

Green Software Process Based on Sustainability Dimensions: The Empirical Investigation

Jamaiah Yahaya¹, Komeil Raisian², Siti Rohana Ahmad Ibrahim³, Aziz Deraman⁴
{jhy@ukm.edu.my¹, Raisiank@hotmail.com², srai.gurlz@gmail.com³, a.d@umt.edu.my⁴}

Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia
Bangi, Selangor, Malaysia ¹, Faculty of Information Science and Technology, Universiti Kebangsaan
Malaysia, Bangi, Selangor, Malaysia ², Faculty of Information Science and Technology, Universiti
Kebangsaan Malaysia, Bangi, Selangor, Malaysia ³, School of Informatics & Applied Mathematics,
Universiti Malaysia Terengganu, Kuala Terengganu, Malaysia⁴

Abstract. Green and sustainability issues are being considered as the alternative effort to ensure the long-time usage and durable of a software. The overall target for green software is to produce sustainable software products with minimal negative impact on the environment. In general, sustainability in software is associated with the ability to operate in a longer time and viable in their environments. Issues regarding sustainability are usually based on economy, social, and environmental view. In order to achieve and maintain sustainability in software, software process should undergo and comply with green. Previous studies emphasis on green hardware in respect to energy consumption, power consumption, waste reduction and disposal. However, it seems lack of studies on green software even though software has impact indirectly to the environment and to the software itself. This paper presents the empirical study conducted in Malaysia to investigate the sustainable practices in software process toward achieving green process and environment. One hundred and two (102) participants were involved in this survey and most of them were software engineers and software developers. The results reveal that the sustainability practices are important and relevant for green software process.

Keywords: software process, green software process, empirical study, sustainability practices

1 Introduction

Green and sustainability issues are being considered as the alternative effort to ensure the long-time usage and durable of a software. The overall target for green software is to produce sustainable software products with minimal negative impact on the environment. In general, sustainability in software is associated with the ability to operate in a longer time and viable in their environments. Issues regarding sustainability are usually based on economy, social, technical and individual view [1]. Previous studies emphasis on green hardware in respect to energy consumption, power consumption, waste reduction and disposal. However, it seems lack of studies on green software even though software has impact indirectly to the environment and to the software itself. Several studies touch on green exclusively but limited integration with sustainability dimension. It is believed that integration between sustainability and green may have potential to make software more tolerate and applicable for the new generation that demand

for greener environment. It also beneficial to improve and enhance the studies of software engineering in future.

Today's modern and largely digitised world is concerned with the development of more recent and practical software. This has led to scientific development in software engineering which was defined as systematic, disciplined and quantifiable methodologies that deal with the development, operation and maintenance. In recent years, creating computing or information technology products and eco-friendly software have become the target of the hardware and software industries. The concept of the eco-friendly target basically revolves around decreasing carbon utilisations, saving energy and minimising dangerous waste.

Green software development is a moderately new study arena in green computing. The industry of software development is now receiving demands to incorporate green software development. Software assumes an expanding part of our lives and its environment and sustainability effects are becoming increasing vital. Therefore, green software necessities require consideration, as well as a conventional quality within software development.

This paper will discuss further on the background works in the related areas. Section 3 discusses the empirical study conducted in Malaysia. The discussion will cover the questionnaire design, questionnaire validation and data collection. Section 4 presents the results and discussion. While section 5, the last section presents the conclusions and future work.

2 Background Work

The background work covers two main areas which are the software process and the green measurements based on sustainability dimensions for software process.

2.1 Software Process

Software Development Life Cycle or SDLC is a well-known and oldest software development model. It was developed based on the systematic implementation and maintenance through break down process in different stages incorporating requirements definition, system and software designs, implementation and unit testing, also operation and maintenance [2]. Each process in SDLC consists of defined activities to support and fulfil the stages. Study conducted by Shenoy and Eeratta showed that current implementation of SDLC may encounter some problems in promoting green environment [3]. The problem is that certain important decisions taken in these phases like use of paper, generation of e-Waste, power consumption and increased carbon foot print by means of travel, air-conditioning and many more may harm the environment directly or indirectly.

