

Sustainable Software Engineering: Research Patterns and Trends through Artifacts from a Practitioner's Perspective

Michaela Degbeon¹, Markus Helfert² and Patrick Rigot-Muller³

Abstract: This study aims to uncover trends and patterns in sustainable software engineering research, with a particular focus on artifact-oriented outcomes and a practitioner's perspective. Continuous research on the topic of environmentally sustainable software engineering practices is essential to mitigate the environmental impact of software products and advance software processes that promote sustainability. Despite recognizing the issue, many software industry practitioners struggle to identify sustainability requirements for software products during software development. Many are unaware of any applied process models or other software engineering artifacts to support sustainability in software engineering practices. This working paper intends to map practitioner-focused outcomes and academic research in sustainable software engineering. By adopting a practitioner's perspective, we categorize 11 types of software engineering artifacts. These artifacts represent tangible research outcomes for software practitioners and help with systematically analyzing academic publications on sustainable software engineering between 2001-2022. The analysis is based on a three-stage literature screening process, out of which three intermediate datasets are analyzed. The study provides valuable insights into the trends and patterns of research output, emphasizing the significance of artifacts and acknowledging their contribution to the field. The aim is to promote sustainable software engineering by considering and mapping the perspectives of both academics and practitioners. Furthermore, it opens up opportunities for future research and development.

Keywords: environmental sustainability, software engineering, artifacts, practitioners, frameworks, models, processes

1 Introduction

The ICT industry's increasing energy consumption is a major contributor to global emissions and is anticipated to worsen in the future ([Fo19]). The issue of environmental sustainability in software development has been acknowledged by the Information Systems (IS) academic community for many years. While 91% of software industry practitioners recognize the importance of sustainability, 92% are unable to identify sustainability requirements during the software development process, and 96% are unaware of any applied process models to support sustainability in Software Engineering practices ([No22]; [Ka21]). The problem of environmental sustainability of ICT has been recognized by the Information Systems academic community as a dynamic field of

¹ School of Business, Maynooth University, Maynooth, Co. Kildare, Ireland

² Innovation Value Institute, Maynooth University, Maynooth, Co. Kildare, Ireland

³ School of Business, Maynooth University, Maynooth, Co. Kildare, Ireland

innovation in research and in the industry [Wa10]. However, it remains a major challenge to consider environmental sustainability as an inherent characteristic of software products, which needs to be reflected in the software development process and life cycle ([HA15]). In the wider discussion on defining software sustainability, [Ko15] assert that sustainability should be regarded as a crucial component of software quality, necessitating its integration into the design and development phases. In contrast, [Na15] describe Sustainable Software Engineering as the approach of crafting software where its influences on sustainable development, both positive and negative, are continuously assessed during its entire lifecycle, with the intent of ongoing refinement. In line with our analysis, we aim to examine artifacts while also suggesting considering the software development lifecycle (SDLC).

This study is driven by the viewpoints of professionals in the software industry and is conducted through an analysis of scholarly literature. Scholars often prioritize academic rigor and theoretical contributions, while practitioners value practical insights and actionable outcomes. Researchers may be unaware of real-world problems, while practitioners may struggle or hesitate to adopt existing research [Ma21]. The study can benefit both academics and software industry practitioners by focusing on research outcomes important to the practitioners and acknowledging their perspectives. It aims to foster knowledge sharing and make research on sustainable software engineering hopefully more applicable in practice. We seek to create a practical reference for future research and collaboration in the field of sustainable software engineering through this study.

2 Methodology

We conduct the literature review that should adopt a practitioner's perspective by using a structured approach that involves designing a concept matrix ([WW02]). Furthermore, through this study, we have identified 11 distinct types of artifacts that revolve around outcomes such as tools, guidelines, frameworks, models, architectures, theories, metrics, concepts, methods, principles, and processes. The analysis is directed ([Ar11]) by its own research question (RQ):

- RQ: What are the trends and patterns in software engineering artifact production in academic articles on sustainable software engineering from 2001 to 2022, mapped with a practitioner's perspective?

