

Software Sustainability from a User Perspective

A case study of a developing country (Kingdom of Saudi Arabia)

Sulaiman Aljarallah

Department of computer science, Loughborough University
Loughborough, UK

Russell Lock

Department of computer science, Loughborough University
Loughborough, UK

Abstract—Interest in sustainable development is increasing. Understanding the user’s perspective toward software sustainability helps to enhance understanding of the concept. The need for developing countries to enhance their ICT infrastructure to align with UN sustainable development goals increases the necessity to understand the current perception of software users, industry and sustainability experts, to improve the level of software sustainability. Software sustainability has a number of challenges with regard to adoption by software users. This study investigates software sustainability from users’ points of view in the Kingdom of Saudi Arabia (KSA). It also addresses key barriers for sustainable software.

Keywords—Software; Sustainability; Sustainable software; Green software; Developing countries.

I. INTRODUCTION

Sustainability is “the ability to be maintained at a certain rate or level” [1]. Many definitions have been proposed for it [2]. [3] states that there are nearly 2000 definitions for sustainability, which indicates the inconsistent and vague nature of this concept. However, the most cited definition is the UN Brundtland report definition [4] as “development that meets the needs of the present generation without compromising the ability of future generations to meet their own needs” [5,p.15]. Sustainability is shaped by three dimensions, namely economic, environmental and social [6].

Similarly, there is no agreed definition for software sustainability [7]. Some authors adopted Dicks *et al.*’s definition [8] such as Penzenstadler *et al.*[6] and Conejero *et al.* [9], who focused their definition of software sustainability on the impact (positive or negative) of software on various dimensions (economy, society, human beings, and environment) when using, deploying or developing software. Correspondingly, authors have proposed five dimensions [10] (environmental, social, economic, technical, individual), [7], [9] for software sustainability while others proposed four dimensions [11], [12] by ignoring the individual. It can be argued that the inconsistency and vagueness seen in software sustainability literature is due to the immaturity of the software engineering (SE) field and software sustainability concept [13].

Software sustainability is a complex concept and non-functional requirements should be considered a component part, as should quality attributes that eliminate or reduce negative impacts on the environmental, social, economic,

political and technical dimensions, while increasing positive practices and impacts over the short and long term.

This study aims to explore the user perspective with regard to sustainable software and explore their *beliefs, intentions, attitudes* and *perception* toward using sustainable software. In addition, the paper investigates the barriers users face in adopting sustainable software.

II. LITERATURE REVIEW

A. Previous studies

Previous studies have identified confusion surrounding software sustainability [7], highlighted its recent development [14], still at the exploration stage [15] and lacks sufficient empirical study[16]. The lack of previous studies is particularly acute in relation to developing countries [17]. No, study has been reported before providing an empirical exploratory quantitative study into understanding the user perspective on software sustainability in the Kingdom of Saudi Arabia (KSA), or within the field of SE.

There are qualitative investigations such as [10], [18], [19], [20], [21], [22], [23] and [24] and other quantitative studies such as [11], [25], [26], [27], [28] and [29], and mixed methods studies [30] and [31], all focusing on software sustainability from different perspectives e.g. Green. Most collect their data from software engineers, project managers, SE students or practitioners in the software industry.

Moreover, there are empirical studies surveying the current literature on software sustainability, such as [17], [32], [15], [16] and [33] or analyzing case studies such as [34]. Most of these studies focus on the green dimension. Other studies utilized the data in software repositories such as GitHub or SourceForge [35], [36], [37], [38] and [39] to be data sources for their research on sustainability within SE; however, some of them focus on sustainability of open source software.

B. Research Design

There are many models that help to measure the acceptance of new technology. However, users currently do not consider systems in terms of sustainability making evaluation using existing models problematic. As a result, a set of hypotheses have been derived from literature and divided into a sub-scale in order to investigate the end user perspective on using sustainable software. Some models, such as TRA and TAM,

influence our proposed model. We argue that extracting user intention on sustainable software would be helpful for the SE industry.

Four sub-scales are proposed which covers various hypothesis (H), namely *Beliefs* (6H), *Intention* (2H), *Attitude* (6H) and *Perceptions* (4H). Barriers are constructed to these hypothesis based on literature forming (10H) for adopting and using sustainable software.

