

## Green Measurements for Software Product Based on Sustainability Dimensions

Komeil Raisian<sup>1</sup>, Jamaiah Yahaya<sup>2,\*</sup> and Aziz Deraman<sup>3</sup>

<sup>1</sup>Department of Computer, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

<sup>2</sup>Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia

<sup>3</sup>Faculty of Ocean Technology Engineering & Informatics, Universiti Malaysia Terengganu, 21030, Terengganu, Malaysia

\*Corresponding Author: Jamaiah Yahaya. Email: jhy@ukm.edu.my

Received: 26 May 2021; Accepted: 30 June 2021

**Abstract:** Software is a central component in the modern world and vastly affects the environment's sustainability. The demand for energy and resource requirements is rising when producing hardware and software units. Literature study reveals that many studies focused on green hardware; however, limited efforts were made in the greenness of software products. Green software products are necessary to solve the issues and problems related to the long-term use of software, especially from a sustainability perspective. Without a proper mechanism for measuring the greenness of a particular software product executed in a specific environment, the mentioned benefits will not be attained. Currently, there are not enough works to address this problem, and the green status of software products is uncertain and unsure. This paper aims to identify the green measurements based on sustainable dimensions in a software product. The second objective is to reveal the relationships between the elements and measurements through empirical study. The study is conducted in two phases. The first phase is the theoretical phase, where the main components, measurements and practices that influence the sustainability of a software product are identified. The second phase is the empirical study that involved 103 respondents in Malaysia investigating current practices of green software in the industrial environment and further identifying the main sustainability dimensions and measurements and their impact on achieving green software products. This study has revealed seven green measurements of software product: Productivity, Usability, Cost Reduction, Employee Support, Energy Efficiency, Resource Efficiency and Tool Support. The relationships are statistically significant, with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationships were all accepted. The contributions of this study revolve around the research perspectives of the measurements to attain a green software product.

**Keywords:** Green software product; sustainability dimension; empirical study; green measurements



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

The software industry is now receiving challenges from the authority and public, in general, to incorporate and focus on the green aspect in software development and product. The software is expanding closure to our everyday lives and the environment. The sustainability effect on the environment is becoming increasingly vital. Green software necessities require consideration and a conventional quality aspect within software development and the product itself. To comprehend the concept of ‘green software’, it was defined as having a direct or indirect effect on society, human beings, the environment, and the economy. The outcome of development, deployment and software usage has a positive impact based on sustainable development [1]. Hence, green software exceptionally relies on having a green software development procedure and a reason for green software usage.

The definition of green software was introduced as “computer software that can be developed and used efficiently and effectively with minimal or no impact to the environment” [2]. The aim is to consider the green aspects in software product development, especially from the non-functional specification perspectives. This would ensure that final products are environmentally friendly. Therefore, green elements and measures of a software product must be investigated and well-defined.

A previous study has revealed that information and communications technology (ICT) signifies around 1.4% of the world’s CO<sub>2</sub> emissions. The 1.4% is only due to hardware and devices, network and data centres and enterprise network [3]. The software has taken a fundamental role due to individual businesses, public sectors, and end-client applications. The software can minimise power consumption by being more energy-efficient (i.e., by becoming greener); utilising lesser power or adopting more sustainable and supported procedures will diminish the environmental effects of software used by governments, organisations, and people.

Significant efforts were made to measure and organise the level of greenness within the hardware field. So far, a few studies have embarked on making greener software [4–7]. Previous studies have explored this aspect in the software development process but do not address a software product’s sustainability and green elements [8,9]. Despite the understanding and knowledge that sustainability can be achieved by applying and balancing all the three dimensions of sustainability and the vital sustainability of our lives [10], we still lack the balance sustainability measurements to attain software product with standard green fulfilment. The literature study has revealed that previous models were not fully covered with green measures needed to acquire the relevant values for software products to achieve a green standard and balance sustainability dimensions.

The remaining parts of this paper are organised as follows: Section 2 reviews the existing related works regarding green software product and software sustainability. Section 3 presents the method used in this study which includes a theoretical and empirical investigation. In Section 4, the analysis of data and results follows by a discussion in Section 5. Section 6 concludes this paper with a conclusion.

## 2 Literature Review

The sustainable product issue is fundamental according to software and hardware manufacturing organisations [11]. In recent years, creating software products and eco-friendly has become the target of software industries. The concepts of the eco-friendly target revolve around decreasing carbon utilisations, saving energy and minimising dangerous waste. The negative consequence of this inattention would be a lack of integrating technology into eco-friendly concepts, resulting in extreme implementations by associations without any positive effect on the environment. Penzenstadler et al. [12] presented a broadened assortment of information for sustainability which covered software engineering and management. They discovered that there were inadequate works on sustainable software engineering compared to other domains.

Studies in green software were continuously carried out. However, only a few studies focused on developing green and sustainable software products, and it remains inadequate. Some studies focused on creating software tools that assess the impact of software on the environment and its impact on improving the natural environment based on energy efficiency. The other studies underlined the working framework of handling the applications of power consumption [13,14]. Some software clarifications incorporate practical algorithms by composing a minimal outline of codes, data constructions, virtualisation, and shutting down applications that are no longer in use in a specific environment was being studied and investigated. Fine-grained green computing was proposed to create energy allocation algorithms for routing data and diminish parallelism overhead by improving burden adjusting algorithms. A reference model named Green Software (GREENSOFT) for sustainable software was further proposed [1]. This model was a four-section model supported by software engineers, developers, managers, and customers to produce, maintain, and utilise the software for a greenness approach. Furthermore, the primary sub-factors of the green element and a new model of software product quality based on the green concept was proposed [15].

Concentrating on improving an energy efficiency assessment for software is a possible approach. Still, it can hinder impact due to the extra code lines they include [16]. Thus, Sharma et al. presented techniques to assess the energy utilisation while running the software in each time frame and consolidated with the operating systems [17]. Alsayyah et al. [18] continued this similar work in cloud computing. Reference [19] exhibited a methodology considering periodic assessments of Green Performance Indicators system (GPIs), Quality of Service (QoS) and reception of Service-Oriented design, which can be applied to streamline energy efficiency of software services. References [20,21] further focused on incorporating sustainability in service-oriented software.