To conduct this study, we utilize the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines ([Pa21]). While we did not fully adhere to them, we believe that our partial use of the guidelines ensures transparency in our literature review, making it replicable and easily accessible.

2.1 Definition of Scope

In the field of software engineering, one popular process model is the Software Development Lifecycle (SDLC), which provides a structured approach to creating software ([BF14]). In this study, we conduct a thorough analysis of literature related to sustainable software engineering, with a particular focus on how it applies to the SDLC. To conduct our research, we have created three datasets, corresponding to the screening stages, to analyze patterns and trends in scholarly articles, as shown in Table 1. Our goal is to identify relevant software engineering artifacts and ultimately take on the practitioner's viewpoint. The following steps are taken:

Identification: The initial steps involve conducting searches on four scientific databases (AIS, ACM, Scopus, and ScienceDirect) using the keywords "Environmental Sustainability," "SDLC," and "Software Engineering." This search process yields a total of 771 articles.

Broad Initial Screening (Dataset A): In this step, we perform a broad initial screening of the 771 articles based on their titles and abstracts. The objective is to exclude articles published before 2001 or after 2022, as well as those that are irrelevant to the research topic (such as covering different software engineering topics like cloud computing) or duplicates. Additionally, we include 40 articles through snowballing techniques [Wo14], which involve examining the references of relevant articles. Following this screening process, we created Broad Initial Screening (Dataset A), comprising 177 articles.

In the first assessment of titles and abstracts, called Software Engineering-Focused Screening (Dataset B), the articles from Broad Initial Screening (Dataset A) were assessed to identify those potentially relevant to software engineering practices. The emphasis is on selecting articles that align with the research objective. As a result, 72 articles were included in the Software Engineering-Focused Screening (Dataset B).

The second assessment of titles and abstracts, called Practitioner's Targeted Screening (Dataset C), involves further reviewing the articles from Software Engineering-Focused Screening (Dataset B). This step includes excluding opinion pieces, editorials, and secondary studies from the dataset. Only articles that meet the criteria are included in Practitioner's Targeted Screening (Dataset C), resulting in a total of 50 articles.

Dataset	Purpose	Screening process	Papers
A - Broad Initial Screening	Overview of research production in the relevant fields	Search: AIS, ACM, Scopus, ScienceDirect with keywords "Environmental Sustainability," "SDLC," "Software Engineering". Time period (2001-2022), reference to the environmental sustainability domain in the title or abstract, relevance, duplicates, snowballing.	177
B – Software Engineering Screening	Screening of papers relevant to software engineering process	Removing IT-department focused papers and papers focused on sustainability for organizations or institutions (rather than product engineering), through titles and abstracts review.	72
C – Practitioner's Targeted Screening	Screening of papers with relevant information for practitioners	Exclusion of papers least software practice-oriented (opinion pieces, editorials, secondary studies), through titles and abstracts review.	50

Tab. 1: Screening process and datasets

In the future, the research process will involve assessing the full-text articles of Practitioner's Targeted Screening (Dataset C) to further refine the dataset analysis and gain deeper insights into sustainable software engineering research. While this step may not be included in the current study, it is crucial for analyzing practical-focused artifacts.

2.2 Concept-Centric Structure

To gain a better understanding of the production of software engineering artifacts within the context of sustainable software engineering research, we conduct an analysis of the 11 artifact types identified. The identification, screening, and assessment of 177 articles as part of the literature review process directly led to the creation of these 11 types of software engineering artifacts in the context of sustainability.

The types of software engineering artifacts include tools that are software applications and utilities meant to support sustainability goals. Guidelines provide best practices and recommendations for developing environmentally sustainable software solutions. Frameworks provide structured approaches for integrating sustainability considerations into the software development process. Models are conceptual or mathematical representations that help in comprehending and evaluating the sustainability impact of software systems and designs. Architectures refer to fundamental concepts or properties

of a software system in our context to optimize the environmental impact. Theories encompass conceptual frameworks and theories that explain the relationship between sustainability and software engineering processes and products. Metrics provide quantifiable measures to assess the sustainability performance of software systems. Concepts represent abstract ideas and fundamental building blocks related to sustainability in software engineering. Methods refer to specific procedures and approaches that can be used to incorporate sustainability elements into the software development process. Principles comprise fundamental guidelines for developing sustainable software solutions. Processes are interrelated activities and systematic workflows that guide sustainable software engineering practices.

