Classifying the Measures of Software Sustainability

Shola Oyedeji
LUT School of Engineering (LENS)
Lappeenranta University of Technology
Lappeenranta, Finland
shola.oyedeji@lut.fi

Ahmed Seffah
LUT School of Engineering (LENS)
Lappeenranta University of Technology
Lappeenranta, Finland
ahmed.seffah@lut.fi

Birgit Penzenstadler
Department of Computer Engineering and Computer Science California State University Long Beach *(*CSULB*)* Long Beach, California, USA
birgit.penzenstadler@csulb.edu

***Abstract*— Energy efficiency is one of the very few measures widely used for evaluating green and sustainable software systems. This paper investigates the current measures of software sustainability from the four different software sustainability perceptions: Sustainability in Software Development, Green Software Systems, Software for Sustainability, Sustainability of the Software Eco System and Software Sustainability Dimensions (Economic, Social, Individual, Technical and Environment). While exploring the literature on green and sustainable software systems, measures of green software and software sustainability were identified, compiled and classified according to the four sustainability perceptions.**

Keywords*— green software, sustainable software, measures, sustainability, sustainability perceptions, green measures, software measurement.

I. Introduction

Sustainability is now one of the world major challenge [1][2]. The United Nations Sustainable development Goals (SDGs) shows the importance of sustainability in all facet of human lives and development. Today's economy rely on information and communications technology (ICT) in which software is a key factor and catalyst for all economic activities and a major driver linking all sectors. As stated in an Ericsson report that ICT can help reduce the global greenhouse gas (GHG) emissions by 15% [3]. Currently ICT itself contributes an estimated 2% to the global CO_2 emissions and accountable for approximately 8% of the European Union (EU) electricity consumption [4]. This shows ICT can a huge potential to help support sustainability and Green [5] but at same time it is important to explore avenues to make ICT domain more green and sustainable because of its huge impact on sustainability. Finding ways to properly evaluate software in regards to green and sustainability will provide avenues to reduce the current negative impacts of ICT.

This research explores the ongoing perceptions in the software engineering domain with the goal to identify the current and future measures used in the evaluation of green and sustainable software. Via triangulation of data from diverse sources, the measures are clustered into the four perceptions of sustainability in software engineering and sustainability dimensions. The long term goal of this research is to answer the following challenging questions: what are the current measures used in evaluating green and sustainability aspects of software systems and how can these measures be grounded in the software sustainability measurement theory.

II. Background

A. *Sustainability in Software Development*

As a measurable attribute, software sustainability is more than the perceptions of capacity to endure [6]. Sustainable software measures should include the direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software [7]. It is also beyond the current focus of sustainability in requirements engineering where sustainability is considered as a nonfunctional requirement (NFR) by some [8][9][10]. In [2], the authors reported on a software project in which sustainability requirements were treated as quality requirements, and systematically elicited and documented. Another work also proposed an approach to tackle sustainability during software systems development and maintenance that decomposes sustainability into four aspect in software development life cycle such as the development process, maintenance process, system production and system usage [11]. This approach is useful for a process engineer who instantiates this approach for a software development company or requirements engineer who instantiates it for a specific system under development.

The Software Sustainability Design Catalogue (SSDC) that quantifies sustainability via a series of guidelines used for incorporating sustainability into the design loop for software system. The SSDC is created to promote effective sustainability engineering and integration in phases of software development life cycle. Design according to the authors Oyedeji et al. [12] is a good way to achieve sustainability in software development.

Furthermore a checklist and guide approach that demonstrates how to include the objective of environmental sustainability from the very early steps of software development can assist in identifying key stakeholders. This will facilitate the ability to accommodate new objectives of improving the environmental sustainability of software systems [13]. Roher et al. [14] suggests the use of sustainability requirement patterns (SRPs), which will provide software engineers with guidance on how to write specific types of sustainability requirements with the goal to overcome the barriers of incorporating environmental sustainability into the requirements engineering process.

B. *Green software system*

Green software is an environmentally friendly software that consumes less energy, provides less impacts on environment and support carbon management [15]. It is also software that fulfils high level requirements, ensuring the software engineering process, maintenance, and disposal saves and/or reduces resource waste [16] [17]. Green software is divided into four parts: software that is energy efficient during execution, software that are embedded to execute and support smart operations in green manner, software to produce environment viable products and policies [18]. The goal of green software engineering is to provide supports for efficient consumption of natural resources while continuously monitoring, evaluating and optimizing the aftermath effects caused during the software system life cycle [19].

