

Green Software Process Assessment: The Theoretical Framework

Siti Rohana Ahmad Ibrahim¹, Jamaiah Yahaya², Hasimi Sallehudin³, Aziz Deraman⁴

^{1,2,3}Centre for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

⁴Faculty of Ocean Technology Engineering & Informatics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia

¹are_na2305@yahoo.com, ²jhy@ukm.edu.my, ³hasimi@ukm.edu.my, ⁴a.d@umt.edu.my

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021; Published online: 05 April 2021

Abstract: Many different domains such as engineering, education and health apply green development through sustainability concept. The aim is to ensure the product and services are still relevance and applicable for the next generation. Previously, they were three common pillars of sustainability elements which were economic, social, and environmental. The technical and individual elements were added later to evaluate the software system. Yet, efforts are made to achieve green development process by reducing wastes and to preserve the environment. Most studies targeted on greenness of software products and less consideration on green in software processes. In this study, focus is given to ensure and guarantee green in software process activities. This paper presents the background works on existing studies in green software process and related. Further, it presents the theoretical framework that is based on literature findings and will be the groundwork of this study. The theoretical framework defined consists of software process, green context, and assessment method. The elements of sustainability and waste reduction will be identified further.

Keywords: green software process, green factors, sustainable dimension, software waste, assessment method

1. Introduction

Sustainable Development Goal (SDG) is a global development program that aims to provide well-being to the world community and preserve the environment (Wu et al, 2018). Nowadays, software engineering is one of domain that leading to sustainable development mainly in software process. Sustainability means the capacity to endure and for humans is the potential for long-term maintenance (Penzenstadler and Fleischmann, 2011). In software, sustainability is the capacity for a longer life of usage and provides greener characteristic inclusive. According to Dick and Naumann (2010), Kern et al. (2013), green software engineering goals are to minimize the negative effect on the environment and produce sustainable software products. Previous studies emphasised on green hardware in the aspect of energy efficiency, power consumption, waste reduction and disposal which were supported by Ardito et al. (2015), Islam et al., (2018), Kocak (2013), Shenoy and Eeratta (2011). However, literature study discovers that lack of studies on green software even though software also has the impact indirectly to the greenness and sustainability environment (Ibrahim et al, 2020).

Researcher suggested that software is no longer viewed in terms of productivity but needs to focus on scalability, usability and quality with recent advances in new technology (Casale et al, 2016). Basically, software development projects have many obstacles and barriers to complete and deliver the project. According to Taherdoost and Keshavarzsaleh (2016), several factors come from organization environment, team and user perspective that lack to well-prepared accordingly in term of requirement, budget and time.

This paper is organised as follows: Section II presents materials and methods which consist of the background of software process, green software process model and sustainability context. Section III presents the theoretical framework of green software process assessment, and Section IV concludes the paper with conclusion and future works.

2. Materials and Methods

Literature highlights that current works in software development do not focus on integrating sustainability measurement and software waste management to achieve green software process (Ibrahim et al, 2020). Previous studies have shown that reducing waste in processes will increase the efficiency and productivity of certain jobs (Taina, 2011). We will discuss the material supporting this study in several contexts: software process, green software process model, sustainability context, and software waste.

2.1 Software Process

Software Development Life cycle (SDLC) is a framework that reflect to activity at each phase of software development (Rastogi, 2015), (Hussain, 2016) and (Kramer, 2018). There are several other software development models available in literature such as waterfall, prototyping, incremental, spiral, Agile, XP, Rational Unified Process (RUP) and Rapid Application Development (RAD). The five basic and fundamental activities of software process are requirement analysis, design, implementation, testing, deployment and maintenance (Rastogi, 2015), (Hussain, 2017) and (Stoica et al, 2013).

Typically, in software development, the problems aroused in initial phase that involved the requirement scope and user understanding. Many developers have problems dealing to incomplete and change frequently of specifications and requirements that may lead to misunderstanding the user needs (Geogy and Dharani, 2016). The complex design occurs during software development may cause the conflicts between users and developers. The design should be simple and easy to be understood by the stakeholders and developers. In the implementation phase, the problem in term of the capability of programmers that do not have expertise and knowledge about programming used is the main challenge. In the testing phase, the selection of tests can reduce the error by using the automation testing. In addition, the re-use of test cases is encouraged to increase the efficiency of testing. In the maintenance phase, complaints for software issue often goes to the forms used which it should be done electronically (Shenoy and Eeratta, 2011), (Mahmoud and Ahmad, 2013). Based to the mentioned issues and problems could lead to the high consumption of energy, power and resources. It causes incompatibility and negatively effect to the environment.

