JOIV

Exploring Potential Factors in Green and Sustainable Software Product

Komeil Raisian[#], Jamaiah Yahaya[#], Aziz Deraman[*]

[#] *Faculty of Information Science and Technology, the National University of Malaysia 43600 UKM Bangi Selangor, Malaysia*
[*] *School of Informatics & Applied Mathematics, Universiti Malaysia Terengganu, Kuala Terengganu, Malaysia*
Email : raisiank@hotmail.com

Abstract— **software engineering is a principle role of our activity in social and economic life. Software engineers are starting to comprehend the significance of designing sustainability regarding software processes. The field of sustainable and green software and green software engineering is still young. Thus, in modern society, researches effort is mainly focused on green and sustainable software engineering itself. In other words, the aim of this study at hand aims an in-depth understanding of the related potential factors for the purpose of improving the recently reluctant trend of sustainability dimension in green software based on life cycle of green software product process. To reach this purpose, this study is reviewed latest research and used conceptual model. A conceptual model is rendered demonstrating the consolidated life cycles of sustainable product and principle sustainable measurement dimensions, for example, energy or information efficiency, low cost, Resource efficiency, Waste reduction, Equality support and human health.**

Keywords— **Software engineering, green and sustainable software, sustainable dimension, society, environment, economic.**

I. INTRODUCTION

Sustainability is currently a ubiquitous term in calls for research proposals and conference sessions (ICSE, CAiSE, RE, etc.). However, in literature, there is no overview of the current state of the art in supporting sustainability in software engineering research and practice[1].

According to current research by Raisian et al. [2] 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.

A sustainable world is broadly demand as one in which humans can survive without Compromising the continued survival of future generations of humans in a healthy environment [3]. This anthropocentric view of sustainability allows us to consider the implications of, and necessities for, human existence in the world.

Sustainability can be also discussed with reference to a concrete system such as an ecological system, a human net-work, or even a specific software system. Here, global sustainability implies the capacity for endurance given the functioning of all these systems in concert. Software Engineering for Sustainability has developed as a current focus of research as a result of software engineers engaging in issues regarding the impact of software systems on global sustainability.

Information and Communication Technology (ICT) represents around 2% of world CO_2 emanations [4], a number comparable to avionics, as indicated by Gartner. Actually, this 2% incorporates just the being used period of hardware: in the staying 98% software, both incorporates the individual business and also the public sector, and conveying end-client applications that pervade [5]. Software can add to diminishing energy utilization (i.e. ended up greener) by being more productive, consequently utilizing lesser power; or by making its upheld forms more sustainable, henceforth diminishing the ecological effect of governments, organizations and people utilizing software. While experts' opinions in gauging and organizing the level of greenness of hardware segments still is there, significant examination is required to associate make greener of hardware to its implementing software [1, 6, 7, 8].

The software engineering study area has as of late been considering sustainability, as the expanded number of papers, observational studies, and gatherings on the point exhibits that Greening in software plans to lessen the natural effect brought on by the software [9, 10, 11]. Penzenstadler [12] in comprehensive seminar address survey showing that software engineering is principle confinement of research which there is one technique in software engineering that expressly addresses sustainability. It is a reference model with particular application in web engineering [13]. So far,

not many studies have been led on the make greener of software itself. Hence, it is critical to examine the significance of sustainable software product and green software engineering and applied model.

II. SOFTWARE SUSTAINABILITY

There exist a few fields in which software sustainability should be connected: software systems, software products, Web applications, data centers and so forth. Different attempts are being made, however the greater part of this deals with data systems, which use fundamentally higher power than business office areas [14]. As noted in [9], the best approach to accomplish sustainable software is primarily by enhancing power utilization. Though hardware has been always enhanced in order to be efficient, software has not. The software advancement life cycle and related improvement devices and techniques seldom, if at any point, take into account energy efficiency as a goal [15]. Energy efficiency has never been a key necessity in the advancement of software-oriented advances, thus there is an extensive potential for enhancing efficiency.

As commented by [16], software assumes a noteworthy part, both as a feature of the issue and as a major aspect of the arrangement. The conduct of the software has critical impact on whether the energy saving components incorporated with the platform are powerful [17]. In [15], it is said that 'the term Sustainable Software can be translated in two ways: [18] the software code being sustainable, rationalist of reason, or [9] the software intention being to bolster sustainability objectives.

