GLOBUS PUBLICATION

Globus An International Journal of Medical Science, Engineering and Technology
A Refereed Research Journal
Vol 12 / No 2 / Jul-Dec 2023 ISSN: 2319-8958

EMERGING ENGIGREENING: UNDERGRADUATE STUDENTS' PERCEPTIONS ON GREEN SOFTWARE DEVELOPMENT IMPLEMENTATION IN THE PHILIPPINE COMPUTER SCIENCE CURRICULUM

[*]Gionna Dalinas
[**]Alainne Remigio
[***]Ahyel Rodriguez
[****]Vin Lorenzo Yap

Paper Received: 02.09.2023 / **Paper Accepted**: 20.10.2023 / **Paper Published**: 22.10.2023
Corresponding Author: Gionna Dalinas; doi:10.46360/globus.met.320232005

Abstract

Information and Communication Technology (ICT) is emerging as the core of sustainable energy solutions, decreasing greenhouse emission rates, and providing efficient alternatives in everyday tasks. However, the existing Software Development life cycle process framework did not consider the energy consumption efficiency of these software technologies, affecting the ICT sector's emissions and energy consumption substantially. The pre-existing developmental gaps led to the sector's challenge of decarbonizing in line with the goals set by the United Nations to mitigate climate change. This study aimed to present factors in integrating green practices in local software development, focusing on methods to improve its energy efficiency and widen its reach in various demographics. The current state of software development in the Philippines was analyzed through a Merleau-Ponty phenomenology study of aspiring local software developers in analyzing software integration factors, user demographics, and energy-efficient development. Inclusion criteria were used to select software developers, and a series of interviews were conducted for data collection. Thematic analysis was used to produce codes and themes from the informants. It was determined that green software development in the Philippine curriculum was inadequate, with most developers and users unaware of its concept and the environmental impact of software. Furthermore, local software development remained client-focused and did not consider green solutions. Recommendations for future research include revisions in interview protocols and widening the scope of the participants.

Keywords: Software Development, Sustainability, Green Software, Merleau-Ponty Phenomenology, ICT.

Introduction

With the increased digital interactions in 2020, global internet traffic grew by more than 40%. ICT was still challenged to decarbonize like all sectors (Cunliff, 2020). Still, the data centers' energy usage had stayed between 1.0-1.4% of the world's energy consumption because the Information and Communications Technology (ICT) sector improved energy efficiency (IEA, 2021). Although energy consumption has improved, there are still issues concerning the energy efficiency of software with the emergence of technology and the internet of things (IoT).

Software might not have a physical form, but it could impact reality (Radzik, 2022). Green software development, an emerging field in software development, enabled financial and environmental efficiency. Sustainability spearheads the field of software development as various practices are improved through green innovations. The field was predicted to rise as climate consciousness and sustainable initiatives promote green innovations (Garnier, 2021). This meant that software technology would increase resource consumption over time.

ICT was one of the viable solutions for reducing emission rates, but its current software development process did not consider its energy consumption. As of 2020, global greenhouse gas emissions by ICT were estimated to be 2.8% and might help reduce emission rates by 15%. However, the existing Software Development life cycle process framework did not apply the Green Based Software Development life cycle process, which concerned the current energy consumption (Muthu et al., 2020). The most sought and popular programming language, python (Eastwood, 2020), was one of the most energy-inefficient software languages (Pereira et al., 2021).

*-****De la Salle University Integrated School - Laguna Campus, Philippines.*

Various literature provided models for green software development. Kern et al. (2018) developed a set of criteria to determine whether a specific software could be deemed "green" or not. Their model described the causal chains leading from software properties, natural resources, behavior and organization, and hardware. Furthermore, Muthu et al. (2019) proposed a new green-based software development life cycle in designing, implementing, and evaluating phases to reduce high-level energy usage and e-hazardous waste. Literature on developers and user behaviors were also reviewed. Elgaanied-Gambier et al. (2021) found that most consumers had a lack of knowledge of internet pollution and were reluctant to change their online behavior due to not understanding the severity of the environmental threat and attributing the responsibility to the companies to innovate the technology to reduce the carbon footprint and the government to regulate its usage. They added that consumers who were more aware of the severity of the environmental threat admitted that they carry responsibility.

