Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SUSTAINABLE SOFTWARE DEVELOPMENT LIFE CYCLE PROCESS MODEL BASED ON CAPABILITY MATURITY MODEL INTEGRATION: A STUDY IN MALAYSIA

[1]KOMEIL RAISIAN, [2]JAMAIAH YAHAYA, [3]AZIZ DERAMAN

[1,2] Faculty of Information Science and Technology, The National University of Malaysia, 43600 UKM Bangi Selangor, Malaysia

[3] School of Informatics & Applied Mathematics, Universiti Malaysia Terengganu, Kuala Terengganu, MALAYSIA

E-mail: [1]raisiank@hotmail.com, [2]jhy@ukm.edu.my, [3]a.d@umt.edu.my
*Corresponding author: komeil Raisian

ABSTRACT

Green software engineering is an important view of software engineering process in the current century. Previously, software engineers concerned mainly with development of hardware or software without giving more importance to sustainability. The field of green software and green software engineering is still young. Thus, in modern society, researches efforts are mainly focused on green and sustainable software engineering itself. The development of sustainable software has been identified as one of the key challenges in the field of computational science and software engineering and also there no is clear idea regarding how to accomplish the Green and sustainability in Software Development Life Cycle (SDLC) stage. However, the aim of this study is the adjustments in the current SDLC and presented sustainable SDLC. In order to access the goal, the study firstly is used Standard Software development process method which is Capability Maturity Model Integration (CMMI) to institutionalization the model. Next, the research is used survey sample population is selected equal with 102 respondents that is Non-probability or not randomly obtained from international software organizations in Malaysia to identify main stages of the Green Software Development Life Cycle and finally is investigated based on that.

Keywords: *Software Development Life Cycle (SDLC), Green and sustainable software, Capability Maturity Model Integration (CMMI), Model*

1 INTRODUCTION AND BACKGROUND

Today and in almost our daily life, software is played main role and society is confronting the regularly expanding deficiency of non-inexhaustible resources and the limits to development [1,2]. At the same time, the Information Communication and Technology (ICT) utilization distributes essentially to the use of our resource of planet [1]. ICT bears a much of potential in all wide range of various application fields to enhancing sustainability. Software engineering study has put sustainability on its emergency issue. Sustainability is currently a ubiquitous term in calls for research proposals and conference sessions. For instance, "Sustainable Software for a Sustainable World" was the greatest subject of the software engineering conference in 2011. Nevertheless, particularly in Software Development (SD), it has long term effect on studies and activities [3]. However, in literature, there is no overview of the current state of the art in supporting sustainability in software engineering research and practice. Sustainable software development is a principle and a going with set of activities that give ability a group to maintain and accomplish an ideal improvement area inconclusively [4].

According to current research by Raisian et al. [5] recognized recent issues in green software engineering and examine the aspect of sustainable and create green software product to explain a conceptual model of sustainable software engineering product to wind up even greener. Consequently, this study recommends a technique to incorporate sustainability in product life cycle.

As indicated by Murugesan [6], the field of green computing is the review and practice of designing, assembling, utilizing and disposing of PCs, servers, and also, related subsystems, for example, printers, screens, storage devices, communications and networking systems with effectively and efficiently

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

by lower or no environment affect. It is about ecologically friendly utilization of PCs and related technologies. The researchers utilize the same identification for the idea of green and sustainable software. In this way they characterize green and sustainable software as 'software, whose immediate and circuitous negative effects on economy, society, people, and environment that outcome from advancement organization, and use of the software are insignificant and/or which positively affects sustainable development [7]. They consider that immediate effects are identified with resource and power consumption amid the creation and utilization of software, when marginal effects will impacts the software product use, together with different procedures and long haul systemic impacts.

Naumann et al. [7] presented that the distinctive viewpoints swinging to the dimensions of green software engineering are contained in the GREENSOFT model. The GREENSOFT model is an applied reference model that backings IT experts and software clients in the sustainable improvement and utilization of software. It comprises of four sections such as the software products life cycle, criteria and measurements, process models and instruments. The methodology models analyze the improvement, acquiring, organization and software use. Penzenstadler et al. [8] proposed a broadened assortment of knowledge for sustainable software engineering that incorporates related application areas and sustainability ideas from related controls that scientists can gain from when further exploring Sustainability in software engineering.

According to the GREENSOFT model by Naumann et al. [7], Mahmoud and Ahmad [9] show a comprehensive SDLC for eco-friendly in software engineering that called the green and sustainable software life cycle. This model is including two level model that comprises in the primary level of a software product engineering process and in the next level of a categorization of tools that can assets and to distinguish how software has effect on resource consumption and energy.

Shenoy and Eeratta [10] introduced other viewpoint of a green software development model. They consider the entire software life cycle and furthermore give recommendations in order to software simply development stages. Kaefer [11] likewise presents calculated and building issues concerning energy consumption of software and concept for fusing issues of energy efficiency into software activities. Researchers particularly provide to requirement engineering [12, 13, 14]. Others take an energy-efficient software design during concentrating on software engineering in the Web knowledge [15, 16]. A few techniques and standards to decrease the websites energy consumption [17, 18]. Shaw et al. [19] recognized that the development of complex software system can greatly benefit by the various life-cycle views and they mentioned there are different perspective of in Software Development Life Cycle (SDLC). Stakeholders of system have different interest and views. The example process and lifecycle model is based upon (International Organization for Standardization [20]), IEEE-SA (“Institute of Electrical and Electronics Engineers Standards Association [21]), (International Organization for Standardization, [22]), ISO 12207 [23] and ISO 14040 [24].