As per [19], sustainable software is 'software, whose effects on economy, society, people, and environment that outcome from improvement, sending, and use of the software are insignificant and/or which positively affect sustainable advancement'.

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 [13]. They consider that immediate effects are identified with assets and power utilization amid the creation and utilization of software, when marginal effects will impact the software product use, together with different procedures and long haul systemic impacts.

A thorough definition is provided by [20] that implies that green and sustainable software is software:
- Direct and indirect utilization of environmental energies, that emerge from arrangement and use, are observed, consistently measured, assessed and enhanced as of now in the development process.
- Appropriation and use fallout can be persistently assessed and advanced
- Development and generation forms consistently assess and minimize their immediate and backhanded utilization of regular power and resources

One more relevant definition is sustainable computing processing. It is utilized to exchange the political idea of sustainability to computers, including material segments (hardware) and also instructive ones (software); it incorporates advancement and in addition utilization forms [21].

This concept is particularly essential on account of software, in light of the fact that different researchers, for example, [13, and 20] use both terms equally. Taking into account [8], this methodology is flawed and that it should be dodged, since we are discussing two unique ideas, as will be seen at the appropriate time. What seems genuine, notwithstanding, is that software sustainability, albeit still in its initial stages, is a critical exploration subject that will be of incredible significance in the following couple of years. All things considered, general work on its centrality is required.

III. THE LIFE CYCLE OF SOFTWARE PRODUCTS PHASES

A. The development phase

Environmental impacts to be considered include, for example, electrical energy that is necessary to power the workstations of software developers and other employees, electrical energy and natural resources that are necessary to operate the IT infrastructure (e.g. networking devices, servers, and storages), energy that is necessary for heating and air conditioning, electrical energy that is necessary for offices lighting, or energy for transportation purposes like long distance business trips for meetings with customers and the development team and even the employees 'daily way to work.

Social impacts can be working conditions and payment of offshore workers (e.g. developers, type setters), which have consequences for the workers and their families. Some of these impacts can be mitigated by introducing teleworking and teleconferencing, or by replacing material products with adequate immaterial substitutes (second-order effects). This in turn may induce, e.g. changes in organizations, software development methods, or life styles (third-order effects).

The development phase also accounts for impacts from software maintenance in the sense of bug solving, because this is also a software development activity and therefore should belong to the development phase.

The development phase also (ecological (environment), human (social), and financial resources (economic)) ((17).

The distribution phase and the disposal phase:

This includes environmental impacts, e.g. of printed manuals (type of paper and ink), chosen means of transport, type and design of the retail and transport packaging (e.g. plastic, polyurethane foam, biodegradable material), or data medium (e.g. CD/DVD, USB memory stick). Furthermore, if the software product is offered as a download, which is common today, then its download size should be considered, as well as the electrical energy and material resources that are necessary to operate the required IT infrastructure.

The Disposal Phase accounts for impacts on SD that result from disposal and recycling of the afore mentioned material sub products.

B. The Usage phase

Software usage has several first-order effects regarding environmental sustainability. As an effect, properly trained users might need less time to complete tasks (which results

in less energy consumption), configure the software system in a way that it consumes less power, or just switch their computer to suspend mode when they leave their workplace.

Where social and environmental standards are very low, leads to considerable social and environmental impacts and using green business processes, which sometimes even culminate in armed conflicts [22].

C. The deactivation phases

If a software product is taken out of service, it is mostly necessary to convert the available data to a format that can be processed by the succeeding software product, or to make it accessible in some other ways. If the data cannot be converted easily, e.g. because it is stored in a proprietary data format, this may have an impact on economic sustainability of an organization. In this phase, even the backup size of data matters, e.g. if legal regulations require long-term storage of data.

IV. MAIN MEASUREMENTS OF SUSTAINABILITY DIMENSIONS IN SOFTWARE ENGINEERING

An aspect of the software sustainability is the software engineering sustainability. In the area of software engineering, very few recommendations have handled the idea of sustainability [15]. In a late overhaul of this work, the researchers monitored that the quantity of proposition has expanded impressively throughout the most recent 2 years [23]. This serves to show that there is a steadily developing worry to handle sustainability with regards to software engineering.