2.2 Green Software Process Model

The GREENSOFT model is a conceptual reference model for green and sustainable software. It supports sustainability matters for software developers, administrators, and users. This model specifically covers the life cycle of a software product. The life cycle which was proposed by Naumann et al. (2011) was a different model from the traditional life cycle because it has the objective of sustainability that included the evaluation of the ecology, social, human compatibility and overall product economics. Another similar model was proposed by Shenoy and Eeratta (2011) for the Green Software Development Life Cycle Model. It aimed to reduce the carbon emissions of power, paper and electronic waste towards green and sustainable development in organizations. The Green and Sustainable Software Engineering Model by Dick et al. (2010) consists of two main parts, namely i) life cycle processes and models, and ii) guidelines and checklists. This process model serves to identify the current activity of the software product's life cycle and its' relation to the evaluation of a product's sustainability. In addition, the Green Model for Sustainable Software Engineering (GSEP) is a model that targeted for life cycle process and software tool (Mahmoud and Ahmad, 2013). There are two levels defined in this model, the first level provides a new greenness guideline and process in activities of SDLC. The second level offers a tool to support green software by optimizing the usage of energy resources. Basically, this model emphasises on environmental of sustainability aspect. However, these models by Naumann et al (2011), Dick et al. (2010), Shenoy and Eeratta (2011), Mahmoud and Ahmad (2013) did not focused on society, economic, technical and individual elements and waste reduction in development phases which we believed will contributing to production of sustainable software.

2.3 Sustainability Context

According to Lago (2015), sustainability is often associated with economic, social and environmental dimensions. However, Penzenstadler et al. (2013) argue that these three dimensions were insufficient and further added two more dimensions which associated with individual and technical. This is specifically for software systems sustainability measurements. In economic dimension, Venters et al. (2014), Amri and Bellamine (2014) stated that green software was applied to ensure the investment could be invested safely for a long time and free from economic risk. The factors of economic sustainability are cost reduction, productivity and usability. For the social dimension, software is developed to increase the value of capital such as maintaining social relations in society. The social dimension includes employee support and tool support (Al Hinai and Chitchyan, 2014). The environment dimension in software context is associated with environmentally friendly to create, use, maintain and dispose of with minimal impact, while technical aspect is referred to the design of the software to be easy to modify, easy to customize and useful for the future (García-Berna et al, 2018), (Lago et al, 2015). Whereas, the individual sustainability in software is to ensure that the people or team enjoy and satisfy their works with appropriate task, skill and experience (Calero and Piattini, 2015).

2.4 Software Waste

In general, waste is considered as irrelevant product or services to production line (Seadon, 2010). Waste formation is not just a form of public solid waste in the manufacturing industry (Abualfaraa et al, 2020) and construction (Dajadian and Koch, 2014) but it also occurs in software development (Al-Baik and Miller, 2014). In software process, waste can be classified such as excessive requirement and specification, out of scope design during design phase, and repeating testing process during testing phase (Shmueli and Ronen, 2017). Previous studies have identified software wastes during development process stated by Sedano et al. (2017), Shmueli and Ronen (2017), Santhiapillai and Ratnayake (2018), Mujtaba et al (2010). Some identified software waste are building wrong product, rework activity, waiting period, ineffective communication and software defect (Sedano et al, 2017), (Santhiapillai and Ratnayake, 2018). Previous study revealed that software waste needs to be eliminated accordingly by prevention, incremental improvement and removing waste by improving the productivity of software itself (Sedano et al, 2019).

3. The Theoretical Frame work

The theoretical framework is an overview of the definitions, theories and concepts involved in this study. The design of the theoretical framework is based on software process activities and sustainability elements. Figure 1 shows the theoretical framework of green software process assessment which consists of three main components: software process, green context, and assessment method.