To examine the three screened datasets, we search through the titles and abstracts of all articles to find any instances of the 11 artifact types that were previously mentioned. Composing a concept matrix using these 11 software engineering artifact types will help in examining patterns, trends, and potential gaps in the investigated field of research.

3 Results

In this study, we generate three datasets (Broad Initial Screening as Dataset A, Software Engineering-Focused Screening as Dataset B, and Practitioner's Targeted Screening as Dataset C) by utilizing the article titles and abstracts. We conduct a time series analysis on published articles and software engineering artifacts, focusing on the period from 2001 to 2022 (see Figure 1 and Figure 2). Furthermore, we develop a concept matrix that comprises all artifact types and articles (examples are provided in Table 2 and Table 3). We searched the titles and abstracts of all articles in the three screened datasets to identify instances of the 11 artifact types. The outcomes of this analysis provide insights into the main research question, which asks for the trends and patterns in software engineering artifact production in academic articles on sustainable software engineering from 2001 to 2022, incorporating a practitioner's perspective.

3.1 Time Series Examination

The examination of the evolution in article publication in different datasets within the context of sustainable software engineering shows that dataset A has a gradual increase in articles over the years, with a peak of 17 articles in 2018. Datasets B and C show a similar trend, but with lower numbers.

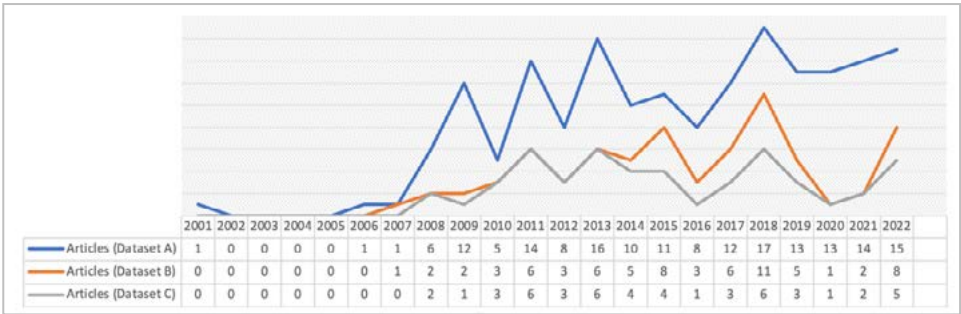


Fig. 1: Number of Articles Published in Different Datasets from 2001 to 2022

Dataset B has 11 articles in 2018 as its highest number, while dataset C reaches its peak in 2011, 2013 and 2018 with 6 articles.

By analyzing the development of artifact types in various datasets regarding sustainable software engineering (Figure 2), dataset A displayed a steady growth in artifact production. Notably, in 2018, there was a significant spike with a total of 36 artifacts produced. Dataset B and C also exhibit similar patterns, albeit with lower artifact counts.

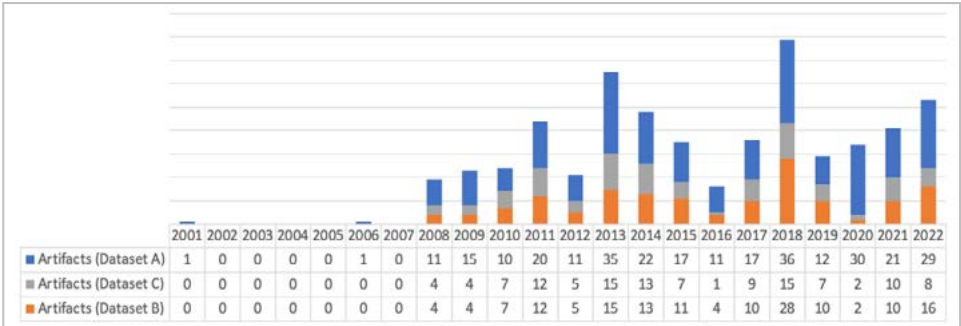


Fig. 2: Number of Artifacts Published in Different Datasets from 2001 to 2022

The highest number of artifacts in dataset B is observed in 2018 with 28 artifacts, while dataset C reaches its peaks in 2013 and 2018 with 15 artifacts. These findings suggest a growing output in sustainable software engineering artifact production over the years.