Dick and Naumann [17] presented the consequences of a serval of master interviews that frame the fundamental for a Sustainable Software process meaning and by and generally in entire life cycle process model to Sustainable and Green Software Engineering and furthermore six solid cases of this model for experts. It is proposed to support starting points to exercises that prompt software that are improved, provisioned and utilized with concern on sustainability issues. Patel et al. [25] recommended that the architectural part perspective in life cycle of software development and its function to get it life cycle design, furthermore characterizes the recently included stage in architectural perspective and the related outcome. This perspective might clear to be procedures from the view of individual partners. Whenever orchestrated, they shape an intelligent perspective of the whole software life cycle.

Based on latest research and specially our review research which was current challenges of green and sustainable software engineering So far not many studies have been led on the development phase of sustainable software independently. However, it is important to investigate the significance of sustainable software development life cycle process and green software engineering and applied model. The novelty of the study is how and which standard software development methodology can be suitable to accomplish the Green and sustainability in software SDLC Process. The other sections of this paper are organized as follows: Section two presents CMMI and its levels to clear and adoption to study goal. Section three describes the Sustainable SDLC and each phase and section four presents conceptual research model. The methodology and analysis of sustainable SDLC

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

model are presented in section five and section six explains around discussion and conclusion for this study.

2 CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

The Capability Maturity Model (CMM) is a methodology utilized to refine and develop software development process in organization [26]. According to Morales et al. [27],the Capability Maturity Model Integration (CMMI) was created by the SEI (Software Engineering Institute), a body founded by the United States Department of Defense and Carnegie Mellon University in order to unite the large number of models created by SEI and other organizations throughout the years and especially in software development. Because of the importance of this model worldwide, CMMI is described based on its basic structure and the levels that make it up. A maturity model, is a structured collection of elements that describe characteristics of effective processes, this model provides among other things a place to start, the advantage of previous experiences of a community and a way of defining what improvement means for your organization, and can primarily be used as a benchmark to evaluate organizations [28]. However, Capability Maturity Model (CMM) is a reference model of mature practices in a specified discipline, used to improve and value a group's ability to perform that discipline. Implementing the CMMI model in an organization brings multiple benefits for both the product offered and the organization as such increase productivity, reduce the cost and improve quality, achieve customer satisfaction, increase the return on investment and increase the morale of workers.

CMMI consists of a structure divided into 5 Levels called Maturity, which show the improvements obtained between one level and another. A Level of Maturity: the level of performance that can be expected from an organization. Areas of Process: it is what is composed each level of maturity and consists of a set of practices made in team to achieve a goal. Practices: it is what make up the objectives and these are specific tasks that are performed in the process area to achieve an objective, these practices are divided into two types, the specific ones that are related to specific objectives and the generic ones that are Those related to generic objectives for institutionalization [29]. The CMMI conforms to 5 levels of maturity as previously mentioned, these levels are Initial, Repeatable, Defined, Managed and Optimized [27]. Each level describes as follow:

2.1 Level 1. Initial (Basic Project Management)

In this Level Company has no standard process for software development. Procedures are normally impromptu and disordered. The organization ordinarily does not give a steady domain. Accomplishment in these organizations relies on upon the fitness and heroics of the general population in the organization and not on the utilization of demonstrated procedures [26]. What's more, organizations regularly deliver items and administrations that work. Hence, they frequently assault the financial cost and plan of their projects.

2.2 Level 2.- Repeatable (Standardized Processes)

Organization has installed fundamental software administration procedures and controls. but there is no consistency or coordination among various groups.

This level is a level that costs a lot of work to companies and sometimes there is loss of workers. This level basically consists of seven areas of process that will contribute to project the effectiveness of the management such as manage the requirements, project planning, supervision and control of the project, management of agreements with the supplier, measurement and analysis, ensure process and product quality and configuration management [27].

2.3 Level 3.- Defined (Proc. Analyzed And Measured)

This level is characterized by its organization, because in the whole organization is involved in the efficient process of the software project, these processes are standardized and documented in a more rigorous way [26]. Among these standardized processes, find the process of software development that is integrated by all that are the processes of software engineering and the management of software projects, in this way the organization lays the groundwork for future continuous progress. The process defined is qualitative, that is, there is little data to indicate how much is produced and how effective the process is, because it does not have a standard that directly establishes the steps to follow and the elements that must be measured in the realization of a project.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This level has the so-called key areas, which are engineering, process management, project management and support. In engineering find the development of requirements: customer, product and product component, technical solutions: design, development and implementation and finally product integration: ensure product integration, verification and validation

In Process Management have a process approach in organization: approach to the organization towards process management; A definition of processes in organization: correct definition of the processes of the organization; And training and education: education and training to improve efficiency and effectiveness. In Project Management find an integrated management of projects = (process + products), so-called risk management and integrated management of suppliers, with a team for integrated development. In the Support, an analysis and resolution of the decisions must be carried out: systematic analysis and implementation of the decisions agreed in an adequate organizational environment for the integrated development of the product and the process [27].