III. METHODOLOGY

This will be the first large scale case study conducted in a developing country, bringing into focus the user perspective. It is an exploratory empirical investigation [40], which adopted a quantitative methodology by utilizing an extensive e-survey in KSA. Non-random sampling was utilized as this is advantageous in exploratory research [40]. Multiple sampling strategies have been used in order to overcome the disadvantages of using each type alone and to gather a large sample for the citizen questionnaire. Closed questions are adopted in the majority of 88 questions in order to quantify the result. A five-option Likert scale is utilized.

In order to ensure reliability and internal consistency of the questionnaire, two methods were used, namely face validity by experts, and Cronbach's Alpha test.

The Cronbach's Alpha value for *Beliefs*, *Intentions*, *Attitudes*, *Perceptions* and *Barriers* are 0.669, 0.500, 0.807, 0.603 and 0.819, respectively. For Cronbach's Alpha values are considered high for 0.7-0.9 where 0.5-0.7 is acceptable [41]. [42] indicates that a low number of items in the scale could strongly underestimate Cronbach's Alpha. [43] states that a scale lower than ten items will reflect low Cronbach's Alpha. In our case, all the scales are quite low on items, with the intention scale the lowest.

Sample size estimation was conducted to evaluate the influence of the intervention. Considering the population size for KSA is thirty million, confidence level is 95% and margin of error is 3.26%, the sample size estimation is 904 respondents. 906 responses were collected, which satisfies the sample size estimation. However, due to the complexity of the notion and the existence of ranking questions which require more effort, there were some incomplete responses.

IV. RESEARCH FINDINGS

A. Factors Influencing Adoption of Software Sustainability

Descriptive analysis, frequencies and mean calculation were utilized in SPSS for the survey analysis. The *beliefs* scale, as seen in Figure 1, contains six questions investigating user