The factors for assessing the carbon impression of software improvement, measuring the resources utilised via software and the amount of harm they cause to the environment were studied [22]. Another study focused on quality-related aspects by considering green software estimations based on quality attributes [23]. The ISO standard suggested that the green ability aspects of the software must be regarded as to accomplish high-quality software [15]; otherwise, it would fail to fulfil the standard since software sustainability is gaining increasing significance in the industry and society. While sustainability is a standardised practice in many engineering disciplines, there is a limited study on awareness in the software engineering community. Similarly, it revealed the significance of paying more attention to sustainability measurements from the perspective of software products to achieve green product [24,25].

Green software is considered a beneficial solution to solve the problems caused by and associated with the long-term use of software, especially from a sustainable perspective. Without a mechanism for measuring the greenness of a particular software product as it functions in a specific environment, the mentioned benefits will not be attained or guaranteed. The literature study shows that there is still a limitation of research that focuses on green software products. The current models by previous studies related to green software products do not address the mentioned problem.

## ***2.1 Green Software Product***

Generally, green software is characterised as software that poses indirect and direct positive effects to the society, economy, and environment from the development phase till the operation phase. The previous researchers limited the definition and characterised green software as the software that concentrates most on environmental necessities alone without considering other factors that influence the green aspect.

There are differences in terminology between green by software and green in software [26,27]. Green in software is defined as how to make software based on environmentally sustainable methods. Creating minimal energy consumption and environmental waste is conceivable in software development's entire

operation and life cycle. Instead, [26] characterised green by software as covering the software utilisation and perspectives to deliver minimum energy consumption and environmental waste as is conceivable by computing methods and techniques. Currently, most of the studies related to green IT is associated with the hardware domain, which concentrates, for the most part, on enhancing energy efficiency due to hardware. It is essential, but the software is also crucial, and one has no significance without the other.

Green software is software that creates less waste as is conceivable between its development and operation. References [10–12] stipulated that green software engineering is the skill of developing a green software process and product based on a defined standard. It is a skill to design and build software products so that awful and constructive effects on the development process from its entire life cycle are considered and taken care of. The green aspects are continuously assessed throughout the life cycle and documented for maximum optimisation of the product.

## **2.2 The Software Sustainability**

Relatively few studies have fingered the sustainability concept in software process and product. It portrays the needs of handling sustainability in the software process and product context. References [12,28] mentioned that three sustainability dimensions are applied to investigate sustainable software frameworks: social, economic, and environmental. The most well-known description of sustainability in software engineering is software that has no explicit and implicit destructive effects on the economy, society, human beings, and environment that result from the development, deployment, and operation of the software.

### *2.2.1 Environmental Dimension*

Environmental sustainability is required via human activity, and it is to improve the quality of life in human being by securing natural resources that include water, air, minerals etc. The development of software product requires investigating energy efficiency outcomes in hardware issues and their parts to accomplish environmental sustainability. A primary investigation is expected to create hardware energy consumption of its performing software. While the software assumes an expanding position in helping the public, its energy efficiency and environmental effect are more vital.

#### *a) Energy efficiency*

Researchers have taken an outward look at energy efficiency elements and states of the CPU. Although most energy-saving aspects are clear to applications and software developers, they still have short, direct control. However, energy-saving attributes created into the platform do influence the primary behaviour of the software, making it effective. Good behaviour software allows energy-saving characteristics to work. Weak behaviour software hinders energy-saving features and leads to lower battery life and more energy costs [29]. Efficiency is also related to the software's behaviour in saving resources and avoiding waste during development and usage. Factors related to the environment's impact include electricity, supply of power, emissions, and consumed material. Organisations are encouraged to have more green strategies in reducing energy costs which later will contribute towards global environmental goals [22]. Whether the software is achieving environmental sustainability or not, energy efficiency is one of the most effective direct measurements to reduce energy growth in its demand and measure [30]. It can take more than one metric in the software system to measure energy efficiency because software consists of different modules with diverse purposes. However, software developers must gather the exact requirements of the software and be confident that the developed software is sufficient to provide user needs without burdening the environment's sustainability.

### *b) Resource efficiency*

The energy consumption is associated with the behaviour of the software running time [31]. These are the usual and primary factors for all manners to evaluate resource efficiency, including CPU usage, memory usage, storage usage, and I/O usage [32]. It is also related to the product and its activity in design, development, maintenance, management, operation, and removal in software engineering. Resource efficiency of the software is associated with the resources used during execution and the platform's usage. It is the relationship between the use and the application required to produce it. Implementing web-based application software has the potential for resource efficiency. The development can enable end-user equipment with low processing spaces and storage, which is attractive. Operations requiring huge processing spaces can be performed on web servers without loading the web client, and this method is typically based on the cloud service [33]. Resources saving can be accomplished by applying a networking environment such as using Local Area Network (LAN) and Wireless Local Area Network (WLAN) instead of the cellular network to gain internet access [33].

The quality of the program codes is of additional interest in resource efficiency. It is reasonable to calculate complex code by analysing the source code and repairing the relationships of components [34]. With these parameters, software defects can also be detected to prevent the probability of software failure occurrences [35]. These metrics are related to the measurements of maintainability in quality factor. Still, they can also support prediction for the probability of defect occurrences in a particular part of the software [36].

Similarly, by analysing the program source code, the maintainability factor can be monitored through compliance with the professional coding convention. Indirectly, this will affect the maintenance activity during software evolution. Thus, software quality and green measurements are integrated.

### *2.2.2 Social Dimension*

The social dimension means keeping up with social capital and protecting the unity and harmony of the societal groups. It is characterised as “a positive and long-term condition inside groups and a procedure inside groups that can accomplish and keep up that condition” [37]. Social sustainability is also associated with how it is measured and how the social sustainability of a software framework is currently assessed [38]. Reference [38] surveyed some common factors in social sustainability in general. Some potential social sustainability factors in software are employment, health, education, security, culture etc. From the theoretical study, the measurements of social sustainability in green software products are as follows:

#### *a) Tool support*

Tool support is defined as providing tools supporting development activities and meeting user needs for a useful software product [39]. Developers use tools to assist essential activities in software development. Automating the analysis, design, implementation, and maintenance of compressed software products are supported by CASE (Multitude of Computer-Aided Software Engineering) tools. However, the challenge is how to reasonably use the tools to help organisation business goals and the technical requirements of product [40].