2.4 Level 4.- Managed (Continuous Improvement Of Processes)

At this level, projects are managed, organizational decisions are made and a measurement of processes, services and product quality is made. mainly at this level is to carry out which are Organizational development process, Quantitative project management. At this level, it is very important to use automated tools to manage information and have experience in collecting data. It is also necessary to participate committed to the top management, have a degree of experience, a large staff and a successive refinement, thus obtaining with the aforementioned, the step to level 4, which is not easy and often results are not obtained Expected [27].

2.5 Level 5.- Optimized

Organization has fulfilled the majority of the above and can now start to see designs in execution after some time, so it can change its procedures keeping in mind the end goal to enhance productivity and diminish absconds in software development over the whole company (ISTQB, 2013)[26].Primarily at this level it is intended to improve the quality of the organization's processes, but in order to achieve this improvement, it is necessary to identify the causes of the variation, determine the root cause of the identified conditions, test the process improvements, Incorporate improvements and corrective actions into the standard processes of the organization. Its objective is the selection and deployment of incremental and innovative improvements that improve the processes and technologies of the organization [27].

3 SUSTAINABLE SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) PHASES

The Sustainable and Green Software Development life cycle Phases

tries to refine the distinctive stage of Software development based on the Standard Software development process which is CMMI, by presenting an arrangement of recommendations that can be followed in each of the stages specifically requirement and gathering, design, coding, testing and implementation.

3.1 Requirement Engineering Perspective

Collection the requirement in process of software development is an essential viewpoint for software advancement organizations. It is including deduction, examining, recording and keeping up the intricate arrangement of prerequisites for a software system [30].

In the Requirements Collection stage, it is important to consider the software useful life. The useful life of software is period amid which the product serves the society to reach a specific objective. On the other hand, it causes assist to improve the software product which it could keep running specially on current legacy hardware and keeps away from hardware disposal. Moreover, the Requirements collection process ought to likewise contain devices switch off, screens or working in low power mode when not being used.

Based on Mohankumar and Anand [31], this will assist to system in order to decrease pointless power consumption. In this procedure, anything may turn out badly or requirement may not be gathered according to the client particular, this it will make a significant issue in future and this may influence the sustainability dimension in software such as economy, social and environment.

Requirement gathering is a vital function, when one gathering the requirement that time itself one need to guarantee about software sustainability, although it is functional or nonfunctional

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

requirement. In the event that one gathered the requirement according to Green and Sustainability based then one can give a mark for that activity process. This procedure is proportional to the CMMI procedure of the Initial level as first stage.

3.2 Design

The second essential stage of the SDLC stage in view of the requirement gathered from the client organizations that need to make a model design to demonstrate the client. In the Design stage, the essential objective ought to be to reach simple design [25]. The system design must be made as straightforward as could be allowed [32]. The stakeholder has to have ability to find the design effectively without much clarification. Dull change in design can demonstrate costly as far as exertion, based on environment friendly not be considered and resources utilized as a part of all different stages of software development life cycle [5.8].

Penzenstadler [8] presented design pattern is a general reusable answer in order to usually happening issue for the process of software design in software engineering. Larsson [33] showed the considerations on sustainability and Green in the design

process. Mahmoud [9] investigated that a solution environmental issues with the utilization of virtualization with Green. Nevertheless, the utilization of virtualization with Green additionally brings up many research oriented questions for future examination. Software technology designers are in charge of the long-term outcomes of design of software. There is a discernment that sustainability is different disciplines of research and efforts with few characterized associations with software whereas sustainability is an unavoidable worry that converts into teach particular inquiries in every fields it applies [34]. Programmers have to write efficient algorithms through writing a reduced outline design of data and codes. Structure is according to the architecture of the hardware, programming language and application, however optimization originates from just through work experience. Business plans redesigning and processes of business based on the standards of adequacy rather than expanding port and resources decoupling economic development [30]. If based on per level of sustainability, a created design is satisfied, then one need to give the second mark for that organizations. It is comparable to the repeatable procedure through the system design, database, Database executive, architecture the developer and architect they require to work for this procedure need to meet the consumer loyalty level. In the event that they fulfilled in the purpose of Green and Sustainability condition here the designer, Database executive and architect need to give more focus for execution of the database, application efficiency and speed of server.

3.3 Coding

Coding is the most critical stage in SDLC. The coding technique will contrast from experienced developer to new developer. The experienced developer will create or write a code in improved style, however fresh developer write in their own particular style which is not legitimate enhancement of coding methodology. Hence, one need to consider this viewpoint additionally to develop the coding and the design [31]. This procedure is explained in three perspectives which are function, system, and time horizon. Function with low effect of environment and adequate economic adjust, System is a software development organization and time horizon relies on upon the organization measure and the general span of the undertakings [13]. Being developed process the organization needs to think about energy consumption, energy efficiency and execution according to suitability of coding development.

