Social Sustainability Approaches for Software Development: A Systematic Literature Review

Ana Carolina Moises de Souza, Daniela Soares Cruzes, Letizia Jaccheri, and John Krogstie

Norwegian University of Science and Technology, Trondheim 7491, Norway
{ana.c.m.de.souza,daniela.s.cruzes,letizia.jaccheri,john.krogstie}@ntnu.no

Abstract. Social aspects in software sustainability refer to the impact of the software on the broader social and societal context. These aspects involve considerations such as accessibility, equity, inclusion, diversity, ethical and human values. While achieving software sustainability requires developers to embrace approaches that support the three dimensions of sustainability, there remains a lack of concrete approaches to address social aspects during software development. This literature review aims to facilitate the integration of social aspects into the software development process by identifying approaches related to social sustainability in software engineering. We extracted and analyzed data from 19 studies through thematic syntheses. The results of our analysis provide a list of recommended tools and practices to support social aspects and attain software sustainability goals. By incorporating these approaches into software development, we ensure that the software is not only technically sustainable but also socially responsible from a human perspective.

Keywords: Social Sustainability · Social Aspects · Agile Software Development · Software Sustainability · Sustainable Software

1 Introduction

Sustainability is an important area of concern in modern software engineering due to the significant environmental and social consequences resulting from the increasing use of technology. Since 1987, sustainable development has been under discussion, and much has been done in society to preserve the same resources we have today for future generations [5]. This definition of sustainability encompasses three interrelated dimensions: economic, social, and environmental. Littig and Grießler [11] argue that all three dimensions of sustainability should be equally considered: "Human needs cannot be sufficiently met just by providing an ecologically stable and healthy environment, but that - if a society is indeed committed to sustainability - the equally legitimate social and cultural needs ought to be taken care of as well. Economic, social, and cultural conditions, efforts, and values are deemed to be resources that also need to be preserved for future generations." In software engineering and other sectors, there has been a focus on the environmental dimension of sustainability, evidenced by [16] [13].

However, there is a growing need to investigate the social dimension, combined with the individual and human dimensions, to achieve sustainability in software engineering [20] [23] [6].

Social sustainability can be achieved when software is designed to promote social aspects of the community, such as equality, diversity, community building, and a sense of belonging [4] [20] [32]. These aspects also relate to individuals' aspirations for an equal society [4]. But the question still remains: how can we develop software to minimize negative societal impacts? To address this question, we aim to review the existing literature on software development and social sustainability to identify tools, approaches, or methods that software developers can use to integrate social sustainability into their software development practices.

Moreover, we are motivated by the assumption that some organizations struggle to apply social sustainability principles effectively in the context of software development [6]. With this context in mind, we have followed SLR guidelines to identify relevant studies on social sustainability approaches in software development. After executing the SLR guidelines [18], we selected 19 out of 5858 papers from the search results. We extracted information, such as the type of study, sustainability dimensions, and empirical validation, from the selected papers. After the data extraction, we conducted a data synthesis using thematic analysis. The approaches contributing to social sustainability were categorized into social aspects and goals. Our contribution demonstrates how these approaches, tools, and practices can help integrate social sustainability concerns into software development. This paper is organized as follows: Section 2 introduces our theoretical framework and related work in software sustainability. Section 3 outlines our systematic literature review protocol. Section 4 presents the SLR findings and addresses each question. Section 5 discusses social aspects of software engineering, along with limitations, opportunities, and future work. Finally, Section 6 summarizes our study.

2 Background

This section provides an overview of the scoping, mapping, or systematic literature reviews we selected during the exploratory phase and through our search string as related work. We will also describe the theoretical framework that supported our higher themes and guided our synthesis analysis.

2.1 Reviews in Software Sustainability

While numerous studies have been conducted in the field of sustainable software engineering, there is a relative lack of focus on social sustainability and the software development process. Secondary studies collectively emphasize the importance of considering both product and process sustainability, highlight the neglect of social aspects in software sustainability, and call for developing tools and frameworks that address social dimensions alongside environmental and economic considerations. In Table 1, we summarized each secondary paper's

outcome and the gap they identified in their study. These identified gaps formed the basis for defining the research questions in our systematic literature review.