Erdélyi [20] paper provides an overview of different activities and advice on what to do in order to develop green software which uses energy efficiently and produce less waste. The paper highlights three ways software engineering can be green such as: produce green software, produce software to support environmentally consciousness (green by software) and produce less waste during development.

Dick et al. [21] provides basis for the right way to engineer green software systems using development process that ensures that the positive and negative effects of the software is continuously monitored and evaluated in order to optimize the software over its life cycle to be more green (environmental friendly).

Colmant et al. [22] presented researches on to improve the software-energy efficiency on multi-core systems. Colmant et al. [28] motivations were driven by the huge impact of the ICT on the world CO2 emissions which represents 2%. Calero et al. [4] highlights some of the meanings of green software notably a software that consumes less energy to run and produces as little waste as possible during its development and operation. Largely, research on green software has focused more on energy consumption and environmentally friendly software systems.

C. *Software for Sustainability*

There has been some interest in various domains such as manufacturing, energy sector, transportation and for different application in recycling, product packaging, data center setup, gas emissions. Some of the good examples are in grid computing, in Human Computer Interaction (HCI) to change the habit of people.

In [23], authors presented a software system that support sustainable lifestyles with an example of a domestic plant guild to show how sustainable human systems can effectively support a sustainable lifestyle, which can reduce the cost of living as well as the ecological footprint. Penzenstadler et al. [24] highlights vision for systems that will be supporting sustainability in the future (2029) with a set of fictional abstracts around the concepts of sustainability, complexity, collapse, and resilience of ICT systems.

Software can also provide support for sustainability in different domains such as:

- The use of software systems for tracking gas emissions
- Software for climate and disaster prediction
- Smart infrastructural management software
- Enterprise carbon and energy management software
- Smart transportation software to reduce CO^2 emissions.
- Sustainability Knowledge and Learning Management software
- Software for environmental awareness on wildlife and plants

D. *Sustainability of the software ecosystem*

Today software systems are the pillars of the economy, the software eco system is probably the biggest system in the world we human created. Software eco system has been defined according to Jansen et al. [25] as a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and they operate through the exchange of information, resources and artifacts.

Thus, the sustainability of software ecosystem involves the sustainment of the global system of software systems and services covering aspect of how different sub systems form a huge interconnected system and all the interactions. It covers all different components such as hardware, software and network that is used to resolve complex relationships among companies/organizations in all the different sectors and industries [26].

Sustainability of the software ecosystem entails how can the system of software systems endure with the evolving user requirements and usage overtime with less negative impact on the environments, social, technical and humans. This means the ability of software ecosystem to continue to function and evolve irrespective of any glitch is some part of the ecosystem and should continuously fulfil users' needs.

III. Perceptions of Sustainability in/for Software Systems

In this research, we defined sustainability as a quality construct in the same ways other factors are defined (see, for example, the ISO 25 000 family of standards). In our perception sustainability aims to create balance in the way humans live, produce, and use products and services (resources) with the objective to have less negative impact on the environment and promote the wellbeing of all living species. This means the capacity of software systems to endure in certain ecosystems under current and future conditions while satisfying the needs of users today and tomorrow with minimum negative impact on the environment; at the same time supporting business growth and societal values.

Currently, the dimensions of software sustainability are known and classified into five: economic, environment, social, individual and technical [27] but there is currently no clear categorisation for the perceptions of sustainability in/for software engineering. This section explains the categorization of software sustainability perceptions based on the literature review from the background section. Software sustainability evolution today can be perceive from one of the following perception (see Figure 1); Sustainability in Software Development, Software for Sustainability, Green Software Systems, Sustainability of Software Ecosystems.

- Sustainability in software development (Development): this refers to the processes involve in the development of software (software development life cycle).

- Software for sustainability (Usage): how software are used to support sustainability, an example is a software in fridge to minimize energy wastage (embedded software).
- Green software systems (Focused impact): software systems that uses less energy resource and promotes policies that supports green awareness.
- Sustainability of software ecosystems (Net effect): This is the total impact of the entire software ecosystem (systems of system)

The advancement of software sustainability from the four perceptions has received different level of research attention and contributions. Sustainability in software development, Green Software system we observed has the most important advancement in research compared to software for sustainability and sustainability of software ecosystem that were not fully explored.