A comparison of article and artifact production over the three datasets from 2001 to 2022 shows that there is a noticeable correlation between artifact and article production within each dataset. As the number of articles increases, so does the production of artifacts. The analysis reveals several peaks and fluctuations in artifact production, often corresponding to specific years or periods within each dataset. In some instances, there

[illegible]

Tab. 2: Overview of the artifact types and articles within Dataset A for the years 2001 to 2010

We can compare the prevalence of different artifact types from earlier years (2001-2007) to more recent years (2008-2010) to identify emerging artifact types or changes in emphasis. Over 94% of the artifacts occurred in the later period.

Upon analyzing the data from 2017-2018 as shown in Table 3, we found that the dominant artifact types in dataset A are slightly more varied than those from the timeframe of 2001-2010.

Pub. Year	Author	Title	tool	guideline	framework	model	architecture	theory	metric	concept	method	principle	process
2017	Geidel, Stefan; Dharati, Pratyush; Fridgen, Gilbert;	The Sustainability Imperative in Information Systems Research						1,00			1,00		
2017	Sodenkamp, Mariya; Kudovskiy, Ilya, Huf.	Smart Meter Data Analytics for Enhanced Energy Efficiency in the Residential Sector											
2017	Nahant, Rohit; Teo, Thompson S. H.; Goh, Mark;	Do Shareholders Value Green Information Technology Announcements?				1,00		1,00					
2017	Gholami, Reza; Ravishanker, M. N.; Soudki, Patrick;	An exploratory study on Sustainable ICT Capability in the Travel and Tourism Industry: The Case of a Global Leisure System Provider			1,00								
2017	Sedera, Darshana; Lokuge, Sachithra	Multi-disciplinary Green IT Archival Analysis: A Pathway for Future Studies											
2017	Giret, Adriane; Trentosauk, Damien; Sainio, Miguel A.;	A holistic multi-agent methodology to design sustainable intelligent manufacturing control systems	1,00	1,00							1,00		
2017	García-Mirreles G.A	Environmental sustainability in software process improvement: A systematic mapping study											1,00
2017	Jagbbeer, Yasmeen; Hallstedt, Sophie I.; Larsson, Torbjörn	Exploration of Simulation-Driven Support Tools for Sustainable Product Development	1,00			1,00					1,00		
2017	Calero, Corac; Piatini, Mario	Puzzling out Software Sustainability								1,00			
2017	Winfree, Tom; Goldacre, Paul; Sherwell, Mohammedosssein;	Learning for Low Carbon Living: The Potential of Mobile Learning Applications for Built Environment Trades and Professionals in Australia			1,00						1,00		
2017	Michanik, Junya; Dewi, Renu; Rutherford, Matthew J.	GreenCS: An adaptive, energy-aware collection for green software development	1,00			1,00	1,00						
2017	Hilly, Lorenz; Bieser, Jan	Opportunities and Risks of Digitalization for Climate Protection in Switzerland								1,00			
2018	Pelaez-Silva, Maria Victoria; Gelfah, Ahmed; Porras, Jari	Infusing sustainability into software engineering education: Lessons learned from capstone projects											
2018	Mourão, Bruno C.; Kanta, Leila; do Carmo Machado, Ivani	Green and Sustainable Software Engineering - a Systematic Mapping Study	1,00						1,00	1,00	1,00		1,00
2018	Ondori-Fernandez N., Lago P.	Characterizing the contribution of quality requirements to software sustainability				1,00	1,00				1,00		
2018	Padón-Romero, J. David; Baldassarre, Maria Teresa;	Green IT Governance and Management based on ISO/IEC 15504			1,00	1,00							1,00
2018	Favi C., Germani M., Mendolini M., Marconi M.	Implementation of a software platform to support an eco-design methodology within a manufacturing firm	1,00			1,00				1,00	1,00		1,00
2018	García-Mirreles G.A., Villa-Martínez J.A.	Practices for addressing environmental sustainability through requirements processes			1,00						1,00		1,00
2018	Jagroop, Irit; Eini, Arjan van der;	The hunt for the gazelle: Architecture based energy profiling using stuns					1,00				1,00		
2018	Kien, Pui; Hilly, Lorenz M.	Sustainable software products—Towards assessment criteria for resource and energy efficiency				1,00							
2018	Feldner, Michael; Pahl, Detmar	Special Section: Automation and Analytics for Greener Software Engineering											
2018	Khoun, Fousse; Abenzouzen, S. Amihosssein	Understanding the impact of cloud patterns on performance and energy consumption		1,00			1,00						1,00
2018	Lettrache, Rhadik; El Beggar, Omar; Hamdani, Mohammed	Green Data Warehouse Design and Exploitation		1,00									
2018	Condon-Fernandez, Nelly; Lago, Patricia	Characterizing the contribution of quality requirements to software sustainability				1,00	1,00				1,00		
2018	García-Mirreles, Gabriel Alberto;	Interactions between environmental sustainability goals and software product quality: A mapping study										1,00	
2018	M. M. Ricardo Salvador; Fomes, Cristiano Hora de O	Sustainable and renewable energy supply chain: A system dynamics overview			1,00	1,00							
2018	Remy, Christian; Rates, Oliver; Dix, Alan	Evaluation Raymond Usability: Validating Sustainable HCI Research									1,00		
2018	Bieser, Jan G. T.; Hilly, Lorenz M.	An Approach to Assess Indirect Environmental Effects of Digitalization Based on a Time-Use Perspective				1,00					1,00		
2018	Jones, Nicola	How to stop data centres from gobbling up the world's electricity											