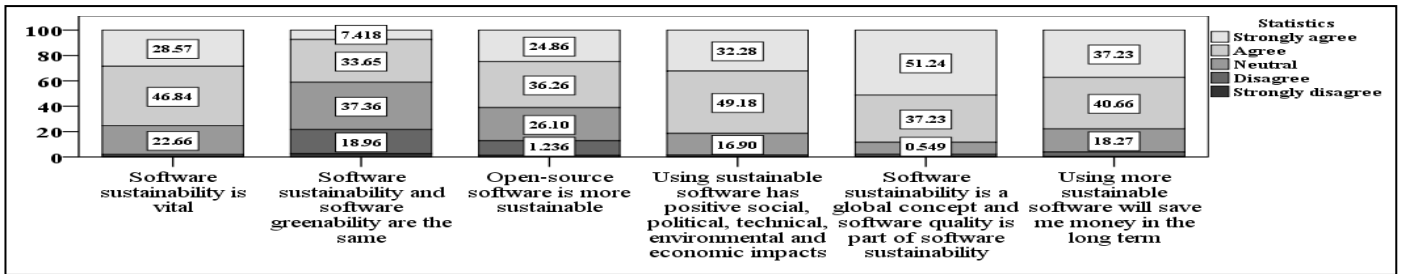


Figure 1: Beliefs scale responses

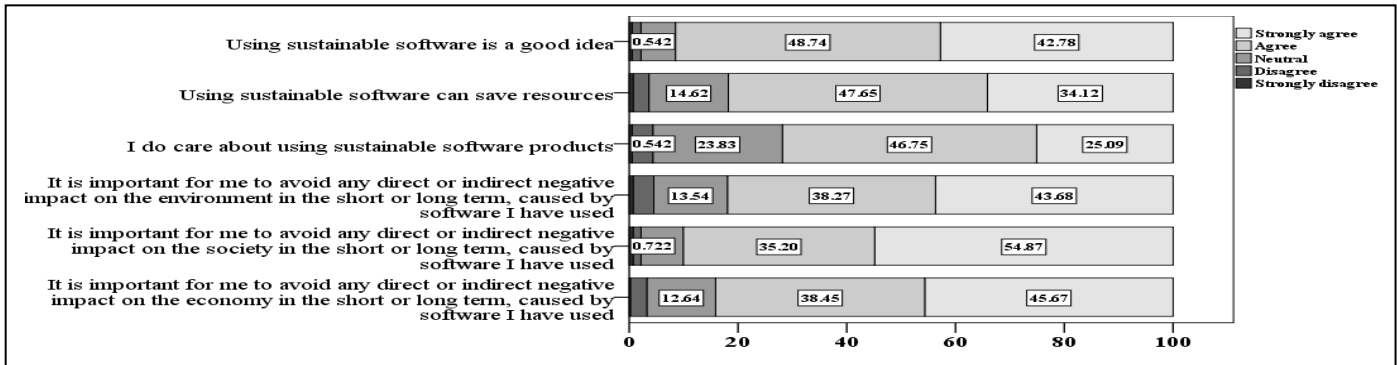


Figure 2: Attitude scale responses

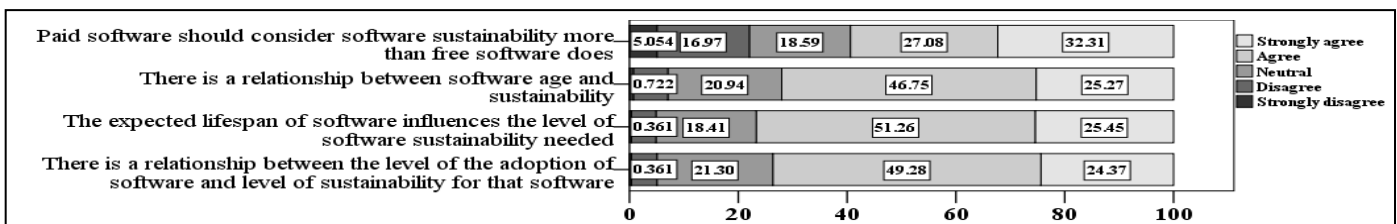


Figure 3: Perceptions scale responses

belief regarding certain statements. The scale examines the importance of software sustainability, which shows that it is vital; however, the middling percentage reflects the uncertainties about respondents understanding of software sustainability. Sustainability can be hard to relate to unless further information is provided [29]. Another study reports software sustainability is not a major concern for users [25]. However, the result reflects reasonable importance among users on software sustainability. Moreover, respondents are asked if they can differentiate between the notions of greenability and sustainability in terms of software. [44] reports that terms like ‘energy-efficient’, ‘carbon-neutral’ and ‘sustainable’ in most cases are used interchangeably. The result shows the differences between neutral and total agreement is marginal, which indicates the ambiguity of the concept; however, 41% showed misunderstanding of the concept of sustainability, a high proportion. It is worth noting that awareness is essential, especially in education; however, the level of education for the sample is considered high as 300 had higher degrees, 425 had bachelor degrees, 86 held diplomas and 95 had high school or below. The third question examines the belief of software users regarding whether open source software (OSS) is more sustainable than other types of software. The result shows an acceptable level of uncertainty (neutral) by a quarter of the sample, while the total agreement is 60%. The fourth question investigated the belief of users regarding using sustainable software, leading to a positive impact among software sustainability dimensions, namely social, economic, environmental, technical and political. This result reflects a significant agreement of the positive impact over these dimensions which prove that users have a strong belief in the benefit and positive influence of sustainable software. The relationship between software quality and sustainability is disputed. [11] describes characterizing sustainability as software quality as a challenge. Some authors classified it as NFR [25], whereas others consider it an encompassing concept [45]. 88% of users consider sustainability a global concept with software quality a part of it. The result paves the way to understand the relationship; however, software experts need to be consulted to develop this. The last question examines the belief of users in return on investment when using sustainable software, in terms of the economic dimension. The result shows a significant belief in the benefit of using sustainable software.

