Sustainable Software Engineering: Visions and Perspectives beyond Energy Efficiency

Christoph König
Karlsruhe Institute of Technology
Karlsruhe, Germany
christoph.koenig@kit.edu

ABSTRACT

In the face of multiple global crises such as climate change, a transformation towards sustainable development is more urgent than ever. Digitalization, as a fundamental change in society and the economy, offers great opportunities for sustainable development, but also poses its own threats, as evident in the immense resource consumption and growing surveillance tendencies. To leverage digitalization for sustainability transformation without compromising it, software engineering requires a significant shift in practices and structures. However, research in this area is still immature, lacking a deeper understanding of sustainability, its application in practice and solid engineering approaches. To bridge these gaps, this thesis aims to operationalize sustainability by proposing sustainability goals for software engineering, followed by the development of novel assessment methods and appropriate tool support.

CCS CONCEPTS

• **Software and its engineering → Software design engineering**.

KEYWORDS

Sustainable development, Sustainability, Sustainable Software Engineering

ACM Reference Format:
Christoph König. 2024. Sustainable Software Engineering: Visions and Perspectives beyond Energy Efficiency. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3639478.3639782

1 MOTIVATION

In the age of the Anthropocene [36], human impact on the Earth's ecosystem progresses unabated. By 2023, six of nine planetary boundaries are crossed, including the atmospheric CO_2 concentration and the release of microplastic, pesticides, and nuclear waste into the environment [28, 29]. As a result, social conflicts and economic damage are more likely to occur, affecting not only the less developed countries of the global South, but increasingly the industrialized countries in the global North as well [2].

ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0502-1/24/04.
https://doi.org/10.1145/3639478.3639782

In the face of these enormous impacts, the call for *sustainable development* is growing louder. In 1987, the UN World Commission on Environment and Development defined sustainable development as permanent development that meets "the needs of the present without compromising the ability of future generations to meet their own needs" [1]. Accordingly, inter- and intragenerational equity, a global perspective and the linking of environmental and development aspects are constitutive elements for sustainability.

Digitalization as a fundamental transformation in society and economy is seen by many [10, 18, 31] as a great opportunity to achieve sustainable development. For example, digital precision agriculture or smart mobility systems could enable large reductions in fertilizer and pesticide use, land use, or pollutant emissions. However, as many reports also note [17, 19, 32], current trends in digitalization have the opposite effect of "optimising the unsustainable status quo rather than transforming it" [19], as evident in the immense energy consumption, hazardous electronic waste, tendencies of mass surveillance, and increasing polarization.

To realize the benefits of digitalization for sustainability without causing the opposite effects, fundamental changes in today's software engineering are inevitable. These changes require not only a rethinking of current economic patterns, but also technological innovation and the empowerment of social actors. Software engineering is challenged to shape this change through new structures, practices, and techniques that align software development with the goal of sustainability. In the past, several approaches have been proposed to address this challenge, such as risk assessment [8, 9], stakeholder involvement [4, 27], or technical solutions such as Green IT [6, 22], which aims to reduce the energy consumption and carbon footprint of digital systems. However, as several studies report, sustainability research in software engineering is still in its infancy, lacking empirical evidence [20], transformative research approaches and perspectives beyond energy efficiency [30].

2 PROBLEM STATEMENT

The key barrier to current sustainability research and practice in software engineering is the implicit and superficial understanding of sustainability, its relation to software development, and its translation into concrete engineering practices. Indicators of this lack of knowledge are, for instance, the one-sided focus on energy consumption in software engineering publications [21, 30], the various definitions of sustainability in software engineering [13, 26, 35] as well as the missing guidelines at conferences like ICSE, for when a technology is actually considered as sustainable. As a result, sustainability in software engineering is currently a loose concept with few points of reference, not reaching its potential to shape software development practices, processes, and decisions.

To overcome this barrier and make sustainability a central concern in software engineering, sustainability needs to be made tangible in the context of software development. This requires not only an advanced understanding of sustainability and its translation into concrete engineering goals, but also an assessment methodology with appropriate tool support. To achieve this operationalization, the thesis investigates the following main research question (**MRQ**):

MRQ: How can we operationalize sustainability in software engineering?