Establishing tool support for a product line involves identifying needs, selection, evaluation, insertion, measurement, and maintenance. The corporate ability of a chosen collection of tools is fundamental to automate production in a software product line [41]. Tool support partners with various skills for embedding sustainable and green techniques in developing, administrating, or using software products are needed [42]. However, today, fast-growing options related to the number of approaches and tools to build sustainability are almost conflicting or competing [43]. Thus, a systematic approach containing

sustainable fundamental principles is needed to show that these tools are an additional advantage and can support software development to ensure sustainability.

#### *b) Employee support*

Employee Support means supporting learning new appliances with advice provided to use the software product correctly [38]. It also means helping and aiding employees to use new software tools for excellent software development [41]. It might be a solid and influential matter in sustainability since every employee can become a hidden green champion. It has revealed that employees are among the leading and secret factors to successful social sustainability that can be effective for an organisation [44]. Enhancing employee engagement in greening businesses has various social and monetary advantages. In some conditions, trainees or fresh employees from academic backgrounds are the most capable and experienced people for a project, even though having less experience in an industrial environment. If any, there are fewer conditions for the academic institution to aid educated and experienced industrial engineers and technologists in green software process and product [45]. Keeping employees involved, cheerful, and productive has been the most effective approach for advancing green organisations and companies [46]. Green members are self-organised, popular, and interoperable with other employees, and they always voluntarily gather to train and educate. They are active and robust employees through sustainability. Since 'green' has become essential, an increasing number of employees prefer to work with a company devoted to the sustainability aspect in a work setting where green practices valued are being implemented, such as work from home and flexible working hours. Employee functions of an organisation can be instrumental in encouraging a thorough approach to providing a culture of environmental supervision and sustainability. The importance of this view in sustainable software development can develop a significant role in organisation systems. Various cases exhibit how emphasising each employee system has supported organisations to create a sustainable culture, successfully leading to financial sustainability [47].

#### *2.2.3 Economic Dimension*

The profit is the sum a person spends during a period and still does perfectly good towards the end. It is also known as economic. In many situations, the extent of economic sustainability exhibit capital conditions [48]. The value of resources and benefits in cash conditions is standard, as is recognising the measure of return produced via effective employment of these resources and assets. In the software product improvement viewpoint, economic sustainability is the capacity of a business to control its operation effectively [49]. Depending on the business's intention, the model is supported if an organisation is concerned with it, maintaining it, and making it relevant for whatever length of time to help business benefits. Consequently, the related terms are efficiency and productivity. The main measurements of economic sustainability in green software products are as follows:

#### *a) Productivity*

Productivity is a measurement within accepted economic ways. Productivity relates to the rate of software services produced, the number of goods, and the labour expense required in the development process [50]. Based on the economic term, Productivity is the proportion between the measure of products or administrations created and the labour or cost of delivering them [51]. The study assumed that Productivity is the proportion between the effort of software designed for the work and the cost of producing it. To define software productivity, it relates to the meaning of software which at its most major level, the software is a program involving lines of code. In any case, lines of code, all by themselves, are not the essential deliverables of a software venture, and clients regularly do not know how many lines of code are in the software they are purchasing [51]. Reference [52] indicated that productivity is the measure and a criterion of the generation of execution.

Similarly, productivity is the rate of output per unit of input [53]. Information is the measure of exertion we spend on the venture to convey the software, but there is no direct estimation for output measurement. Nonetheless, one could likewise observe the software product as a conveyance of usefulness or even express it as far as the quality properties it meets. Every one of these estimations attempts to measure the extent of the conveyed software product.

Furthermore, a study has revealed that the key to economic sustainability development is an expansion in productivity. Reference [54] considered that the total productivity criteria are progressively perceived and used to rebuild advantages and ongoing business process development. The productivity origination implies that it can indeed be seen as the output of per unit input or efficiency used with resources [55].

#### *b) Cost efficiency*

Cost reduction is the measurement that aims to assess and estimate costs of green approach applications and the requirements for software product at any level of activity [56]. Previous researchers agreed that several measurements of economic sustainability in software are associated with warming and lighting office space, supporting staff, administration and interchanges, security, and other advantages such as benefits and medical coverage, and other principal offices such as a library or recreational centres [57,58]. They belong in the cost of software sustainability and the price is the primary concern in the measurement. Thus, cost efficiency is a significant metric to measure software economics. Ajila et al. [49] suggested that the most crucial element of cost-efficiency in software development is adopting Open Source Software (OSS). This also applies to software as a component and product to ensure green and sustainability.

#### *c) Usability*

Usability is a measurement for having ease of learning and using the operating software by the users [59]. In terms of usability in economic sustainability, making software usable establishes durable, modifiable, and reusable software. The mentioned criteria need to be met to produce software that can be maintained and sustainably in the future. It can be achieved since it can adapt or change without losing its usefulness, functionality, and other quality characteristics [5,15,60]. Usability is how the users can use much software to accomplish quantified goals with viability, efficiency, and satisfaction in a measured content of utilisation. Besides, usability measures an item's capability for completing users' objectives [61]. A software system is significant in the future; any system should not be deemed usable if not in a sustainable manner. Economic sustainability and the trust among clients and systems would diminish and decrease if systems have usability problems [62]. It reveals that usability is a critical factor in economic sustainability concerning system applications, clients, and outside stakeholders.

Previous studies have revealed that sustainability can be achieved when balanced dimensions are covered. However, there is still a lack of suitable measurements to achieve green software product in current studies. The literature study revealed that previous models were not fully covered with green elements needed to acquire the relevant values for green products based on a certain standard. [Tab. 1](#) summarises the measurements identified from previous studies. Even though there are several works on sustainability measures in a software product. Still, most of the results do not cover balanced dimensions as needed and suggested from previous study except studies done by Condori-Fernandez et al. [63]. They focus on the quality requirement for software (see [Tab. 1](#)). Therefore, our research focuses on comprehensive green elements based on sustainability dimension to measure the green software product.