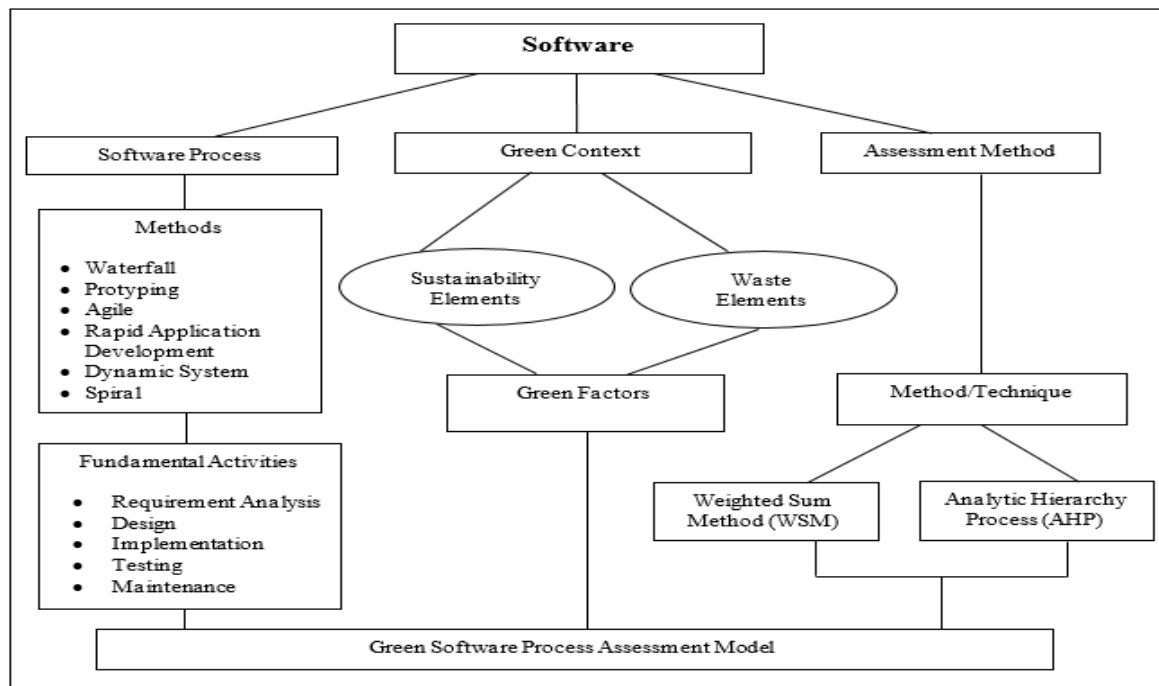


Figure 1. Theoretical framework for Green Software Process Assessment

A. Software Process

The first component defined in this framework is software process that involves five main phases: requirements analysis, design, implementation, testing, and maintenance. These fundamental activities are applicable in any methodology of software development. For example, waterfall model, prototyping approach, agile development approach, Rapid Application Development (RAD), and spiral model (Sommerville, 2016). For instance, in agile method, pair programming principle is considered as fulfilling green practices because it accelerates the programming phase and offers knowledge sharing among team members (Da Silva Estácio and Prikladnicki, 2014).

B. Green Context

The second component is the green context component that consists of sustainability and waste elements. These elements are essential in order to produce green factors for software process (Kern et al, 2013). The

sustainability element is the green factors that consists of resources, people, organizational, technical and environmental (Calero et al, 2014), (Penzenstadler and Femmer, 2013) (Al-Ta'ani and Razali, 2016). In order to produce green factors, the waste elements will be identified at each phases of software process and therefore requires to be eliminated (Murugesan, 2012). Thus, the green factors are one of the influential elements in green software process assessment model.

C. Assessment Method

The third component is assessment method. It defines weightage and significance of factors through a defined algorithm. Weighted Sum Method (WSM) and Analytical Hierarchy Process (AHP) are the two potential method and technique for assessment. It is useful to software practitioners as a guidance for assessment of green software process in their environments. The measurement instrument is needed to be designed and constructed. This requires analysing method using appropriate formulas and algorithm to determine the level of greenness in specific software process.

4. Conclusion

This paper has presented the theoretical framework for green software process assessment. It consists of three components which are software process, green context, and assessment method. This framework focuses on sustainability elements and waste reduction to improve green software process using specific assessment method. The theoretical framework was constructed based on literature study that applicable for green software assessment model development. Previous researchers believe that greener software process is valuable to guarantee the long life of software with green practices. The green assessment mechanism is one alternative to ensure the green processes are being practiced by the developers or practitioners in their organisations. For future work, the green factors of software process will be investigated further through experts' interview.

5. Acknowledgment

This research is funded by Malaysia Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UKM/02/1) and Universiti Kebangsaan Malaysia, The Research University Grant (AP-2017-005/3).