3 RESEARCH GAPS

Based on this MRQ, the following research gaps (**G1–G3**) have been identified:

[G1] Current software engineering lacks a holistic understanding of sustainability, which is reflected in a variety of diffuse definitions of sustainability and references to software. For example, sustainability has been described as either "the capacity to endure" [34], "preserving the function of a system over a defined time span" [26] or, in relation to software, as "composite, non-functional requirement" [33], such as extensibility and maintainability. While these definitions cover some aspects of sustainable development as defined by the UN Commission [1], they do not fully encompass sustainability in its social, environmental and economic dimensions. As a result, software engineering fails to provide guidance on what sustainability goals need to be achieved, as evident in existing concepts such as the *Karlskrona Manifesto for Sustainability Design* [5, 25], which tend to be vague and difficult to apply.

[G2] In addition, software engineering lacks appropriate methodologies for assessing sustainability in software development. Sustainability assessments are an important tool in various fields to evaluate the status of certain sustainability goals and to derive appropriate measures [3]. In software engineering, for example, Kern et al. [14] and Naumann et al. [24] proposed a set of criteria such as resource efficiency, hardware operating life and user autonomy to evaluate the sustainability of software products, along with an assessment methodology. Another approach [33] suggested to evaluate the sustainability of a software using a set of core quality attributes such as maintainability or reusability. However, due to the lack of theory and the ambiguity of sustainability definitions as described in G1, current assessment methods in software engineering appear to be theoretically weak, resulting in partial or inadequate assessments.

[G3] Achieving sustainability requires not only technical and structural innovation, but also continuous management. This includes the support of appropriate tools that systematically document sustainability goals and measures at the software engineering level. However, due to the lack of theory and methodology as described in G1 and G2, software engineering lacks tool support for sustainability assessments. As a result, software engineers are not able to apply, document, and evaluate the impact of certain sustainability-related actions such as techniques [11, 22], process models [23], or stakeholder engagement [4, 27] on certain sustainability goals. This may also contribute to the fact that, according to recent secondary literature [20], few of these approaches seem to be applied in practice or even empirically validated.

4 RESEARCH METHODOLOGY

In order to fill the research gaps and answer the main question, the thesis examines the following sub research questions (**RQ1–RQ3**):

RQ1: What does sustainability mean for software engineering?

To operationalize sustainability in software engineering, sustainability needs to be related to the specific context of software development. RQ1 therefore focuses on translating sustainability into a set of common sustainability goals, such as resource conservation, digital participation, privacy or safety, that provide orientation and guidance in software engineering. For this purpose, I draw on the *Integrative Concept of Sustainable Development* (IKoNE) [15], which has proven to be a well-founded theory and research instrument in the German sustainability discourse [16]. IKoNE concretizes the idea of inter- and intragenerational equity through three main goals, which are further specified by 15 substantive and 10 instrumental rules. These rules need to be interpreted in the context of digitalization in order to derive sustainability goals to be met in software development. These goals will then serve as the basis for a more precise definition of *Sustainable Software Engineering*. RQ1 therefore closes G1 and presents a theoretical contribution, based on the foundations of sustainability science.

RQ2: How can we assess sustainability in software engineering?

RQ2 aims to develop a methodology for sustainability assessments in software engineering. For this purpose, I will define a suitable set of target indicators based on the sustainability goals derived in RQ1. This set of target indicators may include both product-specific and process-specific indicators at the software development and enterprise levels, along with measurement methods for certain indicators. To develop an assessment methodology, I will draw on existing methodologies from other disciplines, such as life cycle assessments [12], as well as general assessment methods from software engineering, such as maturity models [7]. To evaluate the usefulness and applicability of the intended set of indicators and methodology, I will conduct a case study with a software engineering company. Overall, RQ2 closes G2 and provides a theoretical contribution along with an empirical evaluation.

RQ3: How can we support sustainability assessments in software engineering?