Tab. 3: Overview of the artifact types and articles within Dataset A for the years 2017 to 2018

The list of above-average types consists of "Tool," "Concept," "Method," "Model," "Architecture," "Framework," and "Process," but the type "Principle" is not included.

For this preliminary analysis we have purposely chosen to focus on the time periods of 2001-2010 and 2017-18 since we observed their unique significance within the dataset

and a change in pattern. The period of 2001-2010 allows us to examine the foundational stages of research in sustainable software engineering. We can gain insights into the initial growth of artifact occurrences during this time. On the other hand, the 2017-18 period represents a more contemporary era characterized by a substantial peak in artifact production across all datasets. By focusing on these two distinctive time periods, we can showcase the field's evolution from its nascent stages to its current state of heightened research activity.

The complete concept matrix spanning from 2001-2022 for dataset A, highlights further interesting patterns when analyzing the various types of artifacts included in the dataset. The top three artifact types with the highest number of occurrences are "Model" (53), "Process" (42), "Method" (41) and "Framework" (41). These types show varying trends over time, with the "Model" type exhibiting an overall increasing trend. On the other hand, the bottom three artifact types with the lowest number of occurrences are "Architecture" (9), "Guideline" (8), "Metric" (13), "Principle" (14) and "Theory" (14). These categories demonstrate relatively stable trends with sporadic peaks and dips. Overall, the findings highlight the prominence of models, processes, and frameworks in research, while architecture, guidelines, metrics, and theories have received comparatively less coverage. These findings provide some initial insights into the distribution of research efforts across different types of artifacts for sustainable software engineering, highlighting areas of focus as well as potential gaps in the research landscape.