Further, Verdecchia et al. (2021) also investigated the implementation of green software by integration in their practice. These energy optimizations may serve as possible strategies to ease energy conservation in local software development and insights into garnering a green workplace for the target developers. Similarly, Ibrahim et al. (2021) formulated a new model that included important factors and elements for green software. These factors and elements included resource, organizational, technical, environmental, and technology. In the Philippines, energy efficiency solutions via ICT were being considered, although specific populations had no access to this technology. Sabili & Ignacio (2020) explored the use of green technology in different institutions but did not focus much on the development of green software. The Department of Science and Technology (DOST) promotes ICT as the country boosts its energy efficiency (Patel, 2022). However, the lack of accessibility to ICT due to the scarcity of infrastructure affected a non-negligible population (Tabuga & Cabaero, 2021).

This research aimed to identify undergraduate students' perceptions of how Philippine Computer Science (CS) curriculums implement green software development in their courses. The main objective of this study was to determine whether undergraduate students were aware of green software development processes and were implementing them within academic premises, and if it was sufficient for them to be conscious of how beneficial and efficient it was to implement these practices. The study also followed three specific objectives to help achieve the main objective:

1. To understand the factors affecting the integration of green software development frameworks in the curriculum of Computer Science majors in the Philippines.
2. To determine the users' preferences for green applications through the lens of aspiring local software developers.
3. To discern energy-effective processes and suggest applicable green software development frameworks in the CS Curriculum for reduced resource consumption.

While research has been conducted on green software development internationally, there seems to be a literature gap for it in the Philippines. Specifically, the researchers found only a few studies conducted on green software developers in the Philippines. This study aimed to analyze the perceptions of Philippine undergraduate students studying Computer Science (CS) courses on the application of green software development in their curriculum. This research also brought new data through the use of thematic analysis with the perceptions of undergraduate students to fully understand their perspective with local software development.

This study provided localized information about the experiences of green software by Computer Science majors. This study may also aid the environment, as researching green software can bring awareness to a larger audience. This study can also benefit other future software developers as a basis for their approach to their software development. Specific institutions such as the Department of Education (DepEd) and the Department of Information and Communications Technology (DICT) could benefit from the findings of this study since it opens up future software development procedures in the country.

Theoretical Framework

The field of green software development is expanding with the advancement of technology in society. Ibrahim et al. (2021) state that applying green software development frameworks is centered more on factors that affect the environment, such as power consumption and energy efficiency. On other accounts, there have been models created to identify whether specific software could be deemed green or not, such as Kern et al.'s (2018) causal model that aims to assess software sustainability. This model contains different factors that cover important aspects of software development, such as configurability and default settings, hardware requirements, resource management functionality, and continuity and further development of the software product. This causal model can then be applied to investigate whether the current CS

curriculums in Philippine institutions are implementing green software development properly.

Information and Communications Technology is also an important factor to consider in green software development. Muthu et al.'s (2019) ICT for sustainability (ICT4S) is a process that could alleviate heavy manufacturing practices and make its patterns more viable by using the capabilities of ICT. Since the Philippines is starting to consume more ICT for different activities (Tabuga & Cabaero, 2021), this can be explored more in the study. Because of the criteria that set a standard for software sustainability, ICT and other factors can be mixed in to see if undergraduate students perceive that there is indeed a proper implementation of green software development in their institutions' CS curriculums.

The basis of this study stems from the ever growing field of software development and ICT in the Philippines. Growing concepts such as the Internet of Things, AI, and NFT's contribute to the growing consumption of resources and electricity of software development. This is why the emerging field of green software development is one of the solutions to this issue. It collaborates with both environmental factors and technical factors in software development to increase energy-efficiency when developing software (Muthu et al., 2019). These software can range from network systems, databases, websites, games, and more. Thus, if aspiring software developers are aware and are actively applying green practices and methods, the future of software development may be a cleaner one.