**Table 1:** Sustainability measures in software product by previous studies

Researchers	Environmental		Economical			Social	
	Resource efficiency	Energy efficiency	Cost reduction	Productivity	Usability	Employee support	Tool support
Naumann et al. [1]	×	×					×
Calero et al. [5]	×	×			×	×	
Dick et al. [11]						×	×
Taina [22]		×					
Penzenstadler [23]	×	×	×				
Abdullah et al. [30]		×					
Koçak et al. [32]	×						
Taina et al. [31]	×						
Hilty et al. [33]	×	×					
Chidamber et al. [34]	×						
Beuche et al. [40]			×	×	×	×	×
Becker et al. [46]						×	
Liebowitz [47]						×	
Ajila et al. [49]				×			
Hammer et al. [54]				×			
Amri et al. [58]			×				
Condori-Fernandez et al. [63]	×	×	×	×	×	×	×
Al Hinai [64]						×	×

### 3 The Empirical Study

The scope of this study relies on green measurements that focus on software products without considering the development process issues. Based on the literature analysis shown in Tab. 1, the relevant measures for a green software product are proposed. The hypotheses are that there are significant relationships between sustainable dimensions and measurements for green software. The dimensions are social, environmental, and economical, while the measurements are productivity, cost reduction, usability, employee support, tool support, energy efficiency and resource efficiency.

A pilot study and content validation were conducted before the actual survey. The pilot study aimed to test the questions' content, identify ambiguous questions, and test responses obtained from a small, selected sample of the respondents. Findings from the pilot study and content validation process enhanced the instrument via improving the questions to be more coherent or readable, accurate, presentable, justifiable, and relevant. Moreover, the time taken to answer the questionnaire was identified too.

#### 3.1 Data Collection

The sampling technique for this study is purposive and random sampling. A list of possible respondents from private and government agencies related to the software industry was identified. The criteria used for



agency selection of this study are software development or the companies that acquire and use the software in their operations. The respondents are in the group of software developers, administrators, IT practitioners and users. They were contacted and invited to participate in the survey; from December 2016 until February 2017, 220 respondents and 148 (or 67%) returned the questionnaires. A total of 102 questionnaires, or 69% of respondents, were deemed valid answer, which is adequate and acceptable [65,66]. The collection methods included manual, face-to-face meetings, hard mails, and an online survey. However, most respondents preferred to answer the questionnaires by post or online rather than face-to-face. At the same time, an online questionnaire was designed, sent to respondents, and posted for three months. The online survey collected higher feedback than face-to-face meetings; thus, it showed that online form is the most popular data collection method, safe and reduced costs and time.

### 3.2 The Relationship of Factors

The Pearson correlation test was carried out to assess the magnitude and direction of the variables [67], and the correlation threshold was identified based on [68] (see Tab. 2). It has already been stated that the research hypotheses tests are carried out when a preconditioning test for correlation and regression are satisfied. Nevertheless, multiple regression analysis is used to calculate the strength of the variables. Besides, hierarchical multiple regression analysis tests are used to reconcile the impact between variables. The Pearson correlation coefficient measures the strength of a linear association between two variables and is denoted by  $r$ . Based on previous studies, the  $r$ -value  $\geq 0.5$  is considered as a significant and robust relationship. The  $r$ -value of each computational hypotheses is used to determine the robustness of the relationships. The Pearson correlation coefficient,  $r$ , indicates how far away all the data points are to the best fit of the model.

**Table 2:** Cohen's guideline for correlation strength [68]

Value of “r”	Strength
Weak	0.10–0.29
Medium	0.30–0.49
Strong	0.50–1.00

The correlation coefficient formula (1) is normally used to compute the  $r$ -value using statistical package.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \tag{1}$$

### 4 Analysis and Results

During the analysis, the mean value for each dimension and measurements were computed. The 5-point numerical scale was used in the questions of this survey, which ranges from Not Important to Most Important. The scales were then mapped into equal intervals and were calculated using formula (2).

$$\text{Interval ranges} = (n - 1)/n; \tag{2}$$

where  $n$  is the maximum number in the used scale, which is equal to 5. The  $n = 5$  value refers to the 5-point

scale used in the questionnaire and suggested by [69]. Thus, the interval size of the consideration level is 0.8. Tab. 3 shows the interval values.

**Table 3:** Interval values

Degree of importance (DI)	Interval values
Not important (NI)	1.00–1.80
Less important (LI)	1.81–2.60
Neutral (N)	2.61–3.40
Important (I)	3.41–4.20
Most important (MI)	4.21–5.00

#### 4.1 Sustainable Dimensions and Measurements in Software Product

This section analysed the data captured from our respondents on the importance of software sustainability dimensions and the measurements in a green software product. The respondents were asked about sustainable software practices required to achieve high green software products categorised into sustainable dimensions: environmental, economic, and social. The measurements were identified from the literature as discussed in the previous section. The measurements are productivity, cost reduction, usability, employee support, tool support, energy efficiency and resource efficiency. Tabs. 4–6 shows the results obtained by calculating the mean score gained by each practice of sustainable dimensions. It demonstrates that most sustainable practices had obtained high consideration, whereby the mean values are in the range of Important to Most Important and Neutral based on the Interval Values.

**Table 4:** Interval values of social sustainability dimensions practices by respondents

Dimensions	Practice	Mean	Degree of importance
Social sustainability	Employee support, cohesiveness and shifting patterns of teamwork are relevant to software product variables.	4.26	Most important
	Software scoping practices are supported by tools that collect and show the expected and varying features of products.	4.00	Important
	The tool supports the organisation's structure, which is intended for the development process of the software product.	3.19	Neutral
	Tools and employee are needed in software product requirement engineering to represent the requirements for products.	4.29	Most important
	Tools are needed in the production plan practices used to create strategies and methods that support the development of the products.	4.00	Important
	Tools are required to develop and test components to check the consistency with standard and varying features.	4.25	Most important
	Tools are required in the product architecture to portray the product development without violating the defined scope.	2.91	Neutral
	Employ development aids that help project coordination.	4.11	Important

**Table 5:** Interval values of economic sustainability dimensions practices by respondents

Dimensions	Practice	Mean	Degree of importance
Economical sustainability	The main goal of the software development model lies in reducing the business cost.	4.29	Most important
	High productivity and high product quality results could be traced back to observable patterns of teamwork.	4.28	Most important
	I think that I would like to use this software frequently for Usability.	3.24	Neutral
	I think Usability is unnecessarily complex.	4.01	Important
	I think the software is easy to use by the usability development.	3.94	Important
	I find that the usability development well integrates various functions of this software.	4.16	Important
	I need to learn many things before I can get going with this software by the usability development.	3.96	Important
	I feel very confident using the software by the usability development.	3.22	Neutral
	I imagine that most people will learn to use this software very quickly through usability development.	4.19	Important
I think inconsistency in this software by usability development.	3.90	Important	

**Table 6:** Interval values of environmental sustainability dimensions practices by respondents

Dimensions	Practice	Mean	Degree of importance
Environmental sustainability	Software performance depends on the primary computing resources like the CPU and the memory.	4.19	Important
	Software is required for faster CPUs and increased memory.	3.99	Important
	Memory usage is one of the main requirements of software development based on environmental sustainability.	4.31	Most important
	The size of the total storage space for each system is an essential concern in software.	4.25	Most important
	Separate short-term/long-term storage systems are a requirement for software development.	4.20	Important
	Energy consumption has a direct impact on resources during the production and use of the software.	2.87	Neutral
	How significant is functional CO <sub>2</sub> emission concerning environmental sustainability?	4.20	Important
	How important is functional suitability concerning resource efficiency?	2.93	Neutral

For sustainable practices, the results were obtained by calculating the mean score gained by each dimension. The analysis demonstrates that most sustainable software dimensions had obtained high consideration, whereby the mean values are in the range of Important, as shown in [Tab. 7](#).