References

1. Abualfaraa, W., Saloniitis, K., Al-Ashaab, A., & Ala'raj, M. (2020). Lean-Green Manufacturing Practices and Their Link with Sustainability: A Critical Review. *Process Integration and Optimization for Sustainability*, 12, 981.
2. Al-Baik, O., & Miller, J. (2014). Waste Identification and Elimination in Information Technology Organizations. *Empirical Software Engineering*, 19(6), 2019–2061.
3. Al-Ta'ani, R. H., & Razali, R. (2016). A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 846-856
4. Al Hinai, M., & Chitchyan, R. (2014). Social Sustainability Indicators for Software: Initial review. *CEUR Workshop Proceedings*.
5. Amri, R., & Bellamine Ben Saoud, N. (2014). Towards a Generic Sustainable Software Model. *Proceedings - 2014 4th International Conference on Advances in Computing and Communications, ICACC 2014*, 231–234.
6. Ardito, L., Procaccianti, G., Torchiano, M., & Vetrò, A. (2015). Understanding Green Software Development: A Conceptual Framework. *IT Professional*, 17(1), 44–50.
7. Calero, C., Moraga, M. angeles, Bertoa, M. F., & Duboc, L. (2014). Quality in Use and Software Greenability. *CEUR Workshop Proceedings*, 1216, 28–36. <http://ceur-ws.org/Vol-1216/paper6.pdf>
8. Calero, C., & Piattini, M. (2015). Green in Software Engineering. In *Green in Software Engineering*.
9. Casale, G., Chesta, C., Deussen, P., Di Nitto, E., Gouvas, P., Koussouris, S., Stankovski, V., Symeonidis, A., Vlassiou, V., Zafeiropoulos, A., & Zhao, Z. (2016). Current and Future Challenges of Software Engineering for Services and Applications. *Procedia Computer Science*, 97, 34–42.
10. Da Silva Estácio, B. J., & Prikladnicki, R. (2014). A Set of Practices for Distributed Pair Programming. *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems*, 2, 331–338.

11. Dajadian, S. A., & Koch, D. C. (2014). Waste Management Models and Their Applications on Construction Sites. *International Journal of Construction Engineering and Management*, 3(3): 91–98.
12. Dick, M., & Naumann, S. (2010). Enhancing Software Engineering Processes Towards Sustainable Software Product Design. *Integration of Environmental Information in Europe*, 706–715.
13. Dick, M., Naumann, S., & Kuhn, N. (2010). A Model and Selected Instances of Green and Sustainable Software. *IFIP Advances in Information and Communication Technology*, 328, 248–259.
14. García-Berna, J. A., de Gea, J. M. C., Moros, B., Fernández-Alemán, J. L., Nicolás, J., & Toval, A. (2018). Surveying The Environmental and Technical Dimensions of Sustainability in Software Development Companies. *Applied Sciences (Switzerland)*, 8, 2312.
15. Geogy, M., & Dharani, A. (2016). A Scrutiny of The Software Requirement Engineering Process. *Procedia Technology*, 25, 405–410.
16. Ibrahim, S. R. A., Yahaya, J., Salehudin, H., & Bakar, N. H. (2020). Towards Green Software Process: A Review on Integration of Sustainability Dimensions and Waste Management. *International Conference on Electrical Engineering and Informatics (ICEEI)*, 128–133.
17. Hussain, A., Mkpojiogu, E.O.C., Yusof, M.M. (2016). Perceived usefulness, perceived ease of use, and perceived enjoyment as drivers for the user acceptance of interactive mobile maps. *AIP Conference Proceedings*, 1761, art. no. 020051.
18. Hussain, A., Mkpojiogu, E.O.C., Nawi, M.N.M. (2017). Capturing customer satisfaction and dissatisfaction in software requirements elicitation for features in proposed software systems. *Journal of Engineering and Applied Sciences*, 12 (21), pp. 5590-5597.
19. Islam, R., Abdul Ghani, A.B. (2018). Link among energy consumption, carbon dioxide emission, economic growth, population, poverty, and forest area evidence from ASEAN country. *International Journal of Social Economics*, 45 (2), pp. 275-285.
20. Kern, E., Dick, M., Naumann, S., Guldner, A., & Johann, T. (2013). Green Software and Green Software Engineering-Definitions, Measurements, and Quality Aspects. *1st International Conference on Information and Communication Technologies for Sustainability (ICT4S)*, 87–94.
21. Kocak, S. A. (2013). Green Software Development and Design for Environmental Sustainability. *In 11th International Doctoral Symposium an Empirical Software Engineering*. http://www.umbc.edu/eseiw2013/idoese/pdf/eseiw2013_IDoESE_183.pdf
22. Kramer, M. (2018). Best Practices in Systems Development Lifecycle: An Analyses Based on The Waterfall Model. *Review of Business & Finance Studies*, 9(1), 77–84.
23. Lago, P. (2015). Challenges and Opportunities for Sustainable Software. *IEEE/ACM 5th International Workshop on Product Line Approaches in Software Engineering*, 10–11.
24. Lago, P., Koçak, S. A., Crnkovic, I., & Penzenstadler, B. (2015). Framing Sustainability as a Property of Software Quality. *Communications of the ACM*, 58(10), 70–78.
25. Mahmoud, S. S., & Ahmad, I. (2013). A Green Model for Sustainable Software Engineering. *International Journal of Software Engineering and Its Applications*, 7(4), 55–74.
26. Mujtaba, S., Feldt, R., Petersen, K., & Ab, E. (2010). Waste and Lead Time Reduction in a Software Product Customization Process with Value Stream Maps. *Proceedings of the Australian Software Engineering Conference, ASWEC*, 139–148.
27. Murugesan, S. G. R. G. (2012). Harnessing Green IT Principles and Practices, *IT Professional*, 39-57.
28. Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). Sustainable Computing : Informatics and Systems The GREENSOFT Model : A Reference Model for Green and Sustainable Software and its engineering. *Sustainable Computing: Informatics and Systems*, 1(4), 294–304.
29. Penzenstadler, B., & Femmer, H. (2013). A Generic Model for Sustainability With Process- and Product-Specific Instances. *Proceedings of the 2013 Workshop on Green in/by Software Engineering - GIBSE '13*, 231–234.
30. Penzenstadler, B., Femmer, H., & Richardson, D. (2013). Who is The Advocate? Stakeholders for Sustainability. *2013 2nd International Workshop on Green and Sustainable Software, GREENS 2013 - Proceedings*.
31. Penzenstadler, B., & Fleischmann, A. (2011). Teach Sustainability in Software Engineering? *2011 24th IEEE-CS Conference on Software Engineering Education and Training, CSEE and T 2011 - Proceedings*, 454–458.
32. Rastogi, V. (2015). Software Development Life Cycle Models- Comparison, Consequences. *International Journal of Computer Science and Information Technologies*, 6(1), 168–172.
33. Santhiapillai, F. P., & Ratnayake, R. M. C. (2018). Identifying and Defining Knowledge-work