Sustainability ought to by and large be considered from the main phases of software improvement. That is not generally practical, since it is difficult to change how engineers work. Additionally, there is little direction on how software engineering can add to enhancing the sustainability of the frameworks a work in progress [21]. In this work, the creators consider five measurements of supportability that are essential for the examination of software frameworks:

A. Social sustainability

Social sustainability is a common factor for all aspects that evaluate how software supports social equality. Social Sustainability can be a factor of software engineering and software benefit.

As for any other product, the social sustainability of software can be considered in its production, use, maintenance, and disposal stages

This implies keeping up social sustainability and saving the solidarity of societal groups. Social sustainability is speculations and administrations that make the fundamental structure for society [23]. For Software Development (SE), there is a question that: 'What impacts do software systems have on society (e.g. correspondence, cooperation, government)?' so as to create socially sustainable software, the product engineers require a method for surveying, all through the improvement procedure, the impacts that the developed software will have on social sustainability of its planned clients.

Human health is principle perspective from society circle of triple base of Sustainability. Social circle has an extensive variety of variables. Human health and security covers the vast majority of the components from social circle. The effect of human health and security can consider social joy, better way of life, appropriate working environment, and upgraded moral qualities among society. Along these lines, the organization can add to the general public emphatically and accomplish better sustainability.

Green education: Software support for green education: Use software to support education and general knowledge about sustainable development and climate change [24].

Equality support: Equity category includes indicators that should reveal equality measures to all people regardless of their age, gender, ethnicity and social status [25].

B. Economic sustainability

This expects to look after resources. Resources incorporate sustainability as well as included quality. This requires a meaning of wage as the "sum one can expend amid a period and still be well off toward the end of the period, as it declines on using included worth (interest), as opposed to sustainability " [1, 23]. For SE, there is a question that: 'In what capacity can software systems be made so that the partners' long haul speculations are as protected as could be expected under the circumstances from financial dangers?'

Low cost: Cost is principle aspect from monetary circle of triple base of Sustainability. As indicated by the investigation by Corbett [25], the most ordinarily referred to driver for reusing Green IT practices is sparing expenses. Particularly in times of monetary emergency, cost lessening turns into the most essential financial goal [1, 26] of numerous organizations. On the off chance that green practices don't prompt an express (and critical) diminishment of costs, natural objectives are regularly viewed as a pleasant discretionary reward as opposed to an unquestionable target [27].

C. Environmental sustainability

Tries to enhance people's welfare by ensuring maintenance environmental resources, for example, water, land, air, minerals and biological system management; consequently, much is changed over to fabricated or financial sustainability. Environment incorporates the wellsprings of crude materials utilized for human needs, and in addition guaranteeing that sink limits reusing human squanders are not surpassed [28, 29].For SE, we suggest the conversation starter: 'How does programming influence the earth amid, inter alia, improvement and support?'

For SE: How can software be made with the goal that it can without much of a stretch adjust to future change?

Data accessible and power usage are primary perspective from environment circle of triple base of Sustainability within [29]. This implies any investigation of the issue, power or data required in a software engineering environment ought to be aware of the potential effect on different parties.

As commented in the Intel specialized article Impact of Software on Energy Consumption, a great part of the PC power utilized (and spared) depends on the adequacy of settlement too, somehow: while performing a "workload" (is the measure of handling that the computer has been given to do at a specific period .

Resource efficiency: In software engineering, resource efficiency is related to software life cycle, including

software design, management, maintenance, and disposal. Software execution resource efficiency is related to software execution and software platform usage [1, 23].

Waste reduction: Minimising waste is the first and necessary step towards getting rid of waste. Waste reduction is a common factor for all aspects that evaluate how software reduces waste and it can be a factor of software engineering and all software processes and life cycle product, actually, Software itself is an object.

All software processes need to be resource efficient and minimise waste. Taina [30] defined feasibility factor that It defines how resource efficient it is to develop, maintain, and dispose software. Software engineers and managers have already optimised used processes to excellent production efficiency, waste reduction, and process and software reusability. For instance, based on Poppendieck [31], the very good idea of pure software engineering is to reduce waste. Its methods clearly follow green software engineering principles. Green software engineering can add maximum resource efficiency and improved maintainability and configurability to traditional software engineering product [32].