**Table 7:** Interval values of sustainable dimensions in organisation from respondents

Dimensions	Mean	Degree of importance
Social	3.88	Important
Economical	3.92	Important
Environmental	3.87	Important

In the survey, the respondents were asked to rank their level of consideration on the listed sustainable measurements in software products. [Tab. 8](#) shows the interval values obtained by calculating the mean score of each measure. It shows that most of the measures had received high consideration, whereby the mean values are in the range of Important to Most Important.

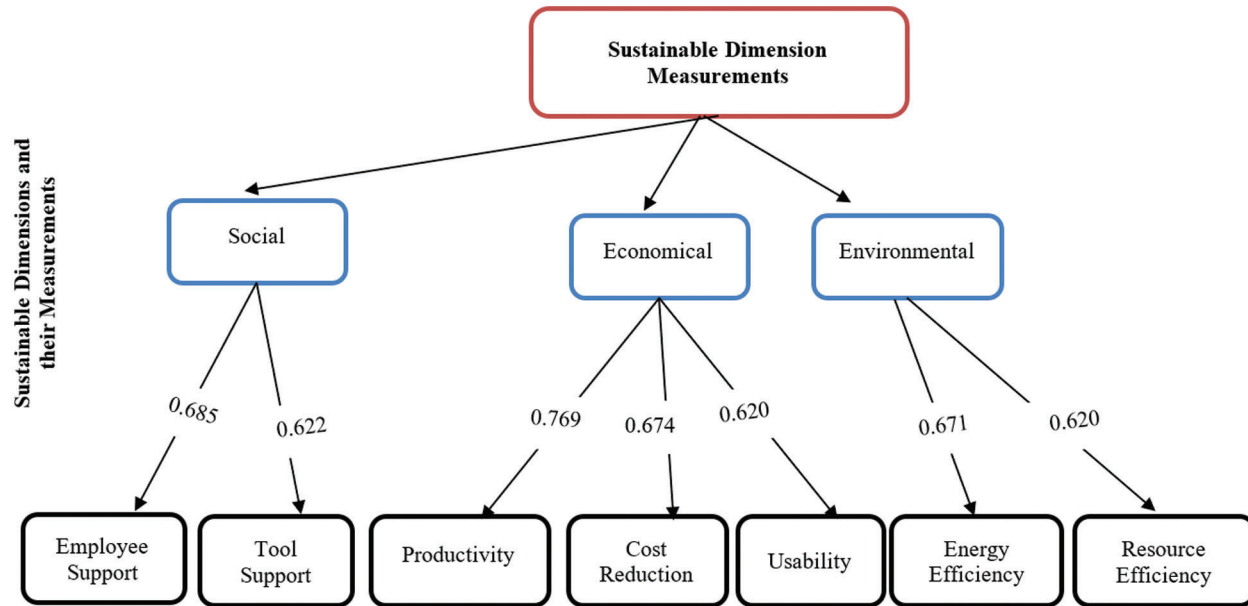
**Table 8:** Interval values of software measurements

Measurements	Description	Mean	Degree of importance
Productivity	In standard economic terms, Productivity is the ratio between the number of goods or services produced and the labour or expense that goes into making them.	4.09	Important
Usability	I have ease of use and learnability of a human-made object such as a tool or device of the software.	4.18	Important
Cost reduction	This metric aims to evaluate or estimate the costs of green policies application and software production processes in the industry at any stage.	4.17	Important
Employee support	Help employees learn to use new tools for software.	4.38	Most important
Tool support	Provide tools that support the work process and tools to convert customer requirements into a valuable product.	4.37	Most important
Resource efficiency	The energy consumption by software relates to the nature of the application and the system configuration of the run time environment.	4.22	Most important
Energy efficiency	Metrics related to the impact of the application on the environment, considering electricity, power supply, consumed material and emissions.	4.19	Important

#### 4.2 The Relationship of Factors

The  $r$ -values of the variables are computed applying the formula (1) using a statistical package. It reveals the results as shown in [Fig. 1](#). The results indicate that Social Sustainability for a software product is positively associated with Employee Support (Relationship = 0.685). The correlation matrix demonstrates

that all hypothesised relationships are recognised at  $p < .01$  degree. The association is considered statistically significant with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationship is accepted. The relationship between Social Sustainability and Tool Support is also high (0.622) and therefore, shows that Social Sustainability is positively connected to Tool Support ( $r = 0.622$ ).



**Figure 1:** The relationship of sustainable dimensions and measurements for green software

Another correlation analysis based on the probability of occurrence had demonstrated a strong correlation between Economic Sustainability and Cost Reduction (Relationship = 0.674). The result indicates that Cost Reduction is positively connected to Economic Sustainability ( $r = 0.674$ ). The correlation matrix demonstrates that all hypothesised relationships are recognised at  $p < .01$  degree. The association is considered statistically significant with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationship is accepted. Accordingly, 0.620 is a strong relationship between Usability and Economic Sustainability. It indicates that Economic Sustainability within the software product sector is positively connected to Usability (Relationship = 0.620) as shown in Fig. 1. Similarly, the correlation matrix shows the relationship is statistically significant with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationship is also accepted. At the same time, the relationship between Economic Sustainability and Productivity is discovered as equal to 0.769. This result indicates that Economic Sustainability is positively connected to Productivity ( $r = 0.769$ ). The relationship is statistically significant with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationship of the research is accepted.