Figure 1 portrays the categorization of software sustainability perceptions.

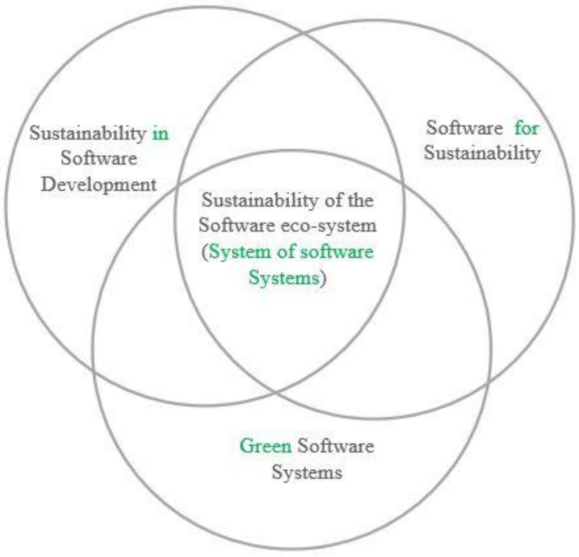
Sustainability in Software Development
Software for Sustainability
Sustainability of the Software eco-system (System of software Systems)
Green Software Systems

Figure 1. Sustainability Perceptions in/for Software Engineering

IV. MEASURES AND MEASUREMENT OF GREEN AND SUSTAINABILITY IN/FOR SOFTWARE SYSTEMS

This section presents different research work relating to green and sustainable software system measures and measurement. According to Britannica [28], measurement is the science of assigning of a quantity, either quantitative or qualitative, to a characteristic of an object or event, while making it comparable to other objects or events. Here object is the software and event is the development process. Sustainability measurement is still a new idea [29] [30] [31] [32]. Indeed, Lami et al. [31] stated that there are few studies about 'what' aspects of sustainability to measure and 'how' to do it. Calero et al. [33] highlighted that nowadays, sustainability is a key factor that should be considered in the software quality models, though there has less research channelled towards sustainability measurement. Seacord et al. [29] stated that planning and management of software sustainment is impaired by a lack of consistently applied, practical measures, and there is no central theoretical framework on measurement of software sustainability.

One of the most referenced sustainability measurement model for software system is the GREENSOFT Model [7]. It is a conceptual reference model for "Green and Sustainable Software", which has the objective to support software developers, administrators, and software users in creating, maintaining, and using software in a more sustainable way [34]. Another software sustainability measurement approach is the Sustainable Business Goal Question Metric (S-BGQM) [35]. It encourages the incorporation and measurement of sustainability during the entire software system development processes. Kramer [36] also wrote about sustainability measurement by proposing some set of questions that should be addressed by any sustainability framework.

A study for monitoring software energy hotspot proposed power model for software energy cost formula as Esoftware = Ecomp +Ecom +Einfra, where Ecomp is the computational cost (i.e., CPU process- ing, memory access, I/O operations), Ecom is the cost of exchanging data over the network, and Einfra is the addi- tional cost incurred by the OS and runtime platform (e.g., Java VM) [37]. The study focused on energy consumption of CPU and network demanding software at different levels of granularity. Also, the formula proposed for software energy efficiency (EF) is UsefulWorkDone/UserdEnergy [38]. This generic measure provide a way for evaluating the energy consumption of different software parts and modules using white box testing to measure which parts are consuming more energy and to see which parts can be optimized for efficient energy usage.

In order to facilitate research on energy usage attribution, software energy footprint lab was setup to provide insight on energy footprint measurements with results interpreting hardware dissipation profiles for various servers under different kinds of software stress [39]. This shows the relations between different hardware resource and the amount of resource required by the running software in relation to the power consumption.

Furthermore, a support tool is presented to analyze legacy systems in order to estimate the energy consumption and detect parts of the system with higher energy consumption. Using the profiling technique, the tool instrument legacy Java systems in order to keep track of its execution. This information, together with the energy consumption, enables the engineer to analyze legacy system consumption detecting energy peaks in the system [40].

Additionally, a modular Energy-Aware Computing Framework (EACOF) is proposed as a way to allow access to energy consumption information of software through API calls. The EACOF is separated into two task for collection and utilization of dynamic energy consumption data which reduce development and maintenance overhead required for the successful completion of each task[41]. Another approach is also proposed for monitoring power consumption of software in order to assist software designers and developer to reduce software power consumption and have better energy efficiency [42]. This approach currently monitors power consumption at source code level, this approach will provide better insights on software energy consumption if extended to the hardware running the software.