The *intention* scale argues two hypothesis related to the technical (trade-off) and economic dimensions (cost) as seen in Figure 4. [46] states that software sustainability adds complexity which increases cost. Users are asked about their willingness to pay more for sustainable software, since the SE industry will pay more attention to sustainability requirements during the software development life cycle (SDLC). Results show that nearly half of respondents are willing to pay more for sustainable software. However, 30% have no opinion, while 20% disagree, which reflects significant uncertainties; however, the cost of sustainable software needs further research. Nearly 70% agree on trading off some features in favour of more sustainable software; however, trading off

software features may be affected by various factors such as culture, education and sustainability awareness. Similarly, a study for green software practitioners done by [31] shows that 80% of practitioners are willing to trade off some features to reduce energy usage. Our result contradicts claims by [47] that software development plays an important role in rebound effects (third order effects), by encouraging engineers to improve products with any savings made.

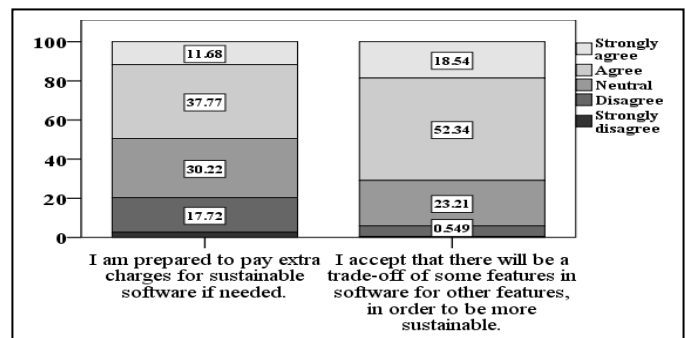


Figure 4: Intention scale responses

The attitudes of users toward using sustainable software consisted of six questions, as seen in Figure 2. In the first question, 90% of users find using sustainable software is a good idea, while [25] indicates in his study that software sustainability is not a primary concern for users. Comparing this result with the first question in the *beliefs* scale, which shows 75% agreement towards the importance of software sustainability, it can be argued that users find the software sustainability notion and using sustainable software is vital. 81% of users find sustainable software saves resources. These results show that users are aware of positive direct impact [45]. [8] states that not only does sustainable software focus on the environmental dimension, i.e. reducing resources consumption, but should also consider economic and social factors. Nearly three-quarters of users show that they care about using sustainable software which shows that they are aware of sustainability importance. The last three questions examined the same concept but in three dimensions, namely ecological, social and economic, in terms of whether users are interested in avoiding negative impacts in the short or long term caused by the software that they use. Software impact on economic, social, environmental and human should be reduced during the software life cycle (SLC), which includes usage [48]. The questions show that users are interested in avoiding negative impacts in all dimensions; however, users are interested in social, economic and environmental dimensions in terms of avoiding negative impacts by the software they use, with calculations proving it as 4.42, 4.26 and 4.20, respectively. A study interviewing 10 team lead developers in Austria shows that the most important sustainability aspects are technical and organisational, followed by economic, whereas the environmental aspect is ignored [18]. The last placement of the environmental dimension requires further investigation in different countries, developing and developed.

The *Perceptions* scale covers three areas, namely software cost and level of sustainability, longevity in terms of lifespan and the adoption and sustainability relationship as seen in Figure 3. Producing sustainable products increases near term costs covered by producer, distributor etc. [49], and is reflected later in the product price. In terms of software development, more software development time and complexity during development will cause production costs to increase. [46] indicates the cost of sustainable development production is a barrier for the SE industry due to complexity in meeting requirements, which will make the software price higher, otherwise the organisation is economically and socially unsustainable in terms of ROI time and effort for software engineers. Increased costs could come from hiring sustainability experts, conforming with sustainability standards, and certification costs. In the *intention* scale, users are willing to pay more for sustainable products; however, they are asked here whether they think paid software is more sustainable than free software. Results shows that 60% of users agree that paid software is more sustainable while 22% disagree and 18% have no opinion. This shows that users expect paid software to meet sustainability requirements more than free software. The second and third questions show that the majority of users agree on the relationship between lifespan and sustainability level. This result is interesting since it shows that users do not expect high levels of sustainability for short-lifespan software. [50] makes a connection between software lifetime and functionality by expressing that decreasing unnecessary functions should be based on software lifespan. In the same context, lifespan can affect strongly the level of sustainability. [32] report in their study that “*It was striking to note that no statistically significant collocates surfaced in the concordance analysis that pointed to the time dimension of sustainability*”. Nevertheless, the adoption level of software and sustainability are related by agreement of nearly three-quarters of the sample.