4 Discussion

4.1 Time series examination - Initial Presumptions of Practitioners

As in some instances, there appears to be a lag between the production of articles and the subsequent artifact/article production ratio, there are possibly several reasons for the observed lag time in artifact/article production ratio compared to articles. Firstly, the process of converting research findings into tangible artifacts could often involve additional stages such as testing, and evaluation. These stages require time and resources, possibly leading to a delay in the production of artifacts. From a practitioner's perspective, practitioners may worry that the lag in artifact production hinders their ability to stay updated with the latest advancements. They might feel that by the time artifacts are developed and made available, newer technologies have already emerged, making the artifacts less relevant or outdated. They might argue that the perceived lengthy process of transforming research findings into artifacts delays the incorporation of practical solutions into real-world software development scenarios and they may feel that the artifacts produced do not adequately address their specific industry challenges and requirements.

4.2 Concept Matrix - Initial Presumptions of Practitioners

Based on the data provided in the concept matrix 2001-2022 for dataset A there could be several concerns from the perspective of practitioners: With only 8 contributions for the "Guideline" type, practitioners may express concerns about the limited availability of guidelines. They might argue that having a small number of guidelines create a barrier to their ability to access best practices within their industry.

The relatively low "Theory" type (14 contributions) as well as the "Architecture" type (9 contributions), might lead practitioners to concerns about the limited availability of theoretical research relevant to their field. They may argue that this small number of contributions restricts their access to the latest theoretical advancements and architectures that could positively inform their system designs.

In summary, we formulate the hypothesis that software industry practitioners may feel that the lengthy process of turning research findings into practical solutions for real-world software development delays their incorporation. In the field of sustainable software engineering, they may also feel impacted by the lack of guidelines, theories, and architectures emerging from the research realm, which limits their access to these resources. We expect to test the hypothesis in future research, and practitioner interviews may be included to provide additional insights. This will not only evaluate the variances in adopting these artifacts but also improve our current study's assumptions.

5 Conclusion

This paper connects practitioner's-focused outcomes and academic research in sustainable software engineering. By adopting a practitioner's perspective, we categorize 11 types of software engineering artifacts. The analysis involves comparing the production of articles and artifacts across three datasets spanning from 2001 to 2022. The correlation between artifact and article production becomes evident, as an increase in articles is paralleled by an increase in artifact creation within each dataset. Throughout this period, there are distinct peaks and fluctuations in artifact production, often aligning with specific years or intervals within the datasets. Notably, at times, there appears to be a delay between article production and the subsequent generation of artifacts. Among the various artifact types, the top ones with the highest occurrence counts are "Model" (53), "Process" (42), "Method" (41) and "Framework" (41). While each type exhibits distinct temporal trends, the "Model" category shows a consistent upward trajectory. Conversely, the least common artifact types are "Guideline" (8), "Architecture" (9), "Metric" (13), "Principle" (14) and "Theory" (14), which demonstrate relatively stable patterns with occasional spikes and declines.

Overall, these findings underscore the prevalence of models, processes, methods and frameworks in research, while guidelines, architecture, metrics, principles and theories

receive comparatively less attention. This insight offers a preliminary understanding of how research efforts are distributed among different artifact types in sustainable software engineering. The analysis allows us to formulate the hypothesis that software industry practitioners might challenge the lag in artifact/article production. Furthermore, the findings guide us to hypothesize that practitioners may express concerns over the restricted availability of guidelines, theoretical research, and architectural insights. This may lead us to infer that the delay in translating research into practical artifacts and the relatively low numbers of artifacts such as guidelines, theories, metrics, principles, and architectures could potentially present challenges for practitioners engaged in sustainable software engineering endeavors.

6 Further Research Direction

To continue our study, we will be utilizing the Design Science Research methodology (DSRM). This methodology is known for its systematic approach, emphasis on innovative artifact creation, rigorous testing, and established track record in the field ([He04]) By following the clear steps for iterative design enhancement and evaluation, with a focus on creating practical solutions and conducting rigorous testing, experimentation ([Os14]) and addressing real-world problems, we believe it will be a good fit with the objectives of this study. As we move forward with our research, we plan to assess suggested artifacts in relation to their practical relevance. We will further expand the dataset analysis to determine if there are any indications of the practical usage and real-world evaluation of the exposed artifacts. We plan to use the current dataset in our expanded research in the future and therefore don't intend to release the dataset publicly until then. To enhance our understanding, we will directly seek feedback from software industry practitioners on the different artifact types, aiming to evaluate their potential adoption and usage. This valuable input will further inform and refine the practitioners' presumptions suggested in the study. The findings of this study are being used in ongoing research on the topic of Environmental Sustainability driven by Software Engineering with a focus on artifact design.