There is lack of models that define how a software can be developed and maintained in an environment friendly way [3]. Study by [3] aimed to build environmentally-friendly software and improved the development process and moved towards greener of development and sustainable software. They proposed an enhanced model for software development life cycle and proposed measurements of carbon emissions of power reduction and paper usage to help organizations moving towards greener and sustainable in software development. In addition, through this model, the researcher claimed that electronic waste could be gradually reduced in terms of the cost reduction during software development process in organisation. Therefore, the products developed are environmentally friendly and give added value to consumers. The model

presents the integration of best practice for green software development based on electronic waste and power reduction.

Previous studies have revealed that SDLC is considered as a software process model that can guarantee to produce good quality software [3-4]. At the same time, there are several other models for software process with specific objectives. Among the most widely used development models are Waterfall Model, V Model, Incremental Model, Rapid Application Development (RAD) Model, Agile Model, Iterative Model and Spiral Model. Each of these models has a goal of having a good quality assurance in software development process [2][5]. No matter what model is chosen for developing software, this activity involves complex processes that are often exposed to errors. Thus, requires to manage and re-examine the activities to reduce errors and maintain sustainability. The sustainable of software products and process are measured with minimal negative impact on the environment.

In order to look in more significant at each stage, the following will explain and discuss in more detail the fundamental of software process that contained in all software process models: collecting or analysis requirement, design, implementation and coding, testing and operation and maintenance [5][8].

2.1.1 Collecting or Analysis Requirements

This stage is the basic to the success of software development project. Requirements and specification are gathered and collected for proper and detail documentation. This is a procedure of iterative with high communication occurring among the stakeholders, end users and developer team [6]. Several methods [7] can be utilized for requirements collection such as: -

- Determine and obtain stakeholder's requirements by utilizing interviews, questionnaire and surveys.
- Multiple utilisation cases in order to define activity within the exacting system or also known as observation.
- Prototypes can be developed and used to demonstrate the user's final product overview and function.

In an enterprise environment, this implies investigating users, making sense of what they need and after that designing what a successful result would resemble in the proposed software. In requirement phase, functional and non-functional requirements must be specified and detail.

2.1.2 Design

Prerequisites of technical design are set up in this stage to guide the development team. The business requirements are utilised to characterise how the application will be composed. Requirements of technical are detailed into tables of database, new input/output are characterised and designed, hardware, system requirements, algorithm, component design, quality specification and also security procedure. The GUI interface designs take place to meet the functional and non-functional requirements. Several issues and steps can be considered during this stage to improve and achieve green.

2.1.3 Implementation and Coding

This stage is the real coding and unit testing of the procedure within the development team. This phase involves coding, programming and unit testing. Issues such as efficient code, programming languages and testing procedures are relevant to be considered in this phase for green process. The completed software product hereby is input to the testing phase.

2.1.4 Testing

While the software or application is in testing phase, various testing types will be executed incorporating system testing and integration. User acceptance test is the last testing part and is executed via the end user to guarantee the system is accordance to their expectations. Based on this activity, defects might be found and more activity might be needed in the requirement specification, design or coding. The test case data and procedures are significant to be deliberate to achieve green.

2.1.5 Operation and Maintenance

The size of the software will distinguish the complexity of the implementation of deployment [2]. Several deployment methods are applied to ensure the deployment of new software or application are smooth and does not interfere the current business operation. Trainings are also included for various level of users as well as the technical team.

2.2 Green measurements based on Sustainability Dimensions

The general recognition about a software is that it is normally green and software barely has an environmental impact. Can software be greener than before? Green and sustainability are being considered as the alternative effort to ensure the long usage and durable of a software. As mentioned before that the main goal for green software is to produce sustainable software products with minimal negative impact on the environment. Therefore, as proposed by Calero and Piattini, green software can be classified into green in software and green by software [26]. Green by software covers aspects such as grid management, cloud management, data centres, dematerialisation and etc. Green in software is identified as how to create software in a more sustainable manner which incorporates issues such as process, product, life cycle and governance [26].