B. Barriers

Ten sustainability barriers for users are proposed based on literature.

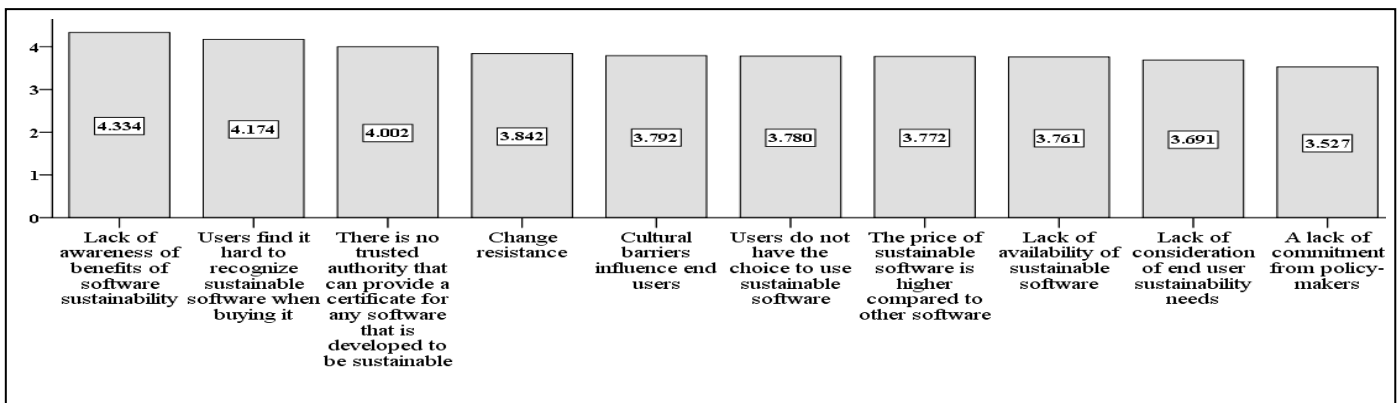


Figure 5: Software sustainability barriers in view of software users

As seen in Figure 5, the mean is calculated for the sake of brevity. The most significant barrier for adopting sustainable software is the lack of awareness of the benefits of software sustainability. A recent study points out the importance of investigating how to increase awareness of software sustainability and how it can be handled in practise among the SE community [18]. [25] reports that the awareness of sustainability is escalating; however, it is still a major barrier and in order to bring sustainability to people’s minds, frameworks and guidelines are needed [6]. Recognizing sustainable software when buying it is the second significant barrier. According to [48], users require some information about sustainability when purchasing software. Next, users find the lack of a trusted authority to provide a sustainability certificate for any software developed to be a barrier. This is related to the previous barrier concerning lack of information about sustainable software products. [6] claims the importance of ‘*Green Software Label*’ for sustainable software can help users find information about resource consumption during development and energy consumption during usage. Similarly, [51] claims that sustainable software labels should be granted by a trusted authority. However, to adopt such a solution, an authority should provide a standard in order to measure compliance for sustainability during SDLC. [52] states users should be supported by a sustainability knowledge base regarding optimizing software impact during the usage phase. Change resistance toward adopting sustainable software involves relinquishing some features as trading off for the sake of sustainability.

The culture barrier is considered an obstacle that holds users back from adopting sustainable software. [10] reports the difficulties in convincing the SE community of the need for change to support sustainability, whereas [46] states intellectual barriers must be overcome by the SE community. In the same context, change resistance and cultural barriers can influence users in adopting sustainable software, which in turn affects the SE industry. [53] argues that culture is a major barrier in convincing people to consider future generations. Next, users find they do not have the choice to use sustainable

software, since the SE industry operates on supply and demand. Next, it is not surprising to see cost listed by fewer respondents since there are no sustainable software products in terms of marketing. Information about sustainable software is lacking, since the SE community is still struggling to cope with the concept of sustainability. [20] warns of the collapse of SE companies if cost is not balanced well with developing sustainable products. Such behaviour will lead to unsustainable development processes. Sustainability increases complexity, which in turn increases software costs for the reasons previously discussed. Lack of availability of sustainable software, lack of consideration of end user sustainability needs and a lack of commitment from policy-makers are considered the lowest ranked barriers according to their mean.