Methodology

The purpose of this study is to identify and explain the potential factors that effect on software development based on sustainable perspective. In the current study, we follow the definition of sustainability software based on RAISIAN et l., [2]. To the best of researchers 'knowledge, as of now, there are lack of studies that investigate sustainability dimension in software development life cycle process.

According to the definition of Sustainability Software, we confine our search to obtain the suitable research papers for our study.Hence, the papers that was not contained our Sustainability Software definition was excluded from our review analysis. To obtain a comprehensive bibliography of research papers on sustainable development in software life cycle process the following electronic journal databases were searched:

- Science direct
- IEEEXplore Digital Library
- SpringerLink
- Emerald
- ProQuest
- JSTOR
- ProQuest

VI. RESEARCH CONCEPTUAL MODEL

By using a single Research proposed model, you can simplify the process of the task implementation. By doing so, you can eliminate different issues and materials, which are not directly related to study, and using this way you can make research questions clear and you can provide a map for research [33, 34]. Different studies, which have been reviewed in the previous sections, supported the researcher to make proposed model of the current study [35]. Figure 1 shows research proposed model of this study. To obtain a sustainable software product any processes contributing to its life cycle should sustainable themselves. Figure 5 presents conceptual model of this study.
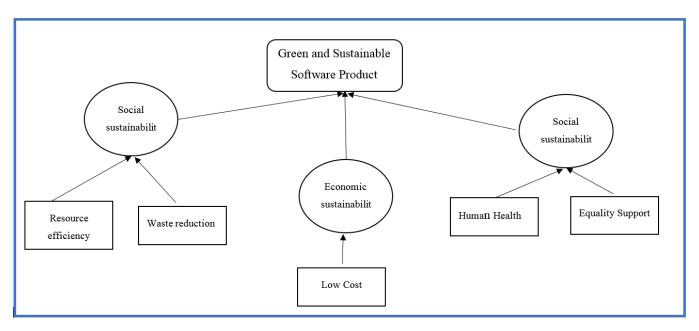

Fig. 1 Research Conceptual model

V. CONCLUSIONS

The majority of the endeavours spent on Green ICT/IT have been committed to tending to the impacts of hardware on environment, yet researcher have been focusing about the impacts of building software products also. In this way, the arrangement was to recognize current challenge in Sustainable software and explore the measurements of

sustainability and create green software product to presents a calculated model of sustainable software engineering product to end up greener of software itself. Subsequently, we proposed a strategy to coordinate sustainability in product life cycle. At last, a theoretical model is introduced demonstrating the consolidated life cycles of maintainable product and fundamental sustainable measurements

dimensions, for example, energy or information efficiency, low cost, Resource efficiency, Waste reduction, Equality support and human health. Currently, we are still in the early phase of the study. This didn't succeed to determine the dimensions of sustainability in software engineering product however the future research can look into this filed with analyzing the conceptual model and appropriate techniques to enhance it

REFERENCES

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

[2] RAISIAN, K., YAHAYA, J. and DERAMAN, A., 2016. CURRENT CHALLENGES AND CONCEPTUAL MODEL OF GREEN AND SUSTAINABLE SOFTWARE ENGINEERING. Journal of Theoretical and Applied Information Technology, 94(2).

[3] B. Brown, M. Hanson, D. Liverman, and R. Merideth, \Global sustainability: Toward de_nition," EnvironmentalManagement, vol. 11, no. 6, pp. pp. 713{719, 1987.

[4] Pettey C. Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions. Dostupno na: https://www. gartner. com/newsroom/id/503867. 2007 Apr 26; 14:2013.

[5] Bozzelli P, Gu Q, Lago P. A systematic literature review on green software metrics. VU University, Amsterdam. 2013.

[6] Betz S, Caporale T. Sustainable Software System Engineering. InBig Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on 2014 Dec 3 (pp. 612-619). IEEE.

[7] Kern E, Dick M, Naumann S, Filler A. Labelling Sustainable Software Products and Websites: Ideas, Approaches, and Challenges.2015.

[8] Calero C, Piattini M. Green in Software Engineering. Springer; 2015 Apr 3.

[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] Bener AB, Morisio M, Miranskyy A. Green software. Ieee Software. 2014 May;31(3):36-9.