According to Noureddine et al. [35], the expanding computers using and other devices of electronic for example sensors and smart phones are persistently affecting ones' general energy consumption. By bringing energy costs up in cell phones and in computer which infers the enhancement and the adjustment of computer system with the perspective both the organizations and the developers actualize the sustainability. The coding has Various style from developer in compare to other developers, if the code is written in upgraded level and furthermore it will consider about the Green and Sustainability approach. Then one will give the third mark to that organization, for keeping up the entire coding as spotless code. The newcomers for that organization also needs ability to follow the explained code.

3.4 Testing

Next stage after design and coding, exercises got over in the sustainable according to SDLC, Testing is critical stage of next essential function. The software system is checked and approved for its accuracy in the testing phase. Unit tests is a noteworthy part in reaching deformity free software at a more granular level. minimize the deformities, minimize the change required in the software

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

product. Test Automation have to be supported as they diminish manual mistakes. They additionally underline on reutilization of test cases and institutionalize the testing procedure. In the manual testing process, this not just enhances the exactness of testing additionally the productivity and decreases power devoured by extra resources [36].

Resource profiling and execution testing have to likewise be advanced as a required stride in the testing procedure. Otherwise, there might be extra request on, processor, hardware memory and cycles. It might not be capable capacity accurately with legacy or current systems that are lower or average level based on hardware resources [37]. AS Kan [38] investigated that testing processes almost decides the product quality, testing the organization discharged the item with according to the quality of software and afterward one will give the fourth marks for that organization. It is equal to the procedure of CMMI level Managed for this reason after discharging the software, the organization needs to maintain and to deal with the application according to Sustainability and Green approach.

3.5 Implementation

Implementation is last phase of the Sustainable SDLC stage, this stage is an essential stage where the design and requirements are epitomized together to create a system working. The organization is executed the refreshed version or new application should be discharge for maintaining the current application in this stage of activity [5, 39]. Hence, they might discharge to assessment such as Digital Versatile Disk (DVD), Compact Disk(CD) and new client manuals in defining how to utilize the application. When other updating version must be discharged, the organization needs to guarantee what they are do with the old one and for that what are the safety measure measures is executed to keep up that sustainability.

Shenoy and Eeratta [10] mentioned that in the implementation stage, the initial step almost goals at follow the perfect procedures of the software development process in the organization. However, in order to upgrades the quality of the software product, environment friendly or sustainable implementation must likewise be another vital objective. Application Programming Interface (API) that are hardware particular ought to be debilitated for sustainable software implementation utilization. This is on account of the software product implementation is coupled to a specific sort of hardware. The weakness here is that if the hardware is old the software become noticeably pointless. A substitute approach is support constructing a hardware deliberation layer which decouples the software product system from the hardware related complexities. Resource serious API and Programming interfaces and programming builds like utilization of Mutex for synchronization of strings inside a procedure, utilization of settled circles, holding a database association for a long period, maintaining excess duplicates of data and so on have to be also preventd. These API have a tendency to expend extra resources as far as memory, hardware, processor cycles, disk space and so on. This makes the software non-scalable and non-functional with legacy hardware or existing resources. A way to deal with keep away from such circumstance could be to profile the resource devouring API and utilize them wisely. repetitive implementation automation assignments may be considered as essential phase of sustainable implementation since they diminish the time taken to accomplish the errands and dodge manual mistakes beyond what many would consider possible. Automated tools which are automatic code survey tools, automatic code generation tools and so on. Match Programming may also be implemented to diminish the quantity of resources as far as PCs or whatever other resources that should be utilized [40]. These methodologies can prompt better understanding and enhanced the software implementation. If the study is followed the sustainability method in related to implementation, then one may give the last mark for that organization and equal to last level of CMMI.

4 CONCEPTUAL RESEARCH MODEL

The conceptual model is a conceptual model that shows the logical sense behind the relationships among different factors under study [41, 42,43]. Such framework helps in developing the hypothesis as well as understanding the situation. According to Collis and Hussey [44], a theoretical framework would allow the measurement and observation through assigning values to the variables. The gaps found in the literature limited our understanding of the issue discussed in this research. Table 1 presents definition of construct used in this paper.

Based on our theoretical foundation that in the last section was discussed, the research conceptual model for this study is proposed which is shown in Figure. 1. As can be seen, it shows the how the CMMI levels model is working equivalent to the SDLC approach and shows the circle values for each stage.

Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

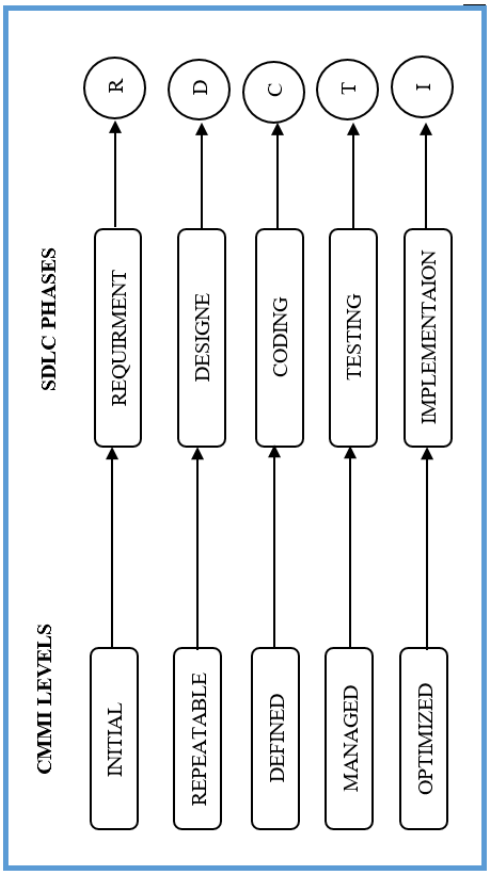

Figure 1: Conceptual SDLC Model

5 METHODOLOGY

The purpose of this study is to identify and propose the sustainable SDLC process in Malaysian based on current theoretical studies such as [5, 13, 7] and so. Regarding this issue, this study is used two methods to access the goal. In first step is used Standard Software development process method which is Capability Maturity Model Integration (CMMI) to institutionalization the model as discussed earlier. The next step is used survey sample papulation method in order to deeply identify and investigated the Sustainable SDLC process.

In the current study, the researcher used a survey method in order to collect data from the participants of the study [45,46,47]. The questionnaire content was established by referring to different theoretical findings and adapting several existing surveys which emphasize on the SDLC processes. To this aim the questionnaire which has been adapted from several sources has been used [36, 10] and so on. The convince sampling used to collect data from 102 participants in a period of three month. For analyzing the findings SPSS software version 20 has been used and descriptive statistics. Table 1 presents the objectives and sources for each question which is consists of 20 questions.

Table 1:. Objective and Sources for Each Question

Phases	Practices	Sources
Requirement	Collect requirements through electronic means	[36]
	Consider software Durability	[10]
	Provide power down capability	[10]
	Feasibility studies must assess the impact of the project on the environment	[36]
Design	Designs must include specific measures and practices that relate to environments	[36]
	Start designing with simple initial design	[10]
	Using fewer hardware	[10]
	Use of virtualized systems	[10]
	Avoid frequent changes	[10]
Coding	Implementing pair programming (two programmers working together)	[40, 45, 10]
	The focus should shift to writing energy efficient codes that minimize the use of system resources.	[36]
	Do not use duplicated code	[10]
	Use automatic code generation	[10]
Testing	The system resource usage metrics need to be added to the list of test cases that need to be passed	[36, 10]
	Use Automatic Testing	[10]
	Should concern reuse test	[10]
	Auditing and reporting the organization's energy consumption and savings	[31]
	Modifying supply chain,	[31]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	production activities and organization flow	
	Creating more efficient business operators, buildings and system	[31]
	Helping the decision-making process by analyzing modeling, and simulating of environmental impacts that may occur	[31]

6 ANALYSIS OF SUSTAINABLE SDLC MODEL

This section basically focuses on the sustainable software development life cycle (SDLC) and each step of them such as Requirement, Design, Coding, testing and Implementation process. It was answered by the respondents who had previous knowledge in it.

Most of the questions in this section relates to the software practitioners' perception on the importance and most consider of each phase of SDLC process in providing sustainable software view, in addition their knowledge, familiarity and experience with this approach.

This segment is essential in order to providing an understanding on the important of sustainable that have influence on software development life cycle of software.

A sum of 102 surveys are usable for this research, which is adequate based on [49, 50]. Distinctive methods are utilized to collect the data, in particular Manual or face-to-face gatherings, mail postages and online overview. Face-to-face meeting was utilized to guarantee that the respondents obviously understand each question and answer them appropriately. Besides, if they have doubts on any question, they can quickly ask for clarification. Although, the majority of the respondents liked to answer the survey by mail postage or online rather than face-to-face. In this manner, an online survey was made by utilizing the google docs and the connection was messaged to the corresponding respondents who consented to take an interest in the survey.

The survey was posted for three months in the Internet within December 2016 till February 2017. The response rate is higher compared to the face-to-face meeting via using this progressively popular method of data collection, and also reduced the cost and might be done faster.

More than one month was apportioned for the respondents to give back the surveys. Updates were sent to the ones who neglected to do as well. Unlikely, there were 28 unreturned surveys, while 18 were rejected because of incomplete replies. Table 3 is showed the outline of the respondents for this study and based on it, the response rate was high with (69%) participation.

Table 2: outline of the respondents

Details	Number	Percentage
Number of respondents seeking to participate	148	100
Online respondents	75	75
Unreturned Surveys	28	19
Manual or Face-to-face respondents	17	17
Mail postage respondents	10	8
Incomplete survey	18	12
Total usable	102	69

The respondents were further inquired about the SDLC practices that need to be performed in order to produce high green software. Survey was categorized by Requirement, Design, Coding, testing and Implementation process. The mean value for each practice is obtained from the analysis, as it represents the most selected answers in average. The 5-point numerical scale was used for these questions, which ranges from Not Important to Most Important. The scale is then mapped to equal intervals, by using SPSS Analysis. This scale was then mapped to equal intervals. The interval ranges are calculated by using the following formula [51]:

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

Where n is the maximum number in the used scale, which is to equal 5. Thus, the interval size of the consideration level between one through seven is 0.8, as the interval values are depicted in Table 3.

Table 3. Interval Values

Degree of Importance (DI)	Interval Values
Not Important (NI)	1.00 – 1.80
Less Important (LI)	1.81 – 2.60
Natural (N)	2.61 – 3.40
Important (I)	3.41 – 4.20
Most Important (MI)	4.21– 5.00

Table 4 shows the mean values obtained by the important green practices in each phase of the SDLC processes. Results from the study show that mostly these SDLC processes obtained high

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consideration among the respondents, whereby the mean values are in the range of Important to Most Important.

