann, S. An Analysis of the Energy Consumption Behavior of Scaled, Containerized Web Apps. *Sustainability* **2018**, *10*, 2710. [[CrossRef](#)]
45. Chitchyan, R.; Becker, C.; Betz, S.; Duboc, L.; Penzenstadler, B.; Seyff, N.; Venters, C.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.
46. Dick, M.; Naumann, S.; Held, A. Green Web Engineering: A Set of Principles to Support the Development and Operation of “Green” Websites and their Utilization during a Website’s Life Cycle. In *WEBIST 2010, Proceedings of the Sixth International Conference on Web Information Systems and Technologies, Valencia, Spain, 7–10 April 2010*; Filipe, J., Cordeiro, J., Eds.; INSTICC Press: Setúbal, Portugal, 2010; Volume 1, pp. 48–55.



© 2018 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/>).