- Waste in Product Development: A Case Study on Lean Maturity Assessment. *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 834–838.
34. Seadon, J. K. (2010). Sustainable Waste Management Systems. *Journal of Cleaner Production*, 18(16–17), 1639–1651.
 35. Sedano, T., Ralph, P., & Péraire, C. (2017). Software Development Waste. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 130–140.
 36. Sedano, T., Ralph, P., & Péraire, C. (2019). Removing Software Development Waste to Improve Productivity. In *Rethinking Productivity in Software Engineering* (pp. 221–240).
 37. Shenoy, S. S., & Eeratta, R. (2011). Green Software Development Model: An Approach Towards Sustainable Software Development. *Proceedings - 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*.
 38. Shmueli, O., & Ronen, B. (2017). Excessive Software Development: Practices and Penalties. *International Journal of Project Management*, 35, 13–27.
 39. Sommerville, I. (2016). *Software Engineering* (10th edition). In *Pearson Education Limited, Essex England*.
 40. Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, 17(4), 64–76 .
 41. Taherdoost, H., & Keshavarzsaleh, A. (2016). Critical Factors that Lead to Projects' Success/Failure in Global Marketplace. *Procedia Technology*, 22, 1066–1075.
 42. Taina, J. (2011). Good, Bad, and Beautiful Software – In Search of Green Software Quality Factors. *The European Journal for the Informatics Professional*.
 43. Venters, C. C., Jay, C., Lau, L. M. S., Griffiths, M. K., Holmes, V., Ward, R. R., Austin, J., Dibsedale, C. E., & Xu, J. (2014). Software Sustainability: The Modern Tower of Babel. *CEUR Workshop Proceedings*, 1216, 7–12.
 44. Wu, J., Guo, S., Huang, H., Liu, W., & Xiang, Y. (2018). Information and Communications Technologies for Sustainable Development Goals: State-of-The-Art, Needs and Perspectives. *IEEE Communications Surveys and Tutorials*, 20(3), 2389–2406.