Conceptual Framework

Figure 1: Conceptual Framework Map

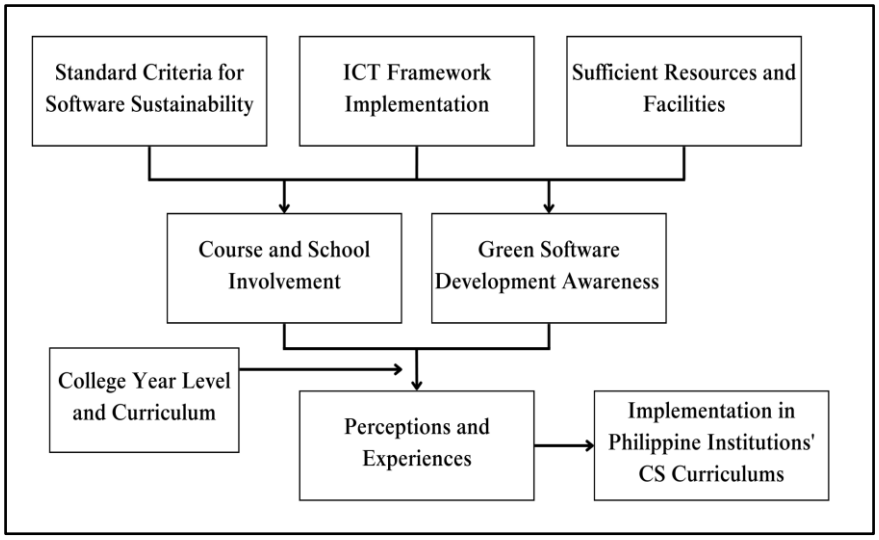

Figure 1 shows the conceptual framework map that is derived from the causal model and other factors from reviewed literature. Course and school involvement and green software development awareness are added factors in the map as they can influence consumer usage and have new priorities be adapted in different sectors (Verdecchia et al., 2021), such as the educational sector. The variables to be used in the data collection were derived from the conceptual framework map as it was used as a basis for formulating interview questions. From the three independent variables, standard criteria, ICT involvement, and sufficient resources and facilities, it can affect the perceptions and experiences of the undergraduate students, which result in the mediator variables: course and school involvement, and green software development awareness. These two mediator variables can affect the dependent variable, perceptions and experiences, as participation and awareness play a part in the perspectives and experiences of these undergraduate students. If other students are more aware than others, they may have more perceptions and ideas for the implementation of green software development in the CS Curriculums of Philippine Institutions.

Furthermore, the college year level and curriculum act as the moderator variable in the conceptual framework map, as it can affect the relationship of the independent and dependent variables. How much a student knows and is involved in green software development is affected by the time they have been studying in their course and under a specific type of curriculum. Finally, learning the perceptions and experiences of these students leads to getting a background and analysis on the implementations of green software development frameworks to different Philippine institutions' CS Curriculums.

Statement of the Problem

This research aimed to identify undergraduate students' perceptions of how Philippine Computer Science (CS) curriculums implement green software development in their courses. The main objective of this study was to determine whether undergraduate students were aware of green software development processes and were implementing them within academic premises, and if it was sufficient for them to be conscious of how beneficial and efficient it was to implement these practices. The study also followed three specific objectives to help achieve the main objective:

SQ1: To understand the factors affecting the integration of green software development frameworks in the curriculum of Computer Science majors in the Philippines.

SQ2: To determine the users' preferences for green applications through the lens of aspiring local software developers.

SQ3: To discern energy-effective processes and suggest applicable green software development frameworks in the CS Curriculum for reduced resource consumption.

Scope and Delimitation

The research informants were undergraduate students in CS majors in Philippine institutions. The study revolved around investigating the views of current students from different institutions to determine the status of green software, and thus applied a qualitative research approach. The study does not include veteran software developers in the field. Since the study revolved only on the perspectives of undergraduate students, other factors such as the curriculum of institutions themselves were not reviewed and studied. The time frame for conducting the study was limited to two months, from March 2023 to April 2023. Thematic analysis was used to analyze the data gathered for two weeks after data gathering. A Merleau-Ponty phenomenological approach was used to synthesize and describe the data.