V. CONCLUSION

This study endeavors to explore software sustainability from the point of view of users in KSA. In doing so, it highlights key factors that may influence user adoption of sustainable software based on four scales that show some interesting focal points. The study also investigates the key barriers which stem from the literature with regard to user perspectives in KSA, proving their importance. The future work is to investigate expert's opinion in software engineering community and develop a software sustainability framework.

REFERENCES

- [1] Oxford Dictionary, "Oxford Dictionary," 2018. [Online]. Available: <https://en.oxforddictionaries.com/definition/sustainability>. [Accessed: 05-Jun-2018].
- [2] M. L. Weitzman, "Sustainability and Technical Progress," *Scand. J. Econ.*, vol. 99, no. 1, pp. 1–13, Mar. 1997.
- [3] M. A. White, "Sustainability: I know it when I see it," *Ecol. Econ.*, vol. 86, no. 2, pp. 213–217, Feb. 2013.
- [4] M. JASIULEWICZ-KACZMAREK, "Is Sustainable Development an Issue for Quality Management?," *Found. Manag.*, vol. 6, no. 2, Jan. 2014.
- [5] J. K. Akotia, "A Framework for Social and Economic Sustainability Benefits Evaluation of Sustainable Regeneration Projects in the UK," University of Salford, 2014.
- [6] B. Penzenstadler, V. Bauer, and A. Fleischmann, "Seminar: Sustainability in Software Engineering," Technische Universit" at M"unchen, Germany, munchen, pp. 1–5, 2011.
- [7] C. Venters, C. Jay, L. M. S. Lau, M. K. Griffiths, V. Holmes, R. R. Ward, J. Austin, C. E. Dibsdale, and J. Xu, "Software sustainability: The modern tower of babel," in *Third International Workshop on Requirements Engineering for Sustainable Systems*, 2014, pp. 7–12.
- [8] M. Dick, S. Naumann, and N. Kuhn, "A Model and Selected Instances of Green and Sustainable Software," in *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience: 9th IFIP TC 9 International Conference, HCC9 2010 and 1st IFIP TC 11 International Conference, CIP 2010, Held as Part of WCC 2010, Brisbane, Australia, Sept, J. Berleur, M. D. Hercheui, and L. M. Hilty, Eds. Berlin: Springer, 2010, pp. 248–259.*
- [9] J. M. Conejero, I. S. Brito, J. C. Ana Moreira, and J. Ara'ujo, "Modeling the Impact of UAVs in Sustainability," in *IEEE 24th International Requirements Engineering Conference Workshops*, 2016, vol. 5, pp. 208–216.
- [10] R. Chitchyan, C. Becker, S. Betz, L. Duboc, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability design in requirements engineering: state of practice," in *38th International Conference on Software Engineering Companion (ICSE '16)*, 2016, pp. 533–542.
- [11] N. Condori-Fernandez and P. Lago, "Characterizing the contribution of quality requirements to software sustainability," *J. Syst. Softw.*, vol. 137, pp. 289–305, Mar. 2018.
- [12] P. Lago, S. A. Koçak, I. Crnkovic, and B. Penzenstadler, "Framing sustainability as a property of software quality," *Commun. ACM*, vol. 58, no. 10, pp. 70–78, Sep. 2015.