To enable continuous sustainability management in software engineering, RQ3 explores ways to provide appropriate tool support for sustainability assessments. For this purpose, I will use the sustainability goals from RQ1 and their indicators from RQ2 to develop a tool for structured and comparable sustainability analysis according to the proposed methodology. This will be preceded by a requirements analysis of the necessary features, technical resources and the possibility of manual and automated data collection for certain indicators. The final tool will be qualitatively evaluated in practice in collaboration with a software engineering company. Thus, RQ3 provides a technical contribution with an empirical evaluation in order to close G3.

REFERENCES

[1] 1987. *Our common future.* Oxford University Press, Oxford. OCLC: 15489268.

[2] 2023. IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland. https://doi.org/10.59327/IPCC/AR6-9789291691647

[3] Lisa Andes, Thomas Lützkendorf, Benjamin Ströbele, Jürgen Kopfmüller, and Christine Rösch. 2019. Methodensammlung zur Nachhaltigkeitsbewertung Grundlagen, Indikatoren, Hilfsmittel. https://publikationen.bibliothek.kit.edu/1000128806

[4] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M. Easterbrook, Birgit Penzenstadler, Norbet Seyff, and Colin C. Venters. 2016. Requirements: The Key to Sustainability. *IEEE Software* 33, 1 (Jan. 2016), 56–65. https://doi.org/10.1109/MS.2015.158

[5] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. 2015. Sustainability Design and Software: The Karlskrona Manifesto. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE, Florence, Italy, 467–476. https://doi.org/10.1109/ICSE.2015.179

[6] Coral Calero and Mario Piattini. 2015. *Green in Software Engineering.* Springer International Publishing, Cham.

[7] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum. 2011. *CMMI for development: guidelines for process integration and product improvement ; [CMMI-DEV, Version 1.3]* (3. ed ed.). Addison-Wesley, Upper Saddle River, NJ Munich.

[8] Leticia Duboc, Stefanie Betz, Birgit Penzenstadler, Sedef Akinli Kocak, Ruzanna Chitchyan, Ola Leifler, Jari Porras, Norbert Seyff, and Colin C. Venters. 2019. Do we Really Know What we are Building? Raising Awareness of Potential Sustainability Effects of Software Systems in Requirements Engineering. In *2019 IEEE 27th International Requirements Engineering Conference (RE).* 6–16. https://doi.org/10.1109/RE.2019.00013 ISSN: 2332-6441.

[9] Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff, and Colin C. Venters. 2020. Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow. *Requirements Engineering* 25, 4 (Dec. 2020), 469–492. https://doi.org/10.1007/s00766-020-00336-y

[10] Martina Fromhold-Eisebith, Ulrike Groth, Ellen Matthies, Dirk Messner, Karen Pittel, Hans-Joachim Schellnhuber, Ina Schieferdecker, Sabine Schlacke, and Uwe Schneidewind. 2019. *Towards our common digital future.* German Advisory Council on Global Change (WBGU), Berlin, Germany.

[11] Stefanos Georgiou, Stamatia Rizou, and Diomidis Spinellis. 2019. Software Development Lifecycle for Energy Efficiency: Techniques and Tools. *Comput. Surveys* 52, 4 (Aug. 2019), 81:1–81:33. https://doi.org/10.1145/3337773

[12] Michael Z. Hauschild, Ralph K. Rosenbaum, and Stig Irving Olsen (Eds.). 2018. *Life Cycle Assessment: Theory and Practice.* Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-56475-3

[13] Lorenz M. Hilty and Bernard Aebischer. 2015. ICT for Sustainability: An Emerging Research Field. In *ICT Innovations for Sustainability (Advances in Intelligent Systems and Computing)*, Lorenz M. Hilty and Bernard Aebischer (Eds.). Springer International Publishing, Cham, 3–36. https://doi.org/10.1007/978-3-319-09228-7_1

[14] Eva Kern, Lorenz M. Hilty, Achim Guldner, Yuliyan V. Maksimov, Andreas Filler, Jens Gröger, and Stefan Naumann. 2018. Sustainable software products—Towards assessment criteria for resource and energy efficiency. *Future Generation Computer Systems* 86 (Sept. 2018), 199–210. https://doi.org/10.1016/j.future.2018.02.044

[15] Jürgen Kopfmüller (Ed.). 2001. *Nachhaltige Entwicklung integrativ betrachtet: konstitutive Elemente, Regeln, Indikatoren.* Number 1 in Global zukunftsfähige Entwicklung - Perspektiven für Deutschland. Ed. Sigma, Berlin.