As summarized in Table 1 and the research work detailed in [43] [44] [45] and [46], other measures of green and sustainable software have been on software and hardware energy consumption with less research for measures covering software sustainability dimensions such as individual, social, economic and software sustainability perceptions (Software for sustainability and Software ecosystem).

The measures detailed in Table 1 are structure based on categorization of software sustainability and green measures for software sustainability dimensions and the four sustainability perceptions. Each column after the main title has a "YES or No" to indicate if the proposed measure in the research paper cover any of the categories listed in Table 1. Most of the measures descriptions does not explicitly indicate that the authors considered sustainability dimensions. Base on the descriptions and explanations of the authors for all measures, we have categorized those measures according to the right sustainability dimension (Economic, Social, Individual, Technical and Environment) to show how it relates to the four sustainability perceptions.

TABLE I. MEASURES FOR GREEN AND SUSTAINABLE SOFTWARE LINKED TO SUSTAINABILITY DIMENSIONS AND PERCEPTIONS

Name	**Definition**	**Formula**	**Software Development Lifecycle**	**Green Software**	**Software for sustainability**	**Software ecosystem**	**Sustainability Dimensions**
[37] Software energy cost	The computational cost of performing task involving CPU processing, memory access, I/O operations and exchanging data over the network.	Esoftware = Ecomp +Ecom +Einfra where Ecomp is the computational cost (i.e., CPU process- ing, memory access, I/O operations), Ecom is the cost of exchanging data over the network, and Einfra is the addi- tional cost incurred by the OS and runtime platform (e.g., Java VM)	Yes	Yes	No	No	Environment
[39] Software energy footprint	Not stated	Experimental lab setup details can be found in [39]	No	Yes	No	No	Environment
Energy Efficiency (EF) [47]	Not stated	Energy Efficiency = UsefulWorkDone /UsedEnergy	No	Yes	No	No	Environment, Technical
Performance Efficiency (PE) [48]	Not stated, sub-characteristics measure listed as Time behavior, Resource utilization, capacity		Yes	Yes	No	No	Environment
Power Usage Effectiveness (PUE) [49]	The ratio of facilities energy (supply side) to IT equipment energy (demand size)	PUE= Total Facility Energy/IT equipment Energy	No	Yes	No	No	Environment, Technical
Performance [50]	Not stated	Not available	No	Yes	No	No	Environment, Technical
Efficiency [50]	Not stated, third level indicators provided as: Time Behaviour, Resource Utilization	Not available	Yes	Yes	No	No	Environment, Technical
Resource usage [50]	Not stated, third level indicators provided as: CPU Usage, I/O Usage, Memory Usage, Storage Usage	Not available	Yes	Yes	No	No	Technical
Energy impact [50]	Not stated, third level indicators provided as: Energy Consumption, CO2 Emission, Green Energy Usage	Not available	Yes	Yes	No	No	Environment
Energy efficiency (Speedup Greenup, Powerup, and) [51]	Speedup is defined as the ratio of serial code runtime over parallel code runtime. Greenup is the ratio of the total energy consumption of the non-optimized code (Eφ) over the total energy consumption of the optimized code (Eo). Powerup implies the power effects of an optimization. A less than	Speedup=Tφ/To where Tφ is the total execution time of non-optimized code, and To is the total execution time of the optimized code. Greenup = Eφ/Eo Assuming, Pφ is the average power consumed by the non-optimized code and Po is the average power consumed by the	No	Yes	No	No	Environment, Technical

	1 Powerup implies power savings while a greater than 1 Powerup indicates that the optimized code consumes more power in average.	optimized code Powerup =Po /Pφ= Speedup /Greenup					
Software Project's Footprint [30]	Natural resources and environ- mental impact used during software development.	Transportation from/to the office, and Long-haul trips. Example used in the article: Work-From-Home Days: 2 days out of 165 total team- days (33 project days * 5 team members)=1.21% Long-Haul Roundtrips: By airplane: 6; By train: 0.	Yes	No	No	No	Environment
Functional Suitability (FS) [48]	Functional Completeness, Functional correctness, Functional appropriateness	Not available	Yes	Yes	No	No	Technical
Compatibility [48]	Not stated, sub-characteristics measure listed as Co-existence, Interoperability	Not available	Yes	Yes	No	No	Technical
Usability [48]	Not stated, sub-characteristics measure listed as Appropriateness recognizability, Learnability, Operability, User error protection, User interface eesthetics	Not available	Yes	Yes	No	No	Technical, Individual
Reliability [48]	Not stated, sub-characteristics measure listed as Maturity, Availability, Fault tolerance, Recoverability	Not available	Yes	Yes	No	No	Technical
Portability [48]	Not stated, sub-characteristics measure listed as Adaptability. Installability, Replaceability	Not available	Yes	Yes	No	No	Technical