Methodology

Research Design

The study followed a qualitative design by conducting a Phenomenology of Perception, adopted from Merleau-Ponty, to analyze the views of Computer Science majors in the Philippines. Implementing this type of phenomenology by Merleau-Ponty created a background for understanding certain insufficient studies, especially in the local aspect, as it aimed to shape the whole experience by compiling individual experiences through time (Laverty, 2003, as cited by Erciyes, 2015). This research aimed to have an in-depth understanding of green software development and consumers' demographics via SSI through the lens of an aspiring software developer.

Participants/Respondents of the Study

12 undergraduate students from different institutions all over the Philippines agreed to participate and served as informants in the study - 10 from Luzon, 1 from Visayas, and 1 from Mindanao. Furthermore, the year level of the informants was recorded in order to track when the institution began to introduce the idea of green software, if at all, in order to eliminate biases in the data. Two of the informants were enrolled in 1st year, one in 2nd year, five in 3rd year, and four in 4th the year of their program. Considering that most informants were studying at the higher levels, it was assumed they had taken enough courses to validate whether green software was included in their curriculum. Purposive and snowballing population sampling methods were used to identify the informants in the research. An inclusion criteria was set to limit the scope of informants. The following are the inclusion criteria that the researchers set:

- A Filipino citizen
- Is currently a student taking up a course related to computers and software development
- Is familiar with green software development

Instrument/s of the Study

The research conducted semi-structured interviews for data collection and thematic analysis for data analysis. One instrument used in the study was an interview questionnaire, where it served as an outline and guide to conduct the semi-structured interview online. This interview questionnaire included self-constructed questions by the researchers following the factors and flow of Ibrahim et al.'s (2021) table of green software development elements. Other factors from Kern et al.'s (2018) causal model and Verdecchia et al.'s (2021) environmental factors were also considered in constructing the interview questions.

The interview protocol is derived from Ibrahim et al.'s (2021) pilot study questionnaire. Some modifications were made to fit the demographic of the informants by the researchers. Since the nature of the interview is semi-structured, the researchers opted to use the constructed interview protocol as an outline to how they would ask the questions to the informants. Using a semi-structured interview enabled flexibility in gathering data, while still being grounded to certain points to avoid bias.

Data Collection and Analysis

The researchers conducted a semi-structured interview (SSI) with several informants involved in software development in their chosen course. Before

a schedule with the informants was set, an informed consent form was given to them via Google Forms. The form briefly summarizes the study and why they were chosen as potential informants. An interview guide was also sent to ensure complete transparency between the researchers and informants. The interview guide included factors referenced from Ibrahim et al.'s (2021) research and other additional elements. Before presenting the final interview protocol, the interview guide was validated by an expert in the field using an accept, remove, or revise format that the researchers formulated in a table. It was then pilot-tested through an initial interview. The interview time ranged from 15-30 minutes and was conducted via Zoom.

Recordings from the SSI were transcribed, coded, and analyzed. The data gathered from the series of interviews was assessed through thematic analysis. Thematic analysis was widely used in qualitative research as it provided a way to discover patterns and transcribe the data gathered from qualitative data collection methods (Kiger & Varpio, 2020). This approach allowed a flexible process of interpreting data from the informant's perspective without extensive knowledge or background in other qualitative analyses. It helped examine the informants' perspectives, contrast them, and generate new insights based on the data gathered.

Ethical Considerations

To ensure that the research would follow ethical guidelines, the researchers constructed a set of guidelines to follow when conducting the study. First, an informed consent form was created and sent to the informants before setting an interview schedule through Google Forms. All of the informants were ensured complete transparency with the interview questionnaire and research guidelines, and all of them had the freedom to voluntarily participate in the study or not. In data analysis, the informants' personal information was kept hidden and stored in a secure database, and their identities were maintained anonymous while writing the analysis.

Results and Discussion

Results

Table 1 shows the tabulated codes from the thematic analysis done on each interview transcript. The most mentioned codes from the informants were Capability, Sufficient Resources, and User-Friendliness with 9 informants who mentioned these in their interviews. Followed by Resourcefulness and Energy-saving with 8 informants. The least mentioned elements that were coded in the thematic analysis were Rules & Policy/Course and Culture, with only 2 informants who stated these elements.