[11] Dustdar S, Li F, Truong HL, Sehic S, Nastic S, Qanbari S, Vögler M, Claeßens M. Green software services: from requirements to business models. InProceedings of the 2nd International Workshop on Green and Sustainable Software 2013 May 20 (pp. 1-7). IEEE Press.

[12] Penzenstadler B, Bauer V, Calero C, Franch X. Sustainability in software engineering: A systematic literature review. InEvaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on 2012 May 14 (pp. 32-41). IET

[13] 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.

[14] Koomey J. Growth in data center electricity use 2005 to 2010. A report by Analytical Press, completed at the request of The New York Times. 2011 Aug 1;9.

[15] Penzenstadler B, Tomlinson B, Richardson D. Re4es: Support environmental sustainability by requirements engineering. InInternational Workshop on Requirements Engineering for Sustainable Systems 2012.

[16] Easterbrook SM. Climate change: a grand software challenge. InProceedings of the FSE/SDP workshop on Future of software engineering research 2010 Nov 7 (pp. 99-104). ACM.

[17] Steigerwald B, Agrawal A. Developing green software. Intel White Paper. 2011.

[18] Murugesan S et al Fostering green IT – guest editors' introduction. IT Prof 15(1):2013.

[19] Penzenstadler, B. and Femmer, H., 2013, March. A generic model for sustainability with process-and product-specific instances. In Proceedings of the 2013 workshop on Green in/by software engineering (pp. 3-8). ACM.

[20] Johann T, Dick M, Kern E, Naumann S. Sustainable development, sustainable software, and sustainable software engineering: an integrated approach. In2011 International Symposium on Humanities, Science and Engineering Research 2011 Jun 6 (pp. 34-39). IEEE.

[21] Rahman, N. and Akhter, S., 2010. Incorporating sustainability into information technology management. International Journal of Technology Management & Sustainable Development, 9(2), pp.95-111.

[22] Behrendt, S., Scharp, M., Erdmann, L., Kahlenborn, W., Feil, M., Dereje, C., Bleischwitz, R. and Delzeit, R., 2007. Rare metals: measures and concepts for the solution of the problem of conflict-aggravating raw material extraction; the example of coltan.

[23] Taina, J. and Mäkinen, S., 2015. Green Software Quality Factors. In Green in Software Engineering (pp. 129-154). Springer International Publishing.

[24] Al Hinai, M. and Chitchyan, R., 2014. Social sustainability indicators for software: Initial review.

[25] Corbett J. Unearthing the Value of Green IT. InICIS 2010 (Vol. 198).

[26] Fernández H, Procaccianti G, Lago P. Economic aspects of green ICT. InGreen in Software Engineering 2015 (pp. 107-127). Springer International Publishing.

[27] Raturi A, Tomlinson B, Richardson D. Green Software Engineering Environments. InGreen in Software Engineering 2015 (pp. 31-59). Springer International Publishing.

[28] Raisian, K., & Yahaya, J., 2015. A Research Framework for Environmental Quarry System. " 2nd International Research Management & Innovation Conference (IRMIC 2015) LANGKAWI, 26 – 27 AUGUST 2015 rmc.kuis.edu.my/irmic/e-proceedings/e-proceeding-irmic-2015.

[29] Raisian, K., Yahaya, J., Deraman, A., Hamdan, A.R., Rais, I.A.I. and Yahaya, N.Z., 2016. A model for environmental quarry system based on particles, vibration and noise components. Journal of Environmental Management & Tourism, 7(2 (14)), p.185.

[30] Taina J (2011) Good, bad, and beautiful software – in search of green software quality factors. Cepis Upgrade 12(4):22–27

[31] Poppendieck M., 2007. Lean software development. In: Proceedings ICSE COMPANION '07 Companion to the proceedings of the 29th international conference on software engineering, pp 165–166.

[32] Penzenstadler, B, and Andreas F. "Teach sustainability in software engineering?" In Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on, pp. 454-458. IEEE, 2011.

[33] Raisian, K. and Yahaya, J, 2016. "Security Issues Model on Cloud Computing: A Case of Malaysia" International Journal of Advanced Computer Science and Applications(IJACSA), 6(8), 2015.http://dx.doi.org/10.14569/IJACSA.2015.060829.

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

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