V. DISCUSSION

Table 1 provides details of measures attributed to green and sustainable software. From Table 1, it can be identified that most measures focused on energy efficiency or power consumptions. With most focus on green software, there is a limitation on having a holistic approach towards software sustainability measurement. The measures of software sustainability should consider the following:

- Human (End users) system interaction: involves the measures of the system sustainability based on how it impacts on users and their level of awareness about sustainability and green. It entails the well-being of the software users' community and the changing of the human mindset.
- Software system developers: evaluate the sustainability of the processes and practices for the development and integration of sustainability in software systems.

One of the key question/concern that should be clearly answered by a sustainability measurement framework is the difference between the different scales of software measurement and the interpretation of these scales of measurement for sustainability. The problem of software sustainability measurement is not only in measuring but rather giving meaningful interpretation of what the measurement means. For example today, fridges are categorized using A+, A++ and A+++ for quantifying and measuring its energy efficiency. Normally A+ consumes less energy, A++ has better energy efficiency than A+ and A++ has the best energy efficiency in today market. According to the EU Directive 92/75/EC which established an energy consumption labelling scheme [52], there are different descriptions of the measures that quantify why Fridge is labelled A+, A++ or A++ based on its energy consumption. In the same line, there is need for a foundation or framework to ground the different measures for software sustainability measures and measurement with clear interpretation.

Currently, there is not enough firm scientific basis for important choices on how sustainability related factors should be defined and measured, the varying purposes for which the measures are used. This makes it difficult to effectively and efficiently evaluate software sustainability using the right measures.

VI. CONCLUSION

In this position paper, we summarized the research results on the categorization of software sustainability perceptions. Using the identified four perceptions of software sustainability, we referenced the current measures to each of the four perceptions. The major focus of all identified green and sustainable software measures are on green software. Energy efficiency has received the most attention. Research work is needed to identify and assess the validity of other measures related to the other perceptions. Research on measures of sustainability has to be grounded in the tradition and theory of software measurement. This requires considering software sustainability as a quality attribute and define it in the same way other attributes are defined.

References

[1] United Nations, *World Economic and Social Survey 2013*. 2013.

[2] G. saval Martin, mahaux, patrick heymans, "Requirements Engineering: Foundation for Software Quality," *Requir. Eng. Found. Softw. Qual.*, vol. 4542, no. January, pp. 247–261, 2007.

[3] Ericsson, "Technology for Good," *Available online: https://www.ericsson.com/assets/local/about-ericsson/sustainability-and-corporate-responsibility/documents/2015-corporate-responsibility-and-sustainability-report.pdf Accessed on 30-11-2017*, 2014.

[4] C. Calero and M. Piattini, "Introduction to Green in software engineering," *Green Softw. Eng.*, pp. 1–327, 2015.

[5] N. Condori-Fernandez, G. Procaccianti, and N. Ali, "Metrics for green and sustainable software: MeGSuS 2014," in *Proceedings - 2014 Joint Conference of the International Workshop on Software Measurement, IWSM 2014 and the International Conference on Software Process and Product Measurement, Mensura 2014*, 2014, pp. 62–63.

[6] B. Penzenstadler and A. Fleischmann, "Teach Sustainability in Software Engineering?," in *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, 2011.

[7] 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, 2011.

[8] A. Raturi, B. Penzenstadler, B. Tomlinson, and D. Richardson, "Developing a sustainability non-functional requirements framework," *Proc. 3rd Int. Work. Green Sustain. Softw. - GREENS 2014*, pp. 1–8, 2014.

[9] C. C. Venters *et al.*, "The Blind Men and the Elephant Towards an Empirical Evaluation Framework for Software Sustainability," vol. 2, no. 1, pp. 1–6, 2014.