Table 1: Table of Codes

Research Questions	Elements/Codes	I1 [NA]	I2 [JL]	I3 [AA]	I4 [JA]	I5 [GV]	I6 [GB]	I7 [MN]	I8 [LB]	I9 [JB]	I10 [JC]	I11 [TL]	I12 [SC]	Total
RQ1	Awareness	x		x			x			x		x		5
	Participation	x		x						x		x	x	5
	Rules & Policy/Course											x	x	2
	Culture											x	x	2
	Capability	x		x	x		x	x	x	x	x	x		9
	Automation	x		x			x	x		x	x	x		7
	Sufficient Resources	x		x	x	x	x			x	x	x	x	9
RQ2	User Friendliness	x		x		x	x	x	x	x	x		x	9
	Purpose/Function/Reliability	x		x	x		x	x			x		x	7
	Energy Efficiency	x		x								x		3
	Optimization					x		x			x	x		4
	Popularity/Brand Loyalty		x		x		x			x		x		5
	Aesthetics		x				x	x						3
RQ3	Resourcefulness		x	x	x	x	x		x			x	x	8
	Energy saving	x	x	x		x				x	x	x	x	8

Based on the data gathered from the twelve informants and from the coding process of the thematic analysis, three major themes were produced. These themes are directly related to each of the specific objectives stated in the research.

Green Software Development: Indirectly Taught

Most of the informants stated that academic institutions did not integrate green software development into their curriculums. Informants who were aware of the existence of green software development educated themselves voluntarily.

Informant 1: *"I am aware, but it's not because it was taught in our course yet, but I believe Green Software is sustainable with software in such a way that it lessens the consumption of electricity, and it lessens the processors needed."*

Informant 6: *"...For me, I think there's no policy that's hindering us or stopping us from developing these kinds of software. It's more of the students who choose not to actively just pursue it, you know, like, it's not their top priority."*

The informants discussed how they were only taught to focus on the main goal of their projects, which is to make them function and suit the needs of the client. Some of the informants try to apply green software development models when programming. However, some of them can only implement green solutions in a certain scope while developing software as they lack the appropriate resources to do so.

Informant 2: *"Last semester, our professor had to let us code on paper since not all of us have laptops to use."*

Informant 10: *"What we're using in our laboratory are kind of not up to date yet, both the hardware and software being used there. However, it can be used just for what they're teaching us."*

Lack of Awareness of Green Software Development

In their accounts, most informants said that their clients and users were not aware of green software development.

Informant 4: *"I'm not that aware when it comes to the energy efficiency of how I developed my software because it wasn't really that engraved in our lectures in our classes of how important it is when it comes to developing software,"*

Instead, they gave a brief description of what they think green software development is.

Informant 11: *"The really common theme here is reducing the carbon footprint and increasing the efficiency, which is really good advocacy to go for. Especially when it comes, when you consider how software is nowadays."*

Aside from this, some informants gave examples of factors they consider to encourage users to use a particular software. These include aesthetics (Informant 7), user-friendliness (Informant 5), and the capability and reliability of software (Informant 9).

Energy Efficiency: Not Top-Priority in Development

Most informants did not prioritize energy efficiency in software development.

Informant 6: *"To my knowledge, if we were given projects to do, the first thing that comes to our mind is just solving the problem that our client has, and then for efficiency, everything else, it comes secondary,"*

However, some informants prioritize energy efficiency in developing software to encourage specific demographics in using their end products.

Informant 3: *"Before we consider making projects. We consider if it can save resources or energy to make cost-effective solutions for the society."*

Informant 11: *"It's definitely going to be the efficiency when it comes to algorithm and software its really important… product or brand loyalty also comes from making sure what you are doing or contributing positively towards the environment."*

Based on the tabulation of similar codes and themes appearing from the interview transcripts, it shows that most of the informants did not actively apply green software development methodologies in their programming courses. There are also reasons as to why undergraduate students are not aware of these methodologies, as it was uncovered in the research. The main factor as to why is due to the lack of implementation of green software frameworks into local curriculums. Although there are some policies being enforced in some parts of the country (Sabili & Ignacio, 2020), it is not widely known or advertised for the public. Aside from this, more factors come into play on the lack of green software development awareness, and they are interconnected.