In general, sustainability in software is associated with the ability to operate in a longer time and viable in their environments. Literature study shows that current studies in software process do not focus on integration of sustainability measurements towards achieving green process. Previous studies revealed that sustainability measurements can be identified as economic, social, and environmental dimension. It permits a simple perception of reconciliation and understanding of environment, social and economic issues [23]. The main and fundamental dimensions for sustainability need to be embedded as the prerequisites for sustainability and green software process [24]. Referring to sustainability in software, the economic dimension is concerned with preserving capital and value. The social dimension is concerned with

maintaining communities and the environmental dimension seeks to improve human welfare by protecting natural resources [25].

Green and sustainability software process is the art of developing green and sustainable software with green and sustainable software engineering process. Therefore, it is the art of defining and developing software products in a way so that the negative and positive impacts on sustainable development that result and/or are expected results are continuously assessed, documented and used for further optimisation of the software [8][9]. Number of works in green software development in general, but not much consideration on embedded green measurements in term of sustainability. The sustainability practices toward green software process can be summarised as shown in **Table 1**.

Table 1. Sustainability practices in SDLC phases

SDLC Phases	Sustainability Practices	Source of References
Requirement Engineering	i) Collect requirements through electronic means.	Agarwal et al. [10], Shenoy et al. [3]
	ii) Consider software durability	
	iii) Provide power down capability	
	iv) Feasibility studies must assess the impact of the project on the environment	
Design	i) Designs must include specific measures and practices that relate to environments.	Agarwal et al. [10], Shenoy et al. [3]
	ii) Start designing with simple initial design.	
	iii) Fewer utilizing devices and hardware	
	iv) Use of virtualized systems	
	v) Avoid frequent change	
Coding	i) Implementing pair programming (two programmers working together)	Williams et al. [11], Shenoy et al. [3], Schindler [12]
	ii) The focus should shift to writing energy efficient codes that minimize the use of system resources.	
	iii) Do not use duplicated code	
	iv) Use automatic code generation	
Testing	i) The system resource usage metrics need to be added to the list of test cases that need to be passed.	Agarwal et al. [10], Shenoy et al. [3]
	ii) Use Automatic Testing	
	iii) Should concern reuse test	
Implementation	i) Auditing and reporting the organization's energy consumption and savings.	Mohankumar & Anand [13], Agarwal et al. [10]
	ii) Modifying supply chain, production activities and organization flow.	
	iii) Creating more efficient business operators, buildings and system.	
	iv) Helping the decision-making process by analyzing modeling, and simulating of environmental impacts that may occur.	

3 The Empirical Study

The approach started with: 1) adapting and designing the questionnaire from the previous literature, 2) questionnaire content validation, 3) distributing the questionnaire and data collection, 4) data analysis, and 5) results and findings. The objectives of this survey were to study the current green practices in software process among software practitioners in the industry and second was to identify essential sustainability software process towards green environment.

3.1 Questionnaire Design

The design of the instrument was based on closed ended questions that allowed the respondents to provide an exact answer for each question. All the questionnaire items were adapted from previous related studies, which related to the context of each factor. Additionally, the recommendations made by experts were used to improve the questionnaire. The questionnaire was written in English.

The adapted process during this design activity started with chosen the proper items based on the previous researches which they used the same items for the identified factors. The items were selected in regard to the aim of this research and context. In this survey we focused on identifying green aspects related to SDLC activities and phases based on sustainability practices as shown in **Table 1**.

One of the designs in this questionnaire is related to software SDLC practices that must be performed in order to produce high green software. The classified software processes are Requirement, Design, Coding, Testing and Implementation process and the practices are related to sustainability dimensions and measures. The scale to answer these questions are using the 5-point numerical scale which ranges from 1 to 5. Numerical value of 1 represent Not Important, 2 represents Less Important, 3 represents for Neutral, 4 represents for Important and 5 means Most Important.