[10] B. Penzenstadler, A. Raturi, D. Richardson, and B. Tomlinson, "Safety, security, now sustainability: The nonfunctional requirement for the 21st century," *IEEE Softw.*, vol. 31, no. 3, pp. 40–47, 2014.

[11] B. Penzenstadler, "Supporting Sustainability Aspects in Software Engineering," *3rd Int. Conf. Comput. Sustain.*, pp. 1–4, 2013.

[12] S. Oyedeji, A. Seffah, and B. Penzenstadler, "A catalogue supporting software sustainability design," *Sustainability*, vol. 10, no. 7, pp. 1–30, 2018.

[13] B. Penzenstadler, "Infusing green: Requirements engineering for green in and through software systems," *3rd Intl. Work. Requir. Eng. Sustain. Syst. 2014*, vol. 1216, no. 1, pp. 44–53, 2014.

[14] K. Roher and D. Richardson, "Sustainability requirement patterns," *2013 3rd Int. Work. Requir. Patterns, RePa 2013 - Proc.*, pp. 8–11, 2013.

[15] S. Murugesan and G. Gangadharan, *Harnessing Green IT : Principles and Practices*, no. September. John Wiley & Sons, Ltd, 2012.

[16] T. Juha, "Good, bad, and beautiful software. In search of green software quality factors," *CEPIS Upgrad. XII 422–27*, no. July, 2011.

[17] L. Erdmann Hilty, L., Goodman, J., Arnfalk, P. and L. Erdmann Hilty, L., Goodman, J., Arnfalk, P., "The Future Impact of ICTs on Environmental Sustainability," *IPTS Publ.*, no. August, p. 68, 2004.

[18] M. N. Malik and H. H. Khan, "Investigating Software Standards: A Lens of Sustainability for Software Crowdsourcing," *IEEE Access*, vol. 6, pp. 5139–5150, 2018.

[19] J. T. Kern Eva, Markus Dick, Naumann Stefan, Guldner Achim, "Green software and green software engineering - definitions, measurements, and quality aspects," *Proc. First Int. Conf. Inf. Commun. Technol. Sustain. ETH Zurich, Febr. 14-16, 2013*, no. January, pp. 175–182, 2013.

[20] K. Erdélyi, "Special factors of development of green software supporting eco sustainability," *SISY 2013 - IEEE 11th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 337–340, 2013.

[21] M. Dick and S. Naumann, "Enhancing software engineering processes towards sustainable software product design," *24th Int. Conf. Informatics Environ. Prot. (EnviroInfo 2010)*, vol. 2010, pp. 706–715, 2010.

[22] M. Colmant, R. Rouvoy, and L. Seinturier, "Improving the energy efficiency of software systems for multi-core architectures," *Proc. 11th Middlew. Dr. Symp. MDS 2014 - co-located with ACM/IFIP/USENIX 15th Int. Middlew. Conf.*, pp. 2–5, 2014.

[23] J. Norton, A. J. Stringfellow, J. J. L. Jr, B. Penzenstadler, and B. Tomlinson, "Domestic Plant Guilds : A Software System for Sustainability," vol. i, 2013.

[24] B. Penzenstadler *et al.*, "ICT4S 2029 : What will be the Systems Supporting Sustainability in 15 Years ?," 2014.

[25] S. Jansen and M. Cusumano, "Defining software ecosystems: a survey of software platforms and

business network governance : Software Ecosystems Analyzing and Managing Business Networks in the Software Industry," *Softw. Ecosyst. Anal. Manag. Bus. Networks Softw. Ind.*, pp. 13–28, 2013.

[26] J. V. Joshua, D. O. Alao, S. O. Okolie, and O. Awodele, "Software Ecosystem: Features, Benefits and Challenges," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 8, pp. 1–6, 2013.

[27] B. Penzenstadler and H. Femmer, "A generic model for sustainability with process- and product-specific instances," *GIBSE 2013 - Proc. 2013 Work. Green Softw. Eng. Green by Softw. Eng.*, no. June 2015, pp. 3–7, 2013.

[28] E. Britannica, "Measurement instruments and systems. Accessed on 7-12-2017 from: https://www.britannica.com/print/article/371701," pp. 1–2, 2017.

[29] R. Seacord *et al.*, "Measuring Software Sustainability," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[30] F. Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang, and C. Liu, "Measuring the Sustainability Performance of Software Projects," *2010 IEEE 7th Int. Conf. E-bus. Eng.*, pp. 369–373, 2010.