Discussion

All of the results from the data gathered are synthesized through the perspective of Merleau-Ponty's phenomenology, the idea of perception. From the informants and the thematic analysis conducted, the implementation of green software development frameworks in Computer Science curriculums in Philippine schools was deemed lacking in many areas. Currently, they do not account for its integration in developing software completely. This knowledge gap led local developers and users to a low prioritization for energy and resource-saving processes in software development. Instead, they prioritized factors such as functionality, user-friendliness, and optimization in software, which emphasizes clients' and users' needs, but disregards its environmental consequences to society.

Figure 2: Word Map of Research Codes

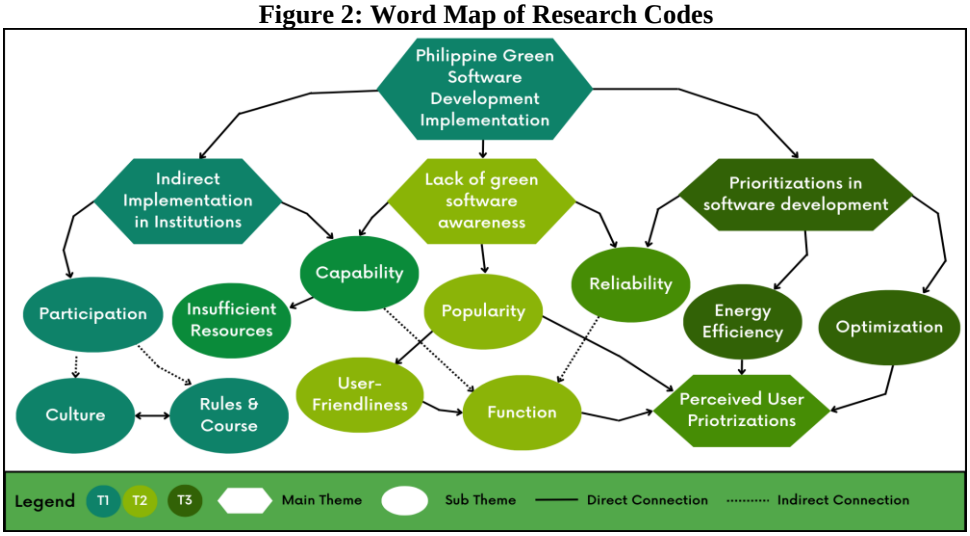

Figure 2 shows the relationships of the themes and codes that were thematized in the data gathering in a word map. They were organized by shape and color as well as different types of lines to distinguish the relationship of one theme to another. A concrete line indicates a direct relationship between two or more themes, and a broken line indicates an indirect relationship or influence on different themes. Each theme is color-coded according to its respective objective and is shaped according to its relevance. From the thematic analysis, it was determined that some of the sub-themes and codes align with 2 or more major themes, hence some having different hues.

The first theme shows that green software is indirectly implemented in the Philippine curriculum where resources, culture, and policies affect the participation of faculty and students. Culture and institutional policies affected the utilization of green software for their tasks wherein the most profitable or the most appealing elements were highlighted. The resources available also affected the application of green software as technical processes of software development weren't discussed influencing a disregard for energy-saving processes. One example is the lack of resources to support green software development frameworks. Though undergraduate students may be able to circumvent this issue by using their creativity, the problem still lies with cost reductions and limitations to resources. This is also seen in Ibrahim et al.'s (2021) study, wherein cost reduction and resources are important factors that affect green software development. Muthu et al. (2019) mentioned standard facilities and assets that are useful to green software developers, but the informants did not have access to most of these, such as small-scale servers, high expandability, and more. The disregard for energy-efficient processes has led to highlighting the target users without keeping in mind environmental consequences.