Table 4: The SDLC Practices Values

Phases	Practices	Mean	DI
Requirment	Collect requirements through electronic means	4.20	I
	Consider software Durability	4.28	MI
	Provide power down capability	4.20	I
	Feasibility studies must assess the impact of the project on the environment	4.22	MI
Design	Designs must include specific measures and practices that relate to environments	4.32	MI
	Start designing with simple initial design	4.18	I
	Using fewer hardware	4.23	MI
	Use of virtualized systems	4.25	MI
	Avoid frequent changes	4.25	MI
Coding	Implementing pair programming (two programmers working together)	4.35	MI
	The focus should shift to writing energy efficient codes that minimize the use of system resources.	4.22	MI
	Do not use duplicated code	4.22	MI
	Use automatic code generation	4.32	MI
Testing	The system resource usage metrics need to be added to the list of test cases that need to be passed	4.14	I
	Use Automatic Testing	4.04	I
	Should concern reuse test	4.19	I
Implementation	Auditing and reporting the organization's energy consumption and savings	4.23	MI
	Modifying supply chain, production activities and organization flow	4.14	I
	Creating more efficient business operators, buildings and system	4.01	I
	Helping the decision-making process by analyzing modeling, and simulating of environmental impacts that may occur	4.23	MI

Also, as can be seen in Figure 2, the respondents were then asked about the frequency of the SDLC being practiced in their organizations. As can be seen in Figure 4.6, the results demonstrate majority of origination none or few time used of the SDLC that principles were used 'Not Apply' and 'Seldom Apply' by the respondents. Nevertheless, minority of the principles were implemented that the respondents performed 'Totally Apply'.

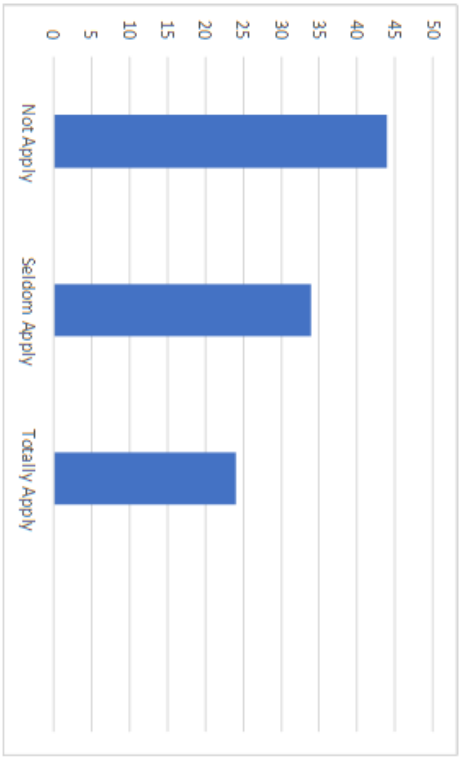

Figure 2: SDLC Implementation

The Green and Sustainable SDLC Model tries to correct the distinctive stage of Software development based on the Standard Software development process which is CMMI, by presenting an arrangement of recommendations that can be followed in each of the stages.

Based on our current review paper Raisian et al. [5] and last section, the Figure 2 shows how to implement the sustainability in SDLC phases in software engineering. Depending up on the implementation one will consider the Green and Sustainability rating of the company.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

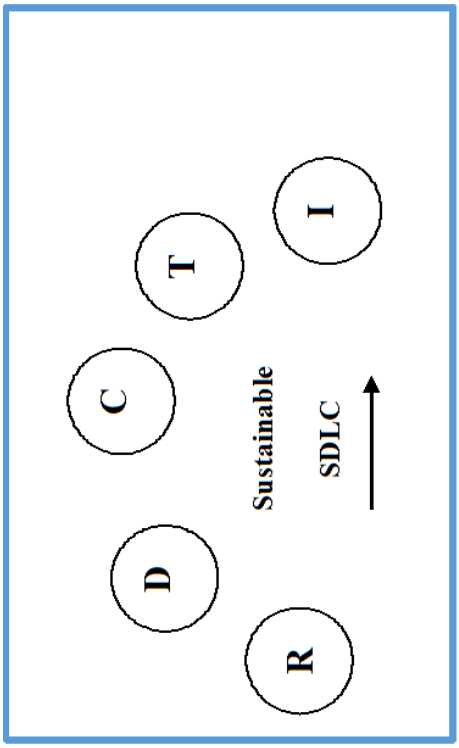

Figure 3. Sustainable SDLC Model

7 DISCUSSION AND CONCLUSION

Nowadays the science of software is a significant component in a society of advancing technological. Up to now several relationships between Information and Communication Technology (ICT) and Sustainable Development (SD) are published. Based on latest research and specially our review research which was current challenges of green and sustainable software engineering, a few studies have been resulted on the development phase of sustainable software. So, the purpose of this research was to investigate the significance of sustainable software development life cycle process and green software engineering and applied model. In addition, the study tries to answer how and which standard software development methodology can be suitable to accomplish the Green and sustainability in software SDLC Process. Based on the goals, the study firstly is used Standard Software development process method which is Capability Maturity Model Integration (CMMI) to institutionalization the model. Next, the research is used survey sample population is selected equal with 102 responders that is Non-probability or not randomly obtained from international software organizations in Malaysia to identify main stages of the Green Software Development Life Cycle and finally is investigated based on that. In future research directions regarding to green and sustainable software, we need to concern and provide critical sustainable measurements and getting values for each stage and the process of SDLC.