Bibliography

- [Ar11] R. Armstrong, B. J. Hall, J. Doyle, and E. Waters, "'Scoping the scope' of a cochrane review," *Journal of Public Health*, vol. 33, no. 1, pp. 147–150, Mar. 2011, doi: 10.1093/pubmed/fdr015.
- [BF14] P. Bourque and R. E. Fairley, *SWEBOK: guide to the software engineering body of knowledge*. IEEE Computer Society, 2014.
- [Fo19] A. Fonseca, R. Kazman, and P. Lago, "A Manifesto for Energy-Aware Software," *IEEE Software*, vol. 36, no. 6, pp. 79–82, Nov. 2019, doi: 10.1109/MS.2019.2924498.
- [He04] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information

- Systems Research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.
- [HA15] L. M. Hilty and B. Aebischer, “ICT for Sustainability: An Emerging Research Field,” in *ICT Innovations for Sustainability*, L. M. Hilty and B. Aebischer, Eds., in *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2015, pp. 3–36. doi: 10.1007/978-3-319-09228-7_1.
- [Ka21] L. Karita, I. Machado, L. Soares, L. Martins, and B. Mourão, “Software industry awareness on sustainable software engineering: a Brazilian perspective,” *Journal of Software Engineering Research and Development*, vol. 9, p. 2:1, Feb. 2021, doi: 10.5753/jserd.2021.742.
- [Ko15] S. A. Koçak, G. I. Alptekin, and A. B. Bener, “Integrating Environmental Sustainability in Software Product Quality.,” in *RE4SuSy@ RE*, 2015, pp. 17–24.
- [Ma21] D. Marijan, T. Zimmermann, M. Ham, and B. Selic, “Bridging Software Engineering Research and Industrial Practice,” *SIGSOFT Softw. Eng. Notes*, vol. 46, no. 1, pp. 30–32, Feb. 2021, doi: 10.1145/3437479.3437488.
- [Na15] S. Naumann, E. Kern, M. Dick, and T. Johann, “Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects,” in *Advances in Intelligent Systems and Computing*, 2015. doi: 10.1007/978-3-319-09228-7_11.
- [No23] H. Noman, N. A. Mahoto, S. Bhatti, H. A. Abosag, M. S. A. Reshan, and A. Shaikh, “An Exploratory Study of Software Sustainability at Early Stages of Software Development,” *MDPI AG*, 2022. Accessed: May 28, 2023. [Online]. Available: <https://doaj.org/article/d0d1fe143dc34c59a6c4ade7d89dcbf4>
- [Os23] L. Ostrowski, “Design and evaluation of activities and reference model for the meta-design phase of design science - demonstrated on business process model artefacts,” doctoral, Dublin City University, 2014. Accessed: May 28, 2023. [Online]. Available: <https://doras.dcu.ie/19713/>
- [Pa20] M. J. Page et al., “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” *BMJ*, vol. 372, p. n71, Mar. 2021, doi: 10.1136/bmj.n71.
- [Wa10] R. T. Watson, M.-C. Boudreau, and A. J. Chen, “Information Systems and Environmentally Sustainable Development: Energy Informatics and New Directions for the IS Community,” *MIS Quarterly*, vol. 34, no. 1, pp. 23–38, 2010, doi: 10.2307/20721413.
- [Wo14] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, in *EASE '14*. New York, NY, USA: Association for Computing Machinery, Mai 2014, pp. 1–10. doi: 10.1145/2601248.2601268.
- [WW02] J. Webster and R. T. Watson, “Analyzing the Past to Prepare for the Future: Writing a Literature Review,” *MIS Quarterly*, vol. 26, no. 2, pp. xiii–xxiii, 2002.