The second theme reveals that there is a distinct lack of green software development awareness among aspiring local developers. This is linked to the resources present and the emphasis on an appealing product or service for users as functionality and user-friendliness were considered for most informants. When developing software, the context behind it is important (Verdecchia et al., 2019), but since most of the local developers just focus on function rather than optimization due to the lack of resources, they do not realize that this consumes more energy. Through the lens of local developers, they also do not think that users are aware of green software, which is seen in Elgaanied-Gamber et al.'s (2021) and Sabili & Ignacio's (2020) studies. However, some local developers stressed the importance of energy-efficient processes as it promotes cost-effective software and plays a role in brand loyalty as it shaves off energy costs and is in line with the current sustainable goals the world is trying to reach.

The third theme highlights the prioritizations of the informants when developing software. Their utmost priority is to meet the needs of the client, to make their product function properly, and to make sure it is reliable for a broad population of users. Some of the informants try to lessen energy consumption by their own means. Even though there are no standard green frameworks that are utilized in the curriculum, some practices such as paperless documents, program optimization, and lessened use of facilities are done by some of the local developers. Kern et al. (2018) provided a set of criteria for green software development that provides organizational structures with usage patterns essential for green software development. However, none of these were mentioned by the informants. Just like the first

theme, there is an informal application of green solutions in software development in Philippine institutions, but these are not enough to reduce the resource consumption done by developers.

A connection between the implementation, awareness, and prioritization of green software development can be seen through the connections of the themes. Because clients and users are not aware of green software, local developers shift their attention to the factors that they look toward the most with user-centered processes. Additionally, the lack of resources makes it difficult for local developers to implement green solutions in developing their projects, even if they may be aware of these in the first place. Furthermore, the popularity of software may be tied to how user-friendly it is, depending on its functionality, which is a recurring theme in the study. Finally, energy efficiency is not prioritized enough by the developers due to both client needs and the lack of resources, so as a result, they prioritize the mentioned factors that encourage users and clients in using software the most.

Conclusion

The expansion of technology throughout time has brought digital interaction to the world. However, alongside this assistance, it had been deemed to negatively impact the environment by increasing users' energy consumption. To counter the increasing emission rates, the ICT implemented energy efficiency via green solutions to alter the programs into sustainability. Nevertheless, software developers need to be well-acquainted and carry out green software development practices to reduce global issues. Through the perspective of Merleau-Ponty phenomenology, the idea of perception, the interviews unraveled the factors that affected the integration of green software framework in the current academic curriculum, the features that affected users' utilization of applications, and energy-effective processes to implement green software development.

Awareness, participation, and capability were emphasized for green software integration factors. All institutions did not implement the integration of green solutions; most who were knowledgeable educated themselves voluntarily. On the other hand, some state universities had a massive gap in meeting their students' needs. Consequently, most users prioritized the features of user-friendliness (easy to use), functionality (practical interface), and popularity (known by the general public). These features could be taken into account if aspiring developers plan to participate in integrating green solutions into their future projects. Based on the informants' responses, the following themes were curated and finalized from the thematic analysis:

T1: Institutions did not teach green software development directly, as only functionality is given importance by their courses.
T2: Software users were not aware of green software, according to the informants, so software developers tend to forget to factor in energy efficiency in software development.
T3: Energy efficiency in software was not prioritized. End products and client demands were emphasized more, leading green software development processes to be disregarded.

Although green software was not thoroughly implemented, most universities still support sustainability by having an online portal for paperless transactions. Even though there were no concrete educational materials nor policies to be seen, it was evident that some unconscious practices might have employed green solutions in developing software as most informants believed they were trying to be energy efficient. Green solutions in the Philippines curriculum were not executed. Institutions should be wary of its existence to combat the alarming consequences of carbon footprint in the country.

Recommendations

For future studies, another group of target local informants, such as professors or experienced software developers, could be sought for an extensive collection of perspectives in the Philippines. The interview structure could be altered in a way to dive deeper into their experiences as seasoned professionals. Quantitative approaches are also recommended to further increase the specifics of the factors in implementing green software development frameworks in a local setup.