[16] Jürgen Kopfmüller (Ed.). 2006. *Ein Konzept auf dem Prüfstand: das integrative Nachhaltigkeitskonzept in der Forschungspraxis.* Number Bd. 12 in Global zukunftsfähige Entwicklung - Nachhaltigkeitsforschung in der Helmholtz-Gemeinschaft. Ed. Sigma, Berlin.

[17] Steffen Lange, Johanna Pohl, and Tilman Santarius. 2020. Digitalization and energy consumption. Does ICT reduce energy demand? *Ecological Economics* 176 (Oct. 2020), 106760. https://doi.org/10.1016/j.ecolecon.2020.106760

[18] Steffen Lange and Tilman Santarius. 2020. *Smart green world? Making Digitalization Work for Sustainability.* Routledge, Taylor & Francis Group, London; New York, NY.

[19] Steffen Lange, Tilman Santarius, Lina Dencik, Tomas Diez, and Hugues Ferreboeuf. 2023. *Digital Reset: redirecting technologies for the deep sustainability transformation.* oekom, München.

[20] Sean McGuire, Erin Schultz, Bimpe Ayoola, and Paul Ralph. 2023. Sustainability is Stratified: Toward a Better Theory of Sustainable Software Engineering. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE).* 1996–2008. https://doi.org/10.1109/ICSE48619.2023.00169 ISSN: 1558-1225.

[21] Brunna C. Mourão, Leila Karita, and Ivan do Carmo Machado. 2018. Green and Sustainable Software Engineering - a Systematic Mapping Study. In *Proceedings of the XVII Brazilian Symposium on Software Quality (SBQS '18).* Association for Computing Machinery, New York, NY, USA, 121–130. https://doi.org/10.1145/3275245.3275258

[22] San Murugesan. 2008. Harnessing Green IT: Principles and Practices. *IT Professional* 10, 1 (Jan. 2008), 24–33. https://doi.org/10.1109/MITP.2008.10 Conference Name: IT Professional.

[23] Stefan Naumann, Markus Dick, Eva Kern, and Timo Johann. 2011. The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems* 1, 4 (Dec. 2011), 294–304. https://doi.org/10.1016/j.suscom.2011.06.004

[24] Stefan Naumann, Achim Guldner, and Eva Kern. 2021. The Eco-label Blue Angel for Software—Development and Components. In *Advances and New Trends in Environmental Informatics (Progress in IS)*, Andreas Kamilaris, Volker Wohlgemuth, Kostas Karatzas, and Ioannis N. Athanasiadis (Eds.). Springer International Publishing, Cham, 79–89. https://doi.org/10.1007/978-3-030-61969-5_6

[25] Shola Oyedeji, Ahmed Seffah, and Birgit Penzenstadler. 2018. A Catalogue Supporting Software Sustainability Design. *Sustainability* 10, 7 (July 2018), 2296. https://doi.org/10.3390/su10072296 Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[26] Birgit Penzenstadler. 2013. Towards a definition of sustainability in and for software engineering. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13).* Association for Computing Machinery, New York, NY, USA, 1183–1185. https://doi.org/10.1145/2480362.2480585

[27] Birgit Penzenstadler, Henning Femmer, and Debra Richardson. 2013. Who is the advocate? Stakeholders for sustainability. In *2013 2nd International Workshop on Green and Sustainable Software (GREENS).* 70–77. https://doi.org/10.1109/GREENS.2013.6606424

[28] Linn Persson, Bethanie M. Carney Almroth, Christopher D. Collins, Sarah Cornell, Cynthia A. de Wit, Miriam L. Diamond, Peter Fantke, Martin Hassellöv, Matthew MacLeod, Morten W. Ryberg, Peter Søgaard Jørgensen, Patricia Villarrubia-Gómez, Zhanyun Wang, and Michael Zwicky Hauschild. 2022. Outside the Safe Operating Space of the Planetary Boundary for Novel Entities. *Environmental Science & Technology* 56, 3 (Feb. 2022), 1510–1521. https://doi.org/10.1021/acs.est.1c04158 Publisher: American Chemical Society.