[31] G. Lami and L. Buglione, "Measuring software sustainability from a process-centric perspective," *Proc. 2012 Jt. Conf. 22nd Int. Work. Softw. Meas. 2012 7th Int. Conf. Softw. Process Prod. Meas. IWSM-MENSURA 2012*, pp. 53–59, 2012.

[32] M. R. Idio, "Measuring Sustainability Impact of Software," vol. 16, no. 1, pp. 5–7, 2014.

[33] C. Calero, M. F. Bertoa, and M. Angeles Moraga, "A systematic literature review for software sustainability measures," *Green Sustain. Softw. ({GREENS)}, 2013 2nd Int. Work.*, pp. 46–53, 2013.

[34] 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, 2011.

[35] S. Oyedeji, A. Seffah, and B. Penzenstadler, "Sustainability Quantification in Requirements Informing Design," *6th Int. Work. Requir. Eng. Sustain. Syst.*, vol. i, 2017.

[36] K.-L. Kramer, *User Experience in the Age of Sustainability*. 2012.

[37] a Noureddine, A. Bourdon, R. Rouvoy, and L. Seinturier, "Runtime monitoring of software energy hotspots," in *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on*, 2012, pp. 160–169.

[38] T. Johann *et al.*, "How to measure energy-efficiency of software : Metrics and measurement results," no. April 2015, pp. 51–54, 2012.

[39] M. A. Ferreira, E. Hoekstra, B. Merkus, B. Visser, and J. Visser, "Seflab: A lab for measuring software energy footprints," in *2013 2nd International Workshop on Green and Sustainable Software, GREENS 2013 - Proceedings*, 2013, pp. 30–37.

[40] V. Cordero, I. G. R. De Guzmán, and M. Piattini, "A first approach on legacy system energy consumption measurement," in *Proceedings - 2015 IEEE 10th International Conference on Global Software Engineering Workshops, ICGSEW 2015*, 2015, pp. 35–43.

[41] H. Field, G. Anderson, and K. Eder, "EACOF: A Framework for Providing Energy Transparency to enable Energy-Aware Software Development," pp. 1194–1199, 2014.

[42] S. Cagri, C. Furkan, C. James, K. Fouad, P. Lori, and W. Kristina, "Towards Power Reduction Through Improved Software Design," pp. 1–8, 2007.

[43] P. Bozzelli, Q. Gu, and P. Lago, "A systematic literature review on green software metrics," *Sis.Uta.Fi*, 2013.

[44] E. Kern, M. Dick, S. Naumann, A. Guldner, and T. Johann, "Green Software and Green Software Engineering – Definitions , Measurements , and Quality Aspects," pp. 87–94, 2013.

[45] T. Debbarma and K. Chandrasekaran, "Green measurement metrics towards a sustainable software: A systematic literature review," *2016 Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2016*, 2017.

[46] R. V. O. Connor and A. D. Eds, *Software Process Improvement and Capability Determination*, vol. 155, no. June. 2011.

[47] T. Johann, M. Dick, S. Naumann, and E. Kern, "How to measure energy-efficiency of software: Metrics and measurement results," *2012 1st Int. Work. Green Sustain. Software, GREENS 2012 - Proc.*, pp. 51–54, 2012.

[48] G. Oleksandr, K. Vyacheslav, and F. Mario, "Software Quality Standards and Models Evolution: Greenness and Reliability Issues," vol. 469, pp. 277–299, 2016.

[49] E. Rondeau, F. Lepage, J. Georges, E. Rondeau, F. Lepage, and J. Georges, "Measurements and Sustainability," 2015.

[50] S. A. . Koçak, G. I. . Alptekin, and A. B. . Bener, "Evaluation of software product quality attributes and environmental attributes using ANP decision framework," *CEUR Workshop Proc.*, vol. 1216, pp. 37–44, 2014.

[51] S. Abdulsalam, Z. Zong, Q. Gu, and M. Qiu, "Using the Greenup, Powerup, and Speedup metrics to evaluate software energy efficiency," *2015 6th Int. Green Sustain. Comput. Conf.*, 2016.

[52] A. Michel, S. Attali, and E. Bush, "Energy efficiency of White Goods in Europe : monitoring the market with sales data," no. June, pp. 1–53, 2015.