ACKNOWLEDGMENT

First and foremost, praise be to Almighty Allah for all his blessings for giving me patience and good health throughout the duration of this research. I wish to thank my supervisor, Co-SUPERVISOR and my family for their love, support, encouragement and sacrifice throughout the course of this study. Next many thanks to Malaysian Ministry of Higher Education. This research is funded by Malaysia Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2015/ICT04/UKM/02/1) and DPP-2015-FTSM. Last, but not least, I would like to thank my beloved wife for standing by me no matter how rough the sea and the journey has been. She is the only one that can attest to all sacrifices we both had to make, so that this manuscript can become a reality.

REFERENCES

[1] Katz DS, Choi SC, Niemeyer KE, Hetherington J, Löffler F, Gunter D, Idaszak R, Brandt SR, Miller MA, Gesing S, Jones ND. Report on the third workshop on sustainable software for science: Practice and experiences (WSSSPE3). arXiv preprint arXiv:1602.02296. 2016 Feb 6.

[2] Chitchyan R, Becker C, Betz S, Duboc L, Penzenstadler B, Seyff N, Venters CC. Sustainability design in requirements engineering: state of practice. InProceedings of the 38th International Conference on Software Engineering Companion 2016 May 14 (pp. 533-542). ACM.

[3] Stefan D, Letier E, Barrett M, Stella-Sawicki M. Goal-oriented system modelling for managing environmental sustainability. In3rd International Workshop on Software Research and Climate Change. Lancaster, United Kingdom 2011 Aug.

[4] Garg SK, Buyya R. Green cloud computing and environmental sustainability. Harnessing Green IT: Principles and Practices. 2012:315-40.

[5] Raisian K, Yahaya J, Deraman A. Current challenges and conceptual model of green and sustainable software engineering. Journal of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Theoretical and Applied Information Technology. 2016 Dec 1;94(2):428.

[6] Murugesan S. Harnessing green IT: Principles and practices. IT professional. 2008 Jan;10(1).

[7] Naumann S, Dick M, Kern E, Johann T. The greensoft model: A reference model for green and sustainable software and its engineering. Sustainable Computing: Informatics and Systems. 2011 Dec 31;1(4):294-304.

[8] Penzenstadler B, Bauer V, Calero C, Franch X. Sustainability in software engineering: A systematic literature review.

[9] Mahmoud SS, Ahmad I. A green model for sustainable software engineering. International Journal of Software Engineering and Its Applications. 2013 Jul;7(4):55-74.

[10] Shenoy SS, Eeratta R. Green software development model: An approach towards sustainable software development. InIndia Conference (INDICON), 2011 Annual IEEE 2011 Dec 16 (pp. 1-6). IEEE.

[11] Kaefer G, Green SE. Ideas for Including Energy Efficiency into your Software Projects, Technical Brief. In31st International Conference on Software Engineering, Vancouver, Canada 2009 (pp. 16-24).

[12] Herzwurm G, Pietsch W. 4 International Workshop on Requirements Prioritization and Communication (RePriCo). In19th International Working Conference on Require-ments Engineering: Foundation for Software Quality (REFSQ 2013) 2013 Jul (p. 101).

[13] Penzenstadler B, Femmer H, Richardson D. Who is the advocate? Stakeholders for sustainability. InGreen and Sustainable Software (GREENS), 2013 2nd International Workshop on 2013 May 20 (pp. 70-77). IEEE.

[14] Kern E, Naumann S, Dick M. Processes for Green and Sustainable Software EngineeringProcesses for Green and Sustainable Software Engineering. InGreen in Software Engineering 2015 (pp. 61-81). Springer International Publishing.

[15] Sahin C, Cayci F, Clause J, Kiamilev F, Pollock L, Winbladh K. Towards power reduction through improved software design. InEnergytech, 2012 IEEE 2012 May 29 (pp. 1-6). IEEE.

[16] Shojaee H. Rules for being a green software engineer. ship software ontime! The blog that helps you build great software,(2007).

[17] Dick M, Naumann S, Kuhn N. A model and selected instances of green and sustainable software. What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. 2010:248-59.

[18] Bordage F. Eco-conception web/Les 115 bonnes pratiques: Doper son site et réduire son empreinte écologique. Editions Eyrolles; 2015 Sep 18.

[19] Shaw M, Garlan D. Software architecture: perspectives on an emerging discipline. Englewood Cliffs: Prentice Hall; 1996 Apr 1.

[20] International Organization for Standardization (ed.): ISO 14040:2006 Environmental management - Life cycle assessment - Principles and framework. International Organization for Standardization, Geneva (2006).

[21] IEEE-SA, Basso, T.S. and DeBlasio, R., 2004. IEEE 1547 series of standards: interconnection issues. IEEE Transactions on Power Electronics, 19(5), pp.1159-1162.