3.2 Questionnaire Validation

The questionnaire was validated using two-step methods before distributed to the respondents to ensure that the items were free of errors, easy to understand, and content appropriateness of the items. In the first step, the questionnaire was validated by three experts who were university academic staff or researchers in the same field. The questionnaire was validated according to its language, clarity, content and its contradictions or duplications if any.

The recommendations and feedbacks from the experts were followed and corrected before the pilot study was conducted. During the pilot study, the reliability test using Cronbach Alpha test was carried out. The Cronbach Alpha test showed that the items in the questionnaire have relatively high internal consistency.

3.3 Distributing and Data Collection

3.3.1 Sampling and Respondents

The list of possible software organisations from private and government companies that have related software background was identified. The sample of respondents was also gathered through members of research centres and working in software development. Out of 220 contacted respondents, 148 (67%) had participated while 72 (33%) refused to participate. A total of 102 questionnaires or 69% of the respondents were considered as valid respondents of this survey. It is considered adequate and acceptable based on [14] and [15]. In this study, an online questionnaire was made by utilising google document and messaged the respondents to consent and complete the questionnaire. Table 2 shows the detail of respondent rate of this survey.

Table 2. Respondents Rate

Detail	Number	Percentage (%)
Number of respondents willing to participate	148	100%
Valid questionnaires	102	69%
Unreturned questionnaires	28	19%
Incomplete questionnaires	18	12%

3.3.2 Demographic

Table 3 shows the demographic of the respondents. It shows that 62 (or 60.7%) of the total respondents are engaged as software developers and software engineers. At the same time almost 26% are IT professionals and researchers related to computing.

Table 3. Respondents Background

Position in the Organisation	Frequency	Percentage (%)
Board of Directors	2	2.0
IT Professional	15	14.7
Researcher	12	11.7
Project Manager	7	6.8
Software Developer	26	25.5
Software Engineer	36	35.4
Others	4	3.9
Total	102	100%

Cross analysis to investigate characteristic of respondents based on their experiences in software development activities were also carried out. Table 4 shows the result. Out of 102 respondents, nearly 5% have over 20 years of experience in software development. Majority of them (43.1%) have between 6 to 10 years' experience and 25.5% have experience between 11 to 20 years (see also **Figure 1** for graphical representation).

Years of Experience	Frequency	Percentage (%)
< 1	8	7.8
1-5	19	18.6
6-10	44	43.1
11-20	26	25.5
21-29	5	4.9
Total	102	100%

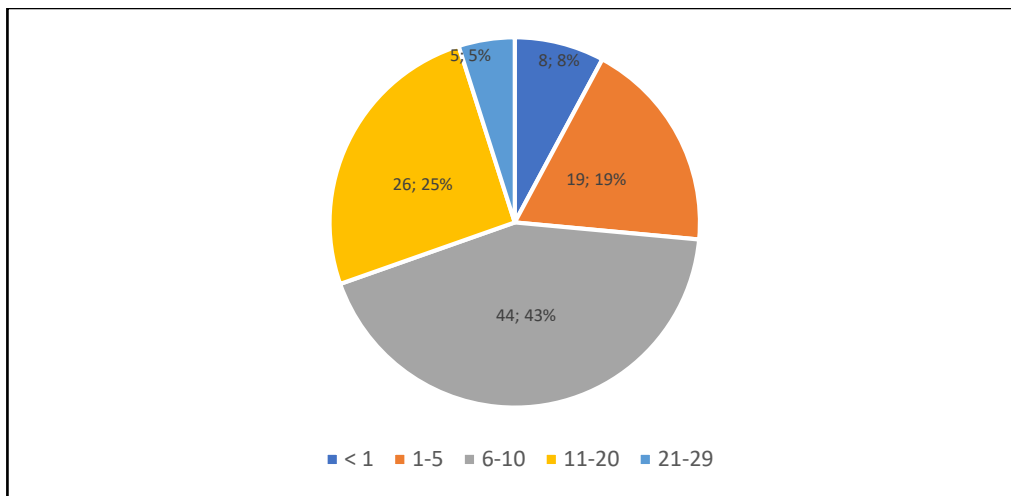


Fig. 1. Years of experience in software development by respondents

4 Results and Discussion

This section addresses software practitioners' opinions and experiences regarding their familiarity of the software process methodology and activity. The sustainable software development life cycle is a framework defining tasks performed at each phase in development process.