[29] Katherine Richardson, Will Steffen, Wolfgang Lucht, Jørgen Bendtsen, Sarah E. Cornell, Jonathan F. Donges, Markus Drüke, Ingo Fetzer, Govindasamy Bala, Werner von Bloh, Georg Feulner, Stephanie Fiedler, Dieter Gerten, Tom Gleeson, Matthias Hofmann, Willem Huiskamp, Matti Kummu, Chinchu Mohan, David Nogués-Bravo, Stefan Petri, Miina Porkka, Stefan Rahmstorf, Sibyll Schaphoff, Kirsten Thonicke, Arne Tobian, Vili Virkki, Lan Wang-Erlandsson, Lisa Weber, and Johan Rockström. 2023. Earth beyond six of nine planetary boundaries. *Science Advances* 9, 37 (Sept. 2023), eadh2458. https://doi.org/10.1126/sciadv.adh2458 Publisher: American Association for the Advancement of Science.

[30] Tilman Santarius and Josephin Wagner. 2023. Digitalization and sustainability: A systematic literature analysis of ICT for Sustainability research. *GAIA - Ecological Perspectives for Science and Society* 32, 1 (March 2023), 21–32. https://doi.org/10.14512/gaia.32.S1.5

[31] Peter Seele and Irina Lock. 2017. The game-changing potential of digitalization for sustainability: possibilities, perils, and pathways. *Sustainability Science* 12, 2 (March 2017), 183–185. https://doi.org/10.1007/s11625-017-0426-4

[32] Hannah Trittin-Ulbrich, Andreas Georg Scherer, Iain Munro, and Glen Whelan. 2021. Exploring the dark and unexpected sides of digitalization: Toward a critical agenda. *Organization* 28, 1 (Jan. 2021), 8–25. https://doi.org/10.1177/1350508420968184 Publisher: SAGE Publications Ltd.

[33] C. Venters, L. M. S. Lau, M. Griffiths, V. Holmes, R. Ward, C. Jay, C. Dibsdale, and J. Xu. 2014. The Blind Men and the Elephant: Towards an Empirical Evaluation Framework for Software Sustainability. *Journal of open research software* 2, 1 (July 2014). http://dx.doi.org/10.5334/jors.ao

[34] Colin C. Venters, Rafael Capilla, Stefanie Betz, Birgit Penzenstadler, Tom Crick, Steve Crouch, Elisa Yumi Nakagawa, Christoph Becker, and Carlos Carrillo. 2018. Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software* 138 (April 2018), 174–188. https://doi.org/10.1016/j.jss.2017.12.026

[35] C. C. Venters, C. Jay, L. M. S. Lau, M. K. Griffiths, V. Holmes, R. R. Ward, J. Austin, C. E. Dibsdale, and J. Xu. 2014. Software Sustainability: The Modern Tower of Babel. http://ceur-ws.org/Vol-1216/paper2.pdf Conference Name: RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems ISSN: 1613-0073 Meeting Name: RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems Pages: 7-12 Place: Karlskrona, Sweden Publisher: CEUR Volume: 1216.

[36] Colin N. Waters, Jan Zalasiewicz, Colin Summerhayes, Anthony D. Barnosky, Clément Poirier, Agnieszka Gałuszka, Alejandro Cearreta, Matt Edgeworth, Erle C. Ellis, Michael Ellis, Catherine Jeandel, Reinhold Leinfelder, J. R. McNeill, Daniel deB. Richter, Will Steffen, James Syvitski, Davor Vidas, Michael Wagreich, Mark Williams, An Zhisheng, Jacques Grinevald, Eric Odada, Naomi Oreskes, and Alexander P. Wolfe. 2016. The Anthropocene is functionally and stratigraphically distinct from the Holocene. *Science* 351, 6269 (Jan. 2016), aad2622. https://doi.org/10.1126/science.aad2622 Publisher: American Association for the Advancement of Science.