[22] International Organization for Standardization (ed.): ISO/IEC 12207.(2008) Systems and software engineering - Software life cycle processes. International Organization for Standardization, Geneva (2008).

[23] International Standard Organization (ISO) (1996): Software life cycle processes—fit for purpose?. *Software Quality Journal*, 5(4), 243-253. ISO 12207, June 19967.

[24] International Standard Organization (ISO) (1997): Environmental management — Life cycle assessment — Principles and framework. ISO 14040, June 1997.

[25] Patel J, Lee R, Kim HK. Architectural view in software development life-cycle practices. InComputer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on 2007 Jul 11 (pp. 194-199). IEEE.

[26] Van Veenendaal E, Cannegieter JJ. Test maturity model integration (TMMi). TMMi Foundation. 2010.

[27] Morales-Fernández R, Brito-Rojas JA, Marcial JA. CAPABILITY MATURITY MODEL INTEGRATION (CMMI).

[28] Team CP. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD, V1. 1, Staged).

[29] Curtis B, Hefley B, Miller S. People capability maturity model (P-CMM) version 2.0. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst; 2009 Jul.

[30] Penzenstadler B, Raturi A, Richardson D, Tomlinson B. Safety, security, now

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sustainability: The nonfunctional requirement for the 21st century. IEEE software. 2014 May;31(3):40-7.

[31] Mohankumar M, Anand MK. A Green IT Star Model Approach for Software Development Life Cycle. International Journal of Advanced Technology in Engineering and Science. 2015;3(1):548-59.

[32] Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorenson W. Object-oriented modelling and design 1991;3(1): 45-9.

[33] Larsson P. Energy-efficient software guidelines. Intel Software Solutions Group, Tech. Rep. 2011 Apr.

[34] Becker C, Chitchyan R, Duboc L, Easterbrook S, Mahaux M, Penzenstadler B, Rodriguez-Navas G, Salinesi C, Seyff N, Venters C, Calero C. The karlskrona manifesto for sustainability design. arXiv preprint arXiv:1410.6968. 2014 Oct 25.

[35] Noureddine A, Bourdon A, Rouvoy R, Seinturier L. A preliminary study of the impact of software engineering on greenit. InGreen and Sustainable Software (GREENS), 2012 First International Workshop on 2012 Jun 3 (pp. 21-27). IEEE.

[36] Agarwal S, Nath A, Chowdhury D. Sustainable approaches and good practices in green software engineering. International Journal of Research and Reviews in Computer Science (IJRRCS). 2012 Feb;3(1).

[37] Jeffries RE. Extreme Testing Aggressive software development calls for radical testing efforts. How testing works in the world of extreme programming. Software Testing and Quality Engineering. 1999;1:22-7.

[38] Kan SH. Metrics and models in software quality engineering. Addison-Wesley Longman Publishing Co., Inc.; 2002 Oct 1.

[39] Voelter M, Groher I. Product line implementation using aspect-oriented and model-driven software development. InSoftware Product Line Conference, 2007. SPLC 2007. 11th International 2007 Sep 10 (pp. 233-242). IEEE.

[40] Williams L, Rubin K, Cohn M. Driving process improvement via comparative agility assessment. InAGILE Conference, 2010 2010 Aug 9 (pp. 3-10). IEEE.

[41] Raisian K, Yahaya J. A Research Framework for Environmental Quarry System.". In2nd International Research Management & Innovation Conference (IRMIC 2015) LANGKAWI 2015 Aug (pp. 26-27).

[42] Raisian K, Yahaya J. Security Issues Model on Cloud Computing: A Case of Malaysia. International Journal of Advanced Computer Science and Applications (IJACSA). 2016;6(8):216-23.

[43] Sekaran U. Research Methods for Business: A skill-bulding approach. New York: John Willey and Son. Inc Year. 2003.

[44] Collis J, Hussey R. Business research: A practical guide for undergraduate and postgraduate students. Palgrave macmillan; 2013 Nov 29.

[45] Raisian K, Yahaya J, Deraman A, Hamdan AR, Rais IA, Yahaya NZ. A model for environmental quarry system based on particles, vibration and noise components. Journal of Environmental Management & Tourism. 2016 Jul 1;7(2 (14)):185.

[46] Raisian K, Minouei A, Khosravi A, Hashemi A, Nilashi M, Ibrahim O, Zakaria R, Nazari M. Multi-Criteria Approach for Customer Trust Model in Internet Banking: A Case of UTM CIBM Bank. Life Science Journal. 2014;11(6).

[47] Raisian K, Ibrahim O, Nilashi M. Customer Trust Model for Malaysian Internet Banking. 2013.

[48] Schindler C. Agile software development methods and practices in Austrian IT-industry: Results of an empirical study. InComputational Intelligence for Modelling Control & Automation, 2008 International Conference on 2008 Dec 10 (pp. 321-326). IEEE.

[49] Saunders MN. Research methods for business students, 5/e. Pearson Education India; 2011.

[50] Fisher C, Buglear J. Researching and writing a dissertation for business students. Financial Times Prentice Hall; 2004.

[51] Ismail W, Abedlazeez N, Hussin Z. Epistemological beliefs of students at high schools: a survey study in Malaysia.