In this survey, respondents were asked whether they were aware with sustainable software process and product. Most respondents (or 61%) claimed that they are familiar with sustainable SDLC practices, while 39% admitted otherwise as shown in **Figure 2**. According to Scheuer [17] and Kossek et al. [18], studies showed that increased in green experience adoption related to a shift in familiarity of that particular aspect which in this study related to green SDLC software process. Familiarity plays a main role in adoption practices.

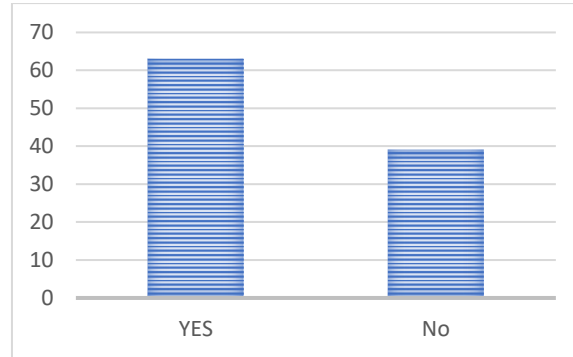


Fig. 2. The familiarity of sustainable SDLC among respondents

In this survey, respondents were asked regarding software SDLC practices and activities that must be performed in order to produce high green software. The questionnaire classified the software processes into: Requirement, Design, Coding, Testing and Implementation. The respondents were asked to answer the questions and gave answer based on Likert scale 1 to 5 where 1 = not important, 2=less important, 3=neutral, 4=important, 5=most important. The mean value for each practice was obtained from the analysis, as it represents the most selected answers on average. The 5-point numerical scale was used for these questions and the scale was then mapped into equal intervals. The interval ranges were calculated using the following formula shown in (1). Formula (1) is adapted from Ismail, Abedlazeed and Hussin [16].

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

where n is the maximum number in the used scale, which is equal to 5. Thus, the interval size of the consideration level between one through seven is 0.8, as the interval values are depicted in **Table 5**.

Table 5. Degree of importance and 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

5.1 Current sustainable practices in SDLC

Table 6 shows the mean values obtained by current sustainable practices for each phase of the SDLC process. Results from the study indicate that most of these SDLC processes obtained high consideration among respondents, whereby the mean values are in the range of Important to Most Important based on the Interval Values in **Table 5**.

The analysis shows and verifies that almost all sustainability practices are considered important and very important towards green software process. This was revealed by the respondents of this survey as shown in **Table 6**.

5.2 Sustainability Dimensions and Measurements in Software Process

This section discusses the responds and feedbacks obtained through the survey regarding the importance of sustainability dimensions and measurements in software process from respondent's perspectives. The respondents were asked about the sustainability practices from sustainability dimensions which needed in order to achieve green process. The dimensions are environment, social and economic as discussed in **Section 2**.

Table 7 shows the results obtained by calculating the mean score by each sustainability dimensions and measurements. The mean scores are then been mapped into Degree of Importance and Interval Values (see **Table 5**). The results show that mostly the sustainable measurements and practices obtain high consideration during development with the mean values of Most Important and Important achieved.