- [13] M. Mahaux and C. Canon, "Integrating the Complexity of Sustainability in Requirements Engineering," in *1st international workshop on Requirements for Sustainable Systems*, 2013, pp. 1–5.
- [14] O. Philippe, N. C. Hong, and S. Hettrick, "Preliminary analysis of a survey of UK Research Software Engineers," in *WSSSPE 2016 Workshop on Sustainable Software for Science: Practice and Experiences Proceedings of the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4) University of Manchester, Manchester, UK, September 12--14, 2016*, 2016.
- [15] G. A. G. Mireles, M. A. Moraga, F. Garcia, and M. Piattini, "A classification approach of sustainability aware requirements methods," in *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, 2017, pp. 1–6.
- [16] G. A. García-Mireles and H. A. Villa-Martínez, "Practices for Addressing Environmental Sustainability through Requirements Processes," in *Advances in Intelligent Systems and Computing*, vol. 688, 2018, pp. 61–70.
- [17] M. Mohankumar and M. A. KUMAR, "Empirical Study on Green and Sustainable Software Engineering," in *14th International Conference on Software Engineering, Parallel and Distributed Systems · SEPADS 2015*, 2015, pp. 95–105.
- [18] I. Groher and R. Weinreich, "An Interview Study on Sustainability Concerns in Software Development Projects," in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2017, pp. 350–358.
- [19] M. R. De Souza, R. Haines, and C. Jay, "Defining Sustainability through Developers' Eyes: Recommendations from an Interview Study," Manchester, 2014.
- [20] P. Stoll, "Exploring Sustainable Industrial Software System Development within the Software Architecture Environment," Malardalen University, 2009.
- [21] T. A. Viaro and G. L. R. Vaccaro, "Exploring Green IT and Green IS: Insights from a case study in Brazil," in *2nd International Conference on Sustainable Intelligent Manufacturing (SIM)*, 2013, no. October, pp. 625–630.
- [22] O. A. Shola, "Early Investigation Towards Defining and Measuring Sustainability as a Quality Attribute in Software Systems," Lappeenranta University of Technology, 2016.
- [23] H. Hygerth, "Sustainable Software Engineering: An Investigation Into the Technical Sustainability Dimension," KTH Royal Institute of Technology, 2016.
- [24] M. L. Gibson, C. C. Venters, L. Duboc, S. Betz, R. Chitchyan, M. Palacin-Silva, B. Penzenstadler, and N. Seyff, "Mind the Chasm: A FishEye Lens View of Sustainable Software Engineering in UK Higher Education," in *6th International Workshop on Requirements Engineering for Sustainable Systems*, 2017, pp. 15–24.
- [25] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson, "Toward sustainable software engineering," in *Proceeding of the 33rd international conference on Software engineering - ICSE '11*, 2011, pp. 976–979.
- [26] J. Kasurinen, M. Palacin-Silva, and E. Vanhala, "What Concerns Game Developers? A Study on Game Development Processes, Sustainability and Metrics," in *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)*, 2017, pp. 15–21.