The correlation analysis based on Environmental Sustainability, Energy Efficiency and Resource Efficiency was conducted and discovered that the  $r$ -value for Energy Efficiency is 0.671, which indicate a strong relationship. At the same time, the correlation between Environmental Sustainability and Resource Efficiency indicates a strong association with the  $r$ -value equal to 0.620. The correlation matrix demonstrates that all hypothesised relationships can be recognised at  $p < .01$  degree. The relationships are statistically significant, with a significance level of less than 0.01 ( $p = 0.000$ ). Thus, the hypothesised relationships are all accepted.

## 5 Discussion

This paper has presented the literature review, mentioning the current activities for a green software product, software sustainability and related issues. The explanation began with an overview of green software, and the strategies and perspectives in green software were defined. Further details were provided with state-of-the-art topics and the relationships between the sustainability dimension and green elements. Limitations and gaps determined in the literature were also provided and explained.

Furthermore, this study was broadened into investigating the green elements and their relationships through empirical research. The empirical analysis reveals that the green aspects are associated with sustainable dimensions: social, economic and environmental. As discussed in Section 5, the three dimensions are broken down into seven primary measurements associated with green elements: productivity, cost reduction, usability, employee support, tool support, energy efficiency, and resource efficiency. The study discloses the relationships between the dimensions and measurements. It shows that the measures influence the green of software products verified by the respondents of this study and the literature review.

Additionally, the specifications of the general population associated with software products and the necessities of green software in the software industry have been uncovered. It was found that most respondents did not perform any mechanism for measuring and assessing green software products in their organisations. Without a suitable green measurement model, the greenness of software products cannot be accomplished. Thus, a mechanism to evaluate and assess the critical green measurements for software products is required. We also investigated the green software process and the current practices from the industry and were analysed as discussed in [70]. The aim was to understand the potential factors for improving the recent trend of sustainability dimension towards green software life-cycle and process. Based on the findings, the green assessment model will be developed and applicable for measuring the greenness of software from both the process and product's perspective.

## 6 Conclusion

Green measurements discovered offer a fundamental contribution in solving the problem of proposing a novel model to achieve green software products. This study has revealed seven measures for a green software product, and they are associated with the three-sustainability dimension as discussed in this paper. The measurements were identified through literature review, and further was verified through an empirical study that involved 103 respondents from private and government agencies in Malaysia. They were invited to participate and confirm the measurements for green software from a sustainability's perspective.

For future work, a green software model (GSM) that includes the assessment process and criteria is needed to guarantee adequate business conditions in software development. It provides a mechanism to assess the green ability of software from a business perspective which might have a specific priority in the processes. Similarly, a guideline and actions of software quality factors are suggested to favour the greenness and sustainability of software products in their operating environment [71]. It can further be improved by evaluating the software process and quality and integrating green measurements through each phase of the software development life cycle. Additionally, the green measures can have different weightage in the assessment since they are different level of priority by the users. Therefore, further work can be carried out to determine the weightage of the elements and embeds them in the proposed GSM.

**Acknowledgement:** The authors would like to express appreciation to the Faculty of Information Science and Technology, UKM, to provide research resources to encourage research activities among faculty members.

**Funding Statement:** This research is funded by the Malaysia Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UKM/02/1).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] S. Naumann, M. Dick, E. Kern and T. Johann, “The greensoft model: A reference model for green and sustainable software and its engineering,” *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 294–304, 2011.
- [2] E. Kern, M. Dick, S. Naumann and A. Filler, “Labelling sustainable software products and websites: Ideas, approaches, and challenges,” in *Proc. 29th Int. Conf. on Informatics for Environmental Protection (EnviroInfo 2015), Third Int. Conf. on ICT for Sustainability (ICT4S 2015)*, Copenhagen, Denmark, pp. 82–91, 2015.
- [3] J. Malmudin and D. Lundén, “The energy and carbon footprint of the global ICT and E&M sectors 2010–2015,” *Sustainability*, vol. 10, 3027, pp. 1–31, 2018.
- [4] E. Kern, M. Dick, S. Naumann, A. Guldner and T. Johann, “Green software and green software engineering—definitions, measurements, and quality aspects,” in *Proc. CT4S 2013, The First Int. Conf. on Information and Communication Technologies for Sustainability*, ETH Zurich, pp. 87–94, 2013.
- [5] C. Calero, M. A. Moraga, M. F. Bertoa and L. Duboc, “Green software and software quality,” in *Green in Software Engineering*, Cham: Springer International Publishing, pp. 231–260, 2015.
- [6] S. Betz and T. Caporale, “Sustainable software system engineering,” in *Proc. Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth Int. Conf. on*, Sydney, NSW, Australia, pp. 612–619, 2014.
- [7] S. S. Mahmoud and I. Ahmad, “Green performance indicators for energy-aware it systems: Survey and assessment,” *Journal of Green Engineering*, vol. 3, no. 1, pp. 33–69, 2012.
- [8] S. Agarwal, A. Nath and D. Chowdhury, “Sustainable approaches and good practices in green software engineering,” *International Journal of Research and Reviews in Computer Science*, vol. 3, no. 1, pp. 1425–1428, 2012.
- [9] S. R. Ahmad Ibrahim, J. Yahaya, H. Sallehudin and A. Deraman, “Green software process assessment: The theoretical framework,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 3, pp. 2011–2016, 2021.
- [10] C. Meschede and M. Henkel, “Library and information science and sustainable development: A structured literature review,” *Journal of Documentation*, vol. 75, no. 6, pp. 1356–1369, 2019.
- [11] M. Dick, S. Naumann and N. Kuhn, “A model and selected instance of green and sustainable software,” In: Berleur, J., Hercheui, M. D., Hilty, L. M. (Eds.), *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. HCC 2010, CIP 2010. IFIP Advances in Information and Communication Technology*, vol. 328, Berlin, Heidelberg: Springer, pp. 248–259, 2010.
- [12] B. Penzenstadler, A. Raturi, D. Richardson, C. Calero, H. Femmer *et al.*, “Systematic mapping study on software engineering for sustainability (SE4S),” in *Proc. the 18th Int. Conf. on Evaluation and Assessment in Software Engineering*, London, England, United Kingdom, pp. 1–14, 2014.
- [13] G. Appasami and K. Suresh Joseph, “Optimisation of operating systems towards green computing,” *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 2, no. 3, pp. 39–51, 2011.
- [14] C. T. D. Lo and K. Qian, “Green computing methodology for next-generation computing scientists,” in *Proc. Computer Software and Applications Conf. (COMPSAC), 2010 IEEE 34th Annual*, Seoul, South Korea, pp. 250–251, 2010.
- [15] C. Calero, M. Moraga and M. F. Bertoa, “Towards a software product sustainability model,” arXiv preprint arXiv:1309.1640. 2013. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1309/1309.1640.pdf>.