Table 6. Degree of importance of SDLC practices by respondents

SDLC Phases	Practices	Mean	Degree of Importance
Requirement	Collect requirements through electronic means	4.20	Important
	Consider software durability	4.28	Most Important
	Provide power down capability	4.20	Important
	Feasibility studies must assess the impact of the project on the environment	4.22	Most Important
Design	Designs must include specific measures and practices that relate to environments	4.32	Most Important
	Start designing with simple initial design	4.18	Important
	Using fewer hardware devices	4.23	Most Important
	Use of virtualised systems	4.25	Most Important
	Avoid frequent changes	4.25	Most Important
Coding	Implementing pair programming (two programmers working together)	4.35	Most Important
	The focus should shift to writing energy efficient codes that minimise the use of system resources.	4.22	Most Important
	Do not use duplicated code	4.22	Most Important
	Use automatic code generation	4.32	Most Important
Testing	The system resource usage metrics need to be added to the list of test cases that need to be passed	4.14	Important
	Use Automatic Testing	4.04	Important

	Should concern reuse test	4.19	Important
Implementation	Auditing and reporting of the organisation's energy consumption and savings	4.23	Most Important
	Modifying supply chain, production activities and organisation flow	4.14	Important
	Creating more efficient business operators, buildings and system	4.01	Important
	Helping the decision-making process by analysing modelling and simulating of environmental impacts that may occur	4.23	Most Important

Table 7. Sustainability dimension and measures in organisations by respondents

Dimension	Measurements and Practices	Mean Score	Degree of Importance
Social Sustainability	Employee support, cohesiveness, and shifting patterns of teamwork are also salient and relevant in green process.	4.26	Most Important
	Process and practices are supported by tools in software development activities (requirement, analysis, design, implementation and maintenance)	4.00	Important
	The tool supports the structure of the organisation which intend to the development process.	3.19	Neutral
	Tools are needed in the requirement engineering to support and automate the process.	4.29	Most Important
	Tools are needed in development practices to create strategies and methods in ensuring the development performance.	4.00	Important
	Tools are required in development life cycle to develop and test components for consistency with common and varies features.	4.25	Most Important
	Tools are required in software product architecture to portray the development without violating the product scope and definition.	2.91	Neutral
	Employ development aids that help project coordination.	4.11	Important.
Economical Sustainability	The main goal of software process 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
	The usability aspect is necessary in software development.	3.23	Neutral
	Usability is a complex development and process.	4.01	Important
	Various functions in software are integrated based on usability development.	4.16	Important
	Usability development requires more times and efforts.	3.96	Important
	Confident with usability development.	3.22	Neutral

	Consistency with usability development	3.90	Important
	Easy to learn related to usability	4.19	Important
Environment Sustainability	Software performance depends on the basic computing resources (such as CPU and memory).	4.19	Important
	Software needs faster CPUs and increased memory.	3.99	Important
	Memory usage is one of main requirements for software development based on environmental sustainability.	4.31	Most Important
	The size of total storage space for each software is important concern for developers.	4,25	Most Important
	Short-term and long-term storage usage in necessary during development.	4.20	Important
	Energy consumption have direct impacts related to resources during development and use of software.	2.87	Neutral
	The functional Co2 emission with respect to environmental sustainability in software development process.	4.20	Important
	The functionality aspect of the software	2.93	Neutral

Sustainability dimension and measures were introduced and asked to the respondents regarding the applicability and considerations in software development process. Following the sustainable SDLC process principles base on [3][10], software needs to be developed and modified in order to avoid of degradation and increase in maintenance activities and lead to carbon footprint. This to ensure and also employ for long time operations of the particular software. Thus, longer lifetime would be essential element in order to achieve green and sustainable software. The results show that these practices gained high consideration by respondents in this survey.

During design phase of software process, the initial aim has to be to achieve simplicity in design. The developers must be able to find out the design in easy way without much definition and this helps to use lower paper works. Complicated design might lead to higher documentation activity and utilising computer devices and also re-design which may lead to fuel consumption on travelling, higher power consumption, and other resources usage. In complicated and difficult design, effort must be provided to decrease the complexity. This is needed to maintain sustainability and green in process [3]. The use of virtualise system, hardware and cloud may lead to shared reusable resources and reduce power consumption and lesser garbage of hardware [10]. Reduce reimplementaion of same modules or elements may accelerate the development process to bring plug-in architecture and software reusability [19]. The results of this survey too align with these thoughts.