- [27] M. Salam and S. U. Khan, "Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey," *J. Softw. Evol. Process*, no. December 2017, p. e1939, Feb. 2018.
- [28] S. A. Koçak, G. I. Alptekin, and A. B. Bener, "Integrating Environmental Sustainability in Software Product Quality," in *Proceedings of the Fourth International Workshop on Requirements Engineering for Sustainable Systems co-located with the 23rd IEEE International Requirements Engineering Conference (RE 2015)*, 2015, pp. 17–24.
- [29] D. Torre, G. Procaccianti, D. Fucci, S. Lutovac, and G. Scanniello, "On the Presence of Green and Sustainable Software Engineering in Higher Education Curricula," in *1st International Workshop on Software Engineering Curricula for Millennials*, 2017, pp. 54–60.
- [30] M. Z. Huber, L. M. Hilty, and M. Glinz, "Uncovering sustainability requirements: An exploratory case study in canteens," in *Fourth International Workshop on Requirements Engineering for Sustainable Systems*, 2015, pp. 35–44.
- [31] I. Manotas, C. Bird, R. Zhang, D. Shepherd, C. Jaspan, C. Sadowski, L. Pollock, and J. Clause, "An empirical study of practitioners' perspectives on green software engineering," in *Proceedings of the 38th International Conference on Software Engineering - ICSE '16*, 2016, pp. 237–248.
- [32] C. C. Venters, N. Seyff, C. Becker, S. Betz, R. Chitchyan, L. Duboc, D. McIntyre, and B. Penzenstadler, "Characterising Sustainability Requirements: A New Species Red Herring or Just an Odd Fish?," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, 2017, pp. 3–12.
- [33] G. A. García-Mireles, M. Á. Moraga, F. García, C. Calero, and M. Piattini, "Interactions between environmental sustainability goals and software product quality: A mapping study," *Inf. Softw. Technol.*, vol. 95, no. April 2017, pp. 108–129, Mar. 2018.
- [34] R. Chitchyan, I. Groher, and J. Noppen, "Uncovering sustainability concerns in software product lines," *J. Softw. Evol. Process*, vol. 29, no. 2, p. e1853, Feb. 2017.
- [35] A. Aldabjan, R. Haines, and C. Jay, "How should we measure the relationship between code quality and software sustainability?," in *Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4)*, 2016, pp. 1–2.
- [36] S. Alhozaimy, R. Haines, and C. Jay, "Track 1 Lightning Talk : Forking as a Tool for Software Sustainability — An Empirical Study," in *Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE5.1)*, 2017, pp. 10–12.
- [37] F. Ahmed, H. Mahmood, and A. Aslam, "Green computing and Software Defects in open source software: An Empirical study," in *2014 International Conference on Open Source Systems & Technologies*, 2014, pp. 65–69.
- [38] C. M. Schweik and E. Raymond, "Sustainability in Open Source Software Commons: Lessons Learned from an Empirical Study of SourceForge Projects," *Technol. Innov. Manag. Rev.*, vol. 3, no. 1, pp. 13–19, 2013.
- [39] J. Long and M. J. Yuan, "Are All Open Source Projects Created Equal? Understanding the Sustainability of Open Source Software Development Model," in *AMCIS 2005*, 2005, p. 435.
- [40] J. Cummings and C. Sibona, "Crowdsourcing Surveys: Alternative Approaches to Survey Collection," *J. Inf. Syst. Appl. Res.*, vol. 10, no. 1, pp. 44–54, 2017.
- [41] P. R. Hinton, C. Brownlow, I. McMurray, and B. Cozens, *SPSS Explained*. London: Routledge, Taylor & Francis, 2004.
- [42] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," *Int. J. Med. Educ.*, vol. 2, pp. 53–55, Jun. 2011.
- [43] J. Pallant, *SPSS survival manual: a step by step guide to data analysis using IBM SPSS*, 4th ed. Open University Press, 2011.
- [44] S. Naumann, E. Kern, M. Dick, and T. Johann, "Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects," in *ICT Innovations for Sustainability*, vol. 310, L. M. Hilty and B. Aebischer, Eds. Cham: Springer International Publishing, 2015, pp. 191–205.
- [45] B. Penzenstadler, "Infusing Green: Requirements Engineering for Green In and Through Software Systems," in *Third International Workshop on Requirements Engineering for Sustainable Systems*, 2014, pp. 44–53.
- [46] K. Roher and D. Richardson, "Sustainability requirement patterns," in *2013 3rd International Workshop on Requirements Patterns (RePa)*, 2013, pp. 8–11.
- [47] W. Hilty, Lorenz M.; Lohmann, "The Five Most Neglected Issues in 'Green IT,'" *Cepis Upgrad.*, vol. 12, no. 4, pp. 11–15, 2011.
- [48] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Sustain. Comput. Informatics Syst.*, vol. 1, no. 4, pp. 294–304, Dec. 2011.
- [49] E. Pavlovskaja, "Sustainability criteria: their indicators, control, and monitoring (with examples from the biofuel sector)," *Environ. Sci. Eur.*, vol. 26, no. 17, pp. 1–12, Dec. 2014.
- [50] T. Tamai and Y. Torimitsu, "Software lifetime and its evolution process over generations," in *Proceedings Conference on Software Maintenance 1992*, 1992, pp. 63–69.
- [51] E. Kern, M. Dick, S. Naumann, and A. Filler, "Labelling Sustainable Software Products and Websites: Ideas , Approaches , and Challenges," in *29th International Conference on Informatics for Environmental Protection*, 2015, pp. 82–91.
- [52] T. Johann, M. Dick, E. Kern, and S. Naumann, "Sustainable development, sustainable software, and sustainable software engineering: An integrated approach," in *2011 International Symposium on Humanities, Science and Engineering Research*, 2011, pp. 34–39.
- [53] J. Ferri, R. M. de Barros, and J. D. Brancher, "Proposal for a Framework Focus on Sustainability," in *2011 30th International Conference of the Chilean Computer Science Society*, 2011, pp. 127–134.