Table 1: Recent Reviews on Software Sustainability

Authors	Outcome	Identified Gap in the Studies
McGuire, S. et al. [21]	A multisystemic nature of sustainability is suggested considering micro and macro levels of each sustainability dimension.	Lack of empirical validation and experiments in social sustainability.
Swacha, J. [30]	Reference and evaluation models used to address software sustainability.	Lack of verification or validation of the models.
Gustavsson, J. et al. [14]	Sustainability should be addressed in a holistic manner and should consider social aspects during software development.	Need for tools to assist in social sustainability adoption.
Khalifeh, A. et al. [17]	A framework for incorporating economic considerations, environmental concerns, and social responsibilities into software product projects.	Incorporation of social sustainability into software projects.
Alharthi, A.D. et al. [3]	Meta-requirements for addressing sustainability in e-Learning systems.	Diversify the social sustainability aspects of a product.

2.2 Theoretical Framework on Social Sustainability

Interdisciplinary research related to society, human values, and sustainability is a traditional combination in software engineering, as it reflects the growing recognition of the need for a more holistic approach to software development and its impact on the world [24] [11]. Incorporating perspectives from fields such as sociology, psychology, and philosophy is essential for understanding the societal impact of software systems and the ethical implications of their development and use [22] [1]. Figure 1 shows the perspectives of society towards social sustainability inspired by [22] and adapted from Ajmal et al. [1]. The External Societal Perspectives contribute to the definition of social sustainability.

Fig. 1: Theoretical Framework adapted from [1]

Social Growth refers to an individual's ability to thrive in society through community engagement, autonomy in decision-making, and contributing to the well-being of others [1]. In our study, we incorporate human and individual dimensions within the concept of Social Growth, which we call "Personal Growth."

Social Development involves building a community that promotes the well-being of its members. It is a collective responsibility shared by government, companies, agencies, and individuals [1]. Social Development encompasses creating

supportive policies and programs, providing resources for individual social development, and actively participating in community initiatives for sustainable development [27].

Social Justice is based on the principle that everyone should have equal access to rights and opportunities, regardless of their background [1]. Achieving social justice requires addressing aspects such as equality, diversity, trust, fairness, transparency, and security. The Covid-19 pandemic has emphasized the need for innovative approaches in various sectors to safeguard sustainability and human security [31].

3 Review method

The Systematic Literature Review (SLR) [18] was performed to investigate studies addressing social aspects during software development. To achieve this goal, we addressed three research questions that range from broad to specific, aligned with the gaps presented in table 1:

- RQ1 - What are the characteristics of the literature addressing social sustainability in software development?
- RQ2 - How do the proposed support tools address social aspects in software engineering?
- RQ3 - How can social sustainability be concretely addressed in software development practice?

3.1 Studies Selection

In this study, the search string focused on sustainability and its relationship with software development. The term 'sustainability' was intentionally chosen to avoid narrowing or refining the dimensions, allowing for the selection of various dimensions. The refinement of the social dimension occurred after a thorough review of the literature. It is worth mentioning that some authors introduced new dimensions, such as 'human', which is related to the social aspect. Therefore, specifying social in the search string would remove the papers that covered related dimensions. To ensure a comprehensive range of results, we tested multiple versions of the search string and refined it through several rounds of testing. Eventually, we settled on the current search string as the most appropriate and relevant for our study:

((Abstract="sustainable software" OR Abstract=sustainability)
AND
(Any field=software OR Any field=mobile))

To ensure the validity and reliability of our search results, we conducted the search using the Web of Science database, which covers a wide range of well-known scientific databases. Our search strategy allowed us to collect relevant data to address the research questions of our study, resulting in a total of 5858

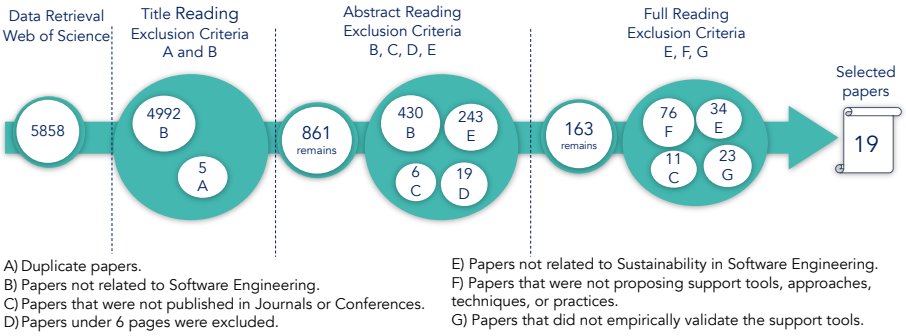

Fig. 2: Filtering steps, exclusion criteria and results numbering.

papers that needed to be screened. The filtering process involved three main activities, as shown in Figure 2. Firstly, we applied a search string and conducted a title reading to exclude duplicates and papers unrelated to software engineering based on the defined exclusion criteria. Secondly, we performed an abstract reading to identify papers that addressed social aspects of software engineering and also applied exclusion criteria, such as non-peer-reviewed papers, papers under six pages, and those not related to software engineering. The final activity involved a full reading of the selected papers to refine our results and identify those that proposed activities, practices, guidelines, frameworks, or models to address social sustainability in software development. After completing the selection phase, we proceeded to synthesize and assess the quality of the studies from the chosen papers.