Following coding stage of software process, pair programming might be useful and applicable for maintaining sustainable and green process. Pair programming aims at removing knowledge silos and improving code quality (including discoverability and readability) [20]. Furthermore the efficiency in code which means less resources used during development but the software must not get less performance in terms of latency, processing speed and others [21]. Automation of repetitive implementation tasks can be considered as very critical part of sustainability implementation because they reduce time taken to achieve the tasks and minimal manual errors. Hardware-specific codes and programs should be discouraged in green process [3]. These practices were revealed essential consideration by previous studies, and this survey has discovered the same findings.

Test automation must be encouraged as it will reduce manual errors in order to implement testing activity based on sustainable SDLC and emphasise on test reuse cases and standardise the testing process. This develops efficiency in testing through productivity and decreases power consumed by additional resources in manual testing process [3]. In order to achieve sustainable process in testing, resource profiling and performance test must also be improved because otherwise there might be demand for extra memory, processor cycles and hardware. The improve computer hardware and other resources are needed to maintain sustainability and green in software process [3][10].

In this survey, the implementation and deployment practice obtain high consideration with Most Important and Important among respondents. This finding is also supported by previous study [13][3]. Previous studies have revealed that software implementation especially in large system such as Enterprise Resource Planning requires appropriate steps such as creating and modifying efficient business operators, supply chain and decision-making process. In sustainable development, teams are able to be proactive about changes in their ecosystem. Their ability to be proactive is enabled by their attention to doing the work that is of the highest value to customers or users with high quality and reliability and an eye toward continual improvement despite increasing complexity. In addition, in this phase the actual software is prepared for being installed in the production environment. The size of the installation package plays a major role, larger the size of installation, more the time taken for installation in the production environment. This also adds an overhead of storage and maintenance and demands lot of disk space for storage. It is also suggested to use of a common artefact model for documentation during development as well as during implementation and maintenance phase, facilitating collaboration in the initial development process and improving reuse or refactoring in the maintenance process. All these will improve sustainability and achieve green in organisation environment [3][22].

5 Conclusions and Future Work

Software engineering has been recognised today as a principle and important role of our life that may have impact in term of social, economy and environment. Software engineers are starting to realise the significance of designing sustainability regarding software processes. The field of sustainable and green software are still young. Thus, in this modern society with new generation demand, researchers' efforts are mainly focused on green and sustainable software process and further toward green software product itself. The aim of this study is to understand the related potential factors for improving the recent trend of sustainability dimension towards green life cycle and software process.

Most studies previously work on green ICT or green IT that in lined to the impacts of hardware towards green environment. Yet our work focuses regarding the impacts of building software products from process point of view. In this way, the arrangement was to recognise current issues in sustainable software practices and further to explore the measurements of sustainability for green software process. This paper has presented the empirical study conducted in Malaysia that investigated software process from sustainability dimension's view which may lead to green software process. For future work, we will propose a model to coordinate sustainability and green in software development process.

Acknowledgments. This research is funded by Universiti Kebangsaan Malaysia, The Research University Grant (AP-2017-005/3).