- [16] E. Capra, C. Francalanci and S. A. Slaughter, "Is software "green"? application development environments and energy efficiency in open-source applications," *Information and Software Technology*, vol. 54, no. 1, pp. 60–71, 2012.
- [17] S. K. Sharma, P. K. Gupta and R. Malekian, "Energy efficient software development life cycle-an approach towards smart computing," in *Proc. 2015 IEEE Int. Conf. on Computer Graphics, Vision and Information Security (CGVIS)*, Bhubaneswar, India, pp. 1–5, 2015.
- [18] A. A. Alsayyah and S. Ahmed, "Energy-efficient software development techniques for cloud-based applications," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 5, pp. 8043–8053, 2020.
- [19] F. G. A. De Oliveira Jr. and T. Ledoux, "Self-optimisation of the energy footprint in service-oriented architectures," in *Proc. the 1st Workshop on Green Computing*, Bangalore, India, pp. 4–9, 2010.
- [20] M. Razavian, D. A. Tamburri, Q. Gu and P. Lago, "Modeling to support communication and engineering of service-oriented software," in *Proc. the First Int. Workshop on European Software Services and Systems Research: Results and Challenges*, Zurich, Switzerland, pp. 8–9, 2012.
- [21] B. Penzenstadler, V. Bauer and A. Fleischmann, "*Seminar: Sustainability in Software Engineering*," Technical Report, Technische Universit at Munchen, 2011.
- [22] J. Taina, "Good, bad, and beautiful software-in search of green software quality factors," *Cepis Upgrade*, vol. 12, no. 4, pp. 22–27, 2011.
- [23] B. Penzenstadler, "Towards a definition of sustainability in and for software engineering," in *Proc. the 28th Annual ACM Symposium on Applied Computing*, Coimbra Portugal, pp. 1183–1185, 2013.
- [24] T. Johann, M. Dick, E. Kern and S. Naumann, "Sustainable development, sustainable software, and sustainable software engineering: An integrated approach," in *Proc. Humanities, Science & Engineering Research (SHUSER), 2011 Int. Symposium on*, Kuala Lumpur, Malaysia, pp. 34–39, 2011.
- [25] R. Ahmad, A. Hussain and F. Baharom, "Software sustainability characteristic for software development towards long living software," *WSEAS Transactions on Business and Economics*, vol. 15, pp. 55–72, 2018.
- [26] K. Erdelyi, "Special factors of development of green software supporting eco-sustainability," in *Proc. Intelligent Systems and Informatics (SISY), 2013 IEEE 11th Int. Symposium on*, Subotica, Serbia, pp. 337–340, 2013.
- [27] NEEF. "The engaged organisation corporate employee environmental education survey and case study findings," The National Environmental Education Foundation Report, 2009.
- [28] C. R. Carter and P. Liane Easton, "Sustainable supply chain management: Evolution and future directions," *International Journal of Physical Distribution & Logistics Management*, vol. 41, no. 1, pp. 46–62, 2011.
- [29] B. Steigerwald and A. Agrawal, "Developing green software," *Intel White Paper*, vol. 9, pp. 1–11, 2011.
- [30] R. Abdullah, S. Abdullah, J. Din and M. Tee, "A systematic literature review of green software development in collaborative knowledge management environment," *International Journal of Advanced Computer Technology (IJACT)*, vol. 9, pp. 136, 2015.
- [31] J. Taina and S. Mäkinen, "Green Software Quality Factors," In: Calero, C., Piattini, M. (Eds.), *Green in Software Engineering*, Cham: Springer, pp. 129–154, 2015.
- [32] S. A. Koçak, G. I. Alptekin and A. B. Bener, "Integrating environmental sustainability in software product quality," in *Proc. RE4SuSy@ RE*, Ottawa, Canada, pp. 17–24, 2015.
- [33] L. Hilty, W. Lohmann and E. Huang, "Sustainability and ICT—An overview of the field," *Politeia*, vol. 27, no. 104, pp. 13–28, 2011.
- [34] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object-oriented design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476–493, 1994.
- [35] L. C. Briand, J. Wüst, J. W. Daly and D. V. Porter, "Exploring the relationships between design measures and software quality in object-oriented systems," *Journal of Systems and Software*, vol. 51, no. 3, pp. 245–273, 2000.
- [36] T. J. McCabe, "A complexity measures," *IEEE Transactions on Software Engineering*, vol. 4, pp. 308–320, 1976.
- [37] P. Willis, S. McKenzie and R. Harris, "Introduction: Challenges in adult and vocational education for social sustainability," in *Rethinking Work and Learning*, Netherlands: Springer, pp. 1–9, 2009.