The findings of the research can be used to more effectively implement the mentioned green software practices by different research. Although some of the data can be polished by more intensive data collection and analysis. Since this research is focused on setting a baseline research for the awareness of green software development in the Philippines, a qualitative setup is done. Future research may focus on quantitative data analysis to pinpoint key areas that need improvement in this field.

Conflicts of Interest

The authors declare that there is no significant competing financial, professional, or personal interests that might have influenced the performance or presentation of the work described in this manuscript.

References

1. Cunliff, C. (2020). Beyond the Energy Techlash: The Real Climate Impacts of Information Technology. ITIF. https://itif.org/publications/2020/07/06/beyond-energy-techlash-real-climate-impacts-information-technology
2. Eastwood, B. (2020). The 10 Most Popular Programming Languages to Learn in 2022. Northeastern. https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/
3. Erciyes, E. (2015). Three Pillars of Phenomenology: Husserl, Heidegger, and Merlau-Ponty. https://www.semanticscholar.org/paper/Three-Pillars-of-Phenomenology%3A-Husserl%2C-Heidegger-Erciyes/a07811f9c2c2589c4eb9970e0d4fc09dc15833d4
4. Elgaanied-Gambier, L., Bertrandias, L. & Bernard, Y. (2021). Cutting the Internet's Environmental Footprint: An Analysis of Consumers' Self-Attribution of Responsibility. Journal of Interactive Marketing, 50(200), 120-135. https://doi.org/10.1016/j.intmar.2020.02.001.
5. Garnier, M.P. (2021). Green Software - A New Trend for a Better Planet. altenergymag. https://www.altenergymag.com/article/2021/09/green-software-%E2%80%93-a-new-trend-for-a-better-planet/35896
6. Ibrahim, S., Yahaya, J., Salehudin, H. & Deraman, A. (2021). The Development of Green Software Process Model. International Journal of Advanced Computer Science and Applications, 12(8). https://doi.org/10.14569/ijacsa.2021.0120869.
7. IEA. (n.d.). Data centres & networks. https://www.iea.org/fuels-and-technologies/data-centres-networks
8. Sabili, M.A. & Ignacio, A. (2020). Green computing awareness: a case of a Philippine public higher education institution. International Journal of Computing Sciences Research, 5(1), 534-540. doi: 10.25147/ijcsr.2017.001.1.57.
9. Kern, E., Hilty, L., Guldner, A., Maksimov, Y., Filler, A., Grőger, J. & Naumann, S. (2018). Sustainable software products - Towards assessment criteria for resource and energy efficiency. Future Generation Computer Systems, 86, 199-210. https://doi.org/10.1016/j.future.2018.02.044.
10. Kiger, M. and Varpio, L. (2020). Thematic analysis of qualitative data: AMEE Guide No. 131. Medical Teacher, 42(8), 846-854. https://doi.org/10.1080/0142159X.2020.1755030.
11. Muthu, M., Banuroopa, K. & Arunadevi, S. (2019). Green and Sustainability in Software Development Lifecycle Process. Sustainability Assessment at the 21st century. https://doi.org/10.5772/intechopen.88030
12. Patel, V. (2022). The Philippines Combats High Energy Usage by Using Smart Technology. Gizchina. https://www.gizchina.com/2022/04/04/the-philippines-combats-high-energy-usage-by-using-smart-technology/
13. Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. & Saraiva, J. (2021). Ranking Programming Languages by Energy Efficiency. Science of Computer Programming, 205. https://doi.org/10.1016/j.scico.2021.102609.
14. Radzick, Z. (2022). The what, why and how of green software development. Boldare. https://www.boldare.com/blog/what-is-green-development/
15. Tabuga, A. & Cabaero, C. (2021). Filipinos' Access and Exposure to ICT: A General Overview based on the National ICT Household Survey. Philippine Institute for Development Studies. https://pidswebs.pids.gov.ph/CDN/PUBLICATIONS/pidsdps2113.pdf
16. Verdecchia, R., Lago, P., Ebert, C. & de Vries, C. (2021). Green IT and Green Software. IEEE Software, 38(6), 7-15. https://doi.org/10.1109/MS.2021.3102254.