References

- [1] Calero, C. and Piattini, M.: Green in software engineering, Springer International Publishing (2015).
- [2] Sommerville, I: Software Engineering, 10th edition, Essex, England (2016).
- [3] Shenoy, S. & Eeratta, R.: Green software development model: An approach towards sustainable software development, India Conference (INDICON), 2011 Annual IEEE, December (2011)
- [4] Stoica, M., Mircea, M. & Ghilic-Micu, B.: Software Development: Agile vs. Traditional Informatica Economica, Vol. 17(4), pp.64-76 (2013).
- [5] Ruparelia, N.B.: Software development lifecycle models. ACM SIGSOFT Softw. Eng. Notes (2010).
- [6] Kalinowski, M., Curty, P. Paes, A., Ferreira, A., Spinola, R., Fernandez, D.M. Felderer, M. and Wagner, S: Supporting Defect Causal Analysis in Practice with Cross-Company Data on Causes of Requirements Engineering Problems, International Conference on Software Engineering (ICSE) - Software Engineering in Practice (SEIP), Buenos Aires, Argentina, May (2017).
- [7] Mylopoulos, J., Chung, L. and Nixon, B.: Representing and using non-functional requirements: A process-oriented approach. IEEE Transactions on software engineering, Vol. 18(6), pp.483-497.(1992).
- [8] Dick, M., Naumann, S.& Kuhn, N.: A model and selected instances of green and sustainable software, IFIP Adv. Inf. Commun. Technol., Vol. 328, pp. 248–259 (2010)
- [9] Naumann, S., Dick, M., Kern, E. and Johann, T.: The GREENSOFT Model: A reference model for green and sustainable software and its engineering,” Sustain. Comput. Informatics Syst., Vol. 1(4), pp. 294–304 (2011)
- [10] Agarwal, S. Nath, A. and Chowdhury, D.: Sustainable approaches and good practices in green software engineering. International journal of research and reviews in computer science (IJRRCS), Vol. 3(1) (2012).
- [11] Williams, L., Rubin, K. and Cohn, M.: Driving process improvement via comparative agility assessment, IEEE AGILE conference, pp. 3-10 (2010).
- [12] Schindler, C.: Agile Software Development Methods and Practices in Austrian IT-Industry: Results of an Empirical Study, 2008 International Conference on Computational Intelligence for Modelling Control & Automation, Vienna, Austria (2008).
- [13] Mohankumar, M. and M. Anand, M. K.: A Green IT Star Model Approach for Software Development Life Cycle, International Journal of Advanced Technology in Engineering and Science, Vol. 3(1) (2015).
- [14] Saunders, M.N.K., Lewis, P. & Thornhill, A.: Research Methods for Business Students (7th Edition), Pearson (2015).
- [15] Fisher, C.: Researching and Writing a Dissertation: An essential guide for business students, 3rd Edition, Pearson Education Canada (2007).
- [16] Ismail, W. , Abedlazeed, N. and Hussin, Z.: Epistemological Beliefs of Students at High Schools: A Survey Study in Malaysia, OIDA International Journal of Sustainable Development, Vol. 02 (08), pp.39-46 (2011).
- [17] Scheuer, C.W.: Adoption of residential green building practices: Understanding the role of familiarity, PhD thesis, University of Michigan (2007).
- [18] Kossek, E.E., Lautsch, B.A & Eaton, S.C.: Flexibility Enactment Theory: Implications of flexibility type, control, and boundary management for work-family effectiveness. Lawrence Erlbaum Associates Publishers (2005).
- [19] Wolfinger, R.: Plug-in architecture and design guidelines for customizable enterprise applications. In Companion to the 23rd ACM SIGPLAN conference on Object Oriented programming systems languages and applications, pp. 893-894. ACM (2008).

- [20] Sedano, T., Ralph, P. & Péraire, C.: Sustainable Software Development through Overlapping Pair Rotation, Conference: 10th International Symposium on Empirical Software Engineering and Measurement (ESEM 2016) (2016).
- [21] Toomin, M., Begel, A. & Graham S.L.: Managing duplicated code with linked editing. In visual language and human centric computing, 2004 IEEE Symposium on, pp. 173-180 (2004),
- [22] Penzenstadler, B.: What does Sustainability mean in and for Software Engineering? January 2013, in 1st International Conference on ICT for Sustainability (ICT4S) (2013).
- [23] Carter, C.R. & Easton, L.: Sustainable supply chain management: evolution and future directions. Intl. journal of physical distribution & logistics management, 41(1), pp. 46-62 (2011).
- [24] Anthony, B. J., & Majid, M. A., & Romli, A.: An empirical study on predictors of green sustainable software practices in Malaysian electronic industries. Journal of Information and Communication Technology, Vol 17, issue 2, pp. 347-391 (2018).
- [25] Lago, P., Koçak, S.A., Crnkovic, I. & Penzenstadler, B.: Framing Sustainability as a Property of Software Quality, Communications of the ACM, Vol. 58(10), pp. 70-78 (2015).
- [26] Calero, C. & Piattini, M.: Green in Software Engineering, Springer International Publishing Switzerland (2015).