- [38] M. Al Hinai and R. Chitchyan, "Building social sustainability into software: Case of equality," in *Proc. 2015 IEEE Fifth Int. Workshop in Requirements Patterns (RePa)*, Ottawa, ON, Canada, pp. 32–38, 2015.
- [39] R. Chitchyan, J. Noppen and I. Groher, "What can software engineering do for sustainability: Case of software product lines," in *Proc. the Fifth Int. Workshop on Product Line Approaches in Software Engineering*, Florence, Italy, pp. 11–14, 2015.
- [40] D. Beuche, A. Birk, H. Dreier, A. Fleischmann, H. Galle *et al.*, "Using requirements management tools in software product line engineering: The state of the practice," in *Proc. Software Product Line Conf., SPLC 2007 the 11th Int.*, Kyoto, Japan, pp. 84–96, 2007.
- [41] H. Leung and Z. Fan, "Software cost estimation," in *Handbook of Software Engineering and Knowledge Engineering Volume II: Emerging Technologies*, University of Pittsburgh, USA & Knowledge Systems Institute, USA, pp. 307–324, 2002.
- [42] J. Fischer, S. Naumann and M. Dick, "Enhancing sustainability of the software life cycle via a generic knowledge base," In: Greve, K., Cremers, A. B. (Hrsg.), *Integration of Environmental Information in Europe*. Aachen, Shaker Verlag, pp. 716–725, 2010.
- [43] K. H. Robèrt, B. Schmidt-Bleek, J. A. De Larderel, G. Basile, J. L. Jansen *et al.*, "Strategic sustainable development—selection, design and synergies of applied tools," *Journal of Cleaner Production*, vol. 10, no. 3, pp. 197–214, 2002.
- [44] G. Ricketts, "Why getting serious about sustainability needs to start with training," [GreenBiz.Com](https://www.greenbiz.com/article/why-getting-serious-about-sustainability-needs-start-training), 2013. [Online]. Available: <https://www.greenbiz.com/article/why-getting-serious-about-sustainability-needs-start-training>.
- [45] P. Lago, R. Kazman, N. Meyer, M. Morisio, H. A. Müller *et al.*, "Exploring initial challenges for green software engineering: Summary of the first GREENS workshop," *ACM SIGSOFT Software Engineering Notes*, vol. 38, no. 1, pp. 31–33, 2013.
- [46] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. M. Easterbrook *et al.*, "Requirements: The key to sustainability," *IEEE Software*, vol. 33, no. 1, pp. 56–65, 2016.
- [47] J. Liebowitz, "The role of HR in achieving a sustainability culture," *Journal of Sustainable Development*, vol. 3, no. 4, pp. 50, 2010.
- [48] A. Pandit, E. A. Minné, F. Li, H. Brown, H. Jeong *et al.*, "Infrastructure ecology: An evolving paradigm for sustainable urban development," *Journal of Cleaner Production*, vol. 163, pp. S19–S27, 2017.
- [49] S. A. Ajila and D. Wu, "Empirical study of the effects of open source adoption on software development economics," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1517–1529, 2007.
- [50] J. Iqbal, M. Omar and A. Yasin, "An empirical analysis of the effect of agile teams on software productivity," in *Proc. 2019 2nd Int. Conf. on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, pp. 1–8, 2019.
- [51] C. Jones, *Applied Software Measurement: Global Analysis of Productivity and Quality*, New York, USA: McGraw-Hill Education Group, 2008.
- [52] B. J. Orser, S. Hogarth-Scott and A. L. Riding, "Performance, firm size, and management problem-solving," *Journal of Small Business Management*, vol. 38, no. 4, pp. 42, 2000.
- [53] S. M. A. Shah, E. Papatheocharous and J. Nyfjord, "Measuring productivity in agile software development process: A scoping study," in *Proc. the 2015 Int. Conf. on Software and System Process*, Tallinn Estonia, pp. 102–106, August 2015.
- [54] M. Hammer and J. Champy, *Reengineering the Corporation: Manifesto for Business Revolution*, HarperCollins e-books, 2009.
- [55] P. A. Samuelson and W. D. Nordhaus, *Economics*, Intl. ed., New York: McGraw-Hill Inc, 1995.
- [56] E. Van Oort, J. Nicholson and J. D'Agostino, "Integrated borehole stability studies: Key to drilling at the technical limit and trouble cost reduction," in *Proc. SPE/IADC Drilling Conf.*, Amsterdam, Netherlands: Society of Petroleum Engineers, 2001.

- [57] V. Tebaldo, G. G. di Confiengo and M. G. Faga, "Sustainability in machining: "Eco-friendly" turning of inconel 718. surface characterisation and economic analysis," *Journal of Cleaner Production*, vol. 140, pp. 1567–1577, 2017.
- [58] R. Amri and N. B. Ben Saoud, "Towards a generic sustainable software model," in *Proc. 2014 Fourth Int. Conf. on Advances in Computing and Communications*, Cochin, India, pp. 231–234, 2014.
- [59] J. H. Yahaya, A. Deraman and A. R. Hamdan, "Software quality and certification: Perception and practices in Malaysia," *Journal of Information and Communication Technology*, vol. 5, pp. 63–82, 2006.
- [60] S. Naumann, E. Kern, M. Dick and T. Johann, "Sustainable software engineering: Process and quality models, life cycle, and social aspects," In: Hilty, L., Aebischer, B. (Eds.), *ICT Innovations for Sustainability Advances in Intelligent Systems and Computing 310*, Cham: Springer, pp. 191–205, 2015.
- [61] D. Green and J. M. Pearson, "Development of a web site usability instrument based on ISO 9241-11," *Journal of Computer Information Systems*, vol. 47, no. 1, pp. 66–72, 2006.
- [62] P. Stoll, L. Bass, E. Golden and B. E. John, "Supporting usability in product line architectures," in *Proc. the 13th Int. Software Product Line Conf.*, San Francisco, USA, pp. 241–248, 2009.
- [63] N. Condori-Fernandez and P. Lago, "Characterising the contribution of quality requirements to software sustainability," *Journal of Systems and Software*, vol. 137, pp. 289–305, 2018.
- [64] M. Al Hinai, "Quantification of social sustainability in software," in *Proc. Requirements Engineering Conf. (RE), 2014 IEEE 22nd Int.*, Karlskrona, Sweden, pp. 456–460, 2014.
- [65] M. Saunders, P. Lewis and A. Thornhill, *Research Methods for Business Students*, Essex, England: 5/e. Pearson Education, 2011.
- [66] C. Fisher, *Researching and Writing a Dissertation-for Business Students*, Prentice-Hall, Pearson Education Limited, 2007.
- [67] K. Khalid, H. Hilman and D. Kumar, "Get along with quantitative research process," *International Journal of Research in Management*, vol. 2, no. 2, pp. 15–29, 2012.
- [68] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed., Hillsdale, NJ, Lawrence Erlbaum Associates, 1988.
- [69] W. Ismail, N. Abedlazeed and Z. Hussin, "Epistemological beliefs of students at high schools: A survey study in Malaysia," *OIDA International Journal of Sustainable Development*, vol. 2, no. 8, pp. 39–46, 2011.
- [70] J. Yahaya, K. Raisian, S. R. Ahmad Ibrahim and A. Deraman, "Green software process based on sustainability dimensions: The empirical investigation," in *Proc. the 1st Int. Conf. on Informatics, Engineering, Science and Technology (INCITEST 2019)*, Bandung, Indonesia, 2019.
- [71] Z. H. Abdullah, J. Yahaya, S. R. Ahmad Ibrahim, S. Fadzli and A. Deraman, "The implementation of software anti-ageing model towards green and sustainable products," *Int. Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 5, pp. 42–50, 2019.