3.2 Data Extraction, Quality Assessment and Data Synthesis

A protocol was developed to identify the answers to the questions and better understand the chosen studies. Table 2 shows the extracted information proposed in the studies and discussed in the results section 4. Each study was assessed based on the criteria derived from two sources: Dybå and Dingsøyr [10] and Hernandez et al. [15]. To help assess the quality of the studies, we created a set of questions. The answer for each question was either Yes (1) or No (0). The sum of the answers provides the quality score of the studies (see Figure 3). This score helps identify a research's relevance, rigor, and credibility. Thematic synthesis was performed to synthesize the findings. In this method, researchers typically read through the data and identify recurring ideas, concepts, or themes [8]. These themes are then grouped together into higher-order categories or themes that capture the overall essence of the data. Four steps were performed during data synthesis in the MAXQDA tool (https://www.maxqda.com/literature-review): 1) Identified and coded the keywords related to social aspects (e.g. equality, diversity, etc...) explicitly in the text; 2) Selected and synthesized from the text the relevant practices that connected with the social aspects; 3) Categorized the practices-related codes into software life cycle phases or organizational practices;

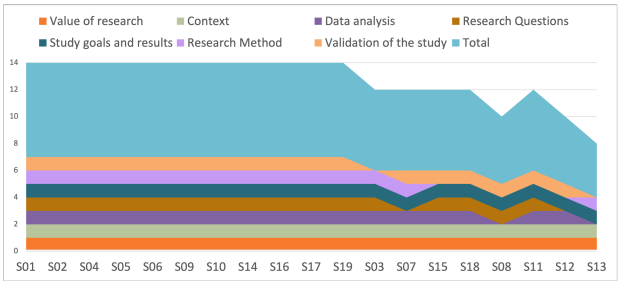

Fig. 3: Quality Assessment Results.

and 4) Grouped these codes into higher themes (social aspects) based on the theoretical framework. To better visualize the connections between social aspects, social goals, software life cycle phases, and practices, we created three maps for each social dimension, which are available in this research package.

4 Results

In this study, we identified practices in 19 papers that proposed support tools to address sustainability in software. The studies are mostly published at conferences resulting in 12 papers. Only 7 papers were published in Journal. Further results are discussed in the following sections.

4.1 RQ 1 - Social sustainability studies in software development

To answer this question, we will go through the data extraction protocol and explain how we extracted and summarized the findings of studies that address social sustainability. The extracted data and the corresponding number of findings are presented in Table 2. The **Business Domain** information was extracted from the papers to determine the applicability of the approaches proposed in the studies based on the specific context. 15 out of 19 papers did not specify a particular business domain and conducted their studies across various domains. The **Software Life Cycle** was collected to identify which phases of the software development life cycle were addressed by studies. The studies primarily focused on the Software Requirements (10 out of 19), with no coverage of other areas such as construction, maintenance, and testing. Within the software requirements domain, there is a clear focus on sustainability as it pertains to the purpose of the software, its functionalities, and the associated business constraints. The incorporation of social sustainability approaches helps in the development of software that takes social aspects into consideration right from the requirements stage. The **Study Type** was extracted to understand the research method and its empirical validation. Various study types were identified, in which case studies research method was the most popular (3 out of 19). A case study involves analysis of a particular situation to gain a better understanding of complex phenomena. On the **Sustainability Dimensions**, we observed that

Table 2: Results of the extracted data from the studies.

Data Extraction	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	#
Business Domain																				
Not specified	•	•	•	•	•	•	•		•	•	•	•	•	•		•			•	15
Health															•		•			2
Energy																		•		1
OSS								•												1
Phase of the Software Life Cycle (SLC)																				
Requirements	•		•	•					•	•	•	•	•		•				•	10
Overall SLC								•						•		•	•	•		5
Design					•	•														2
Not specified		•					•													2
Study Type																				
Case study				•							•								•	3
Documental														•	•					2
Action Research						•											•			2
Delphi							•			•										2
Design Science								•	•											2
Survey	•				•															2
Pilot Study			•										•							2
Experience Report																		•		1
Experiment																•				1
Grounded theory		•																		1
Mixed Methods												•								1
Sustainability Dimensions																				
Social	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	18
Environmental		•	•	•	•	•		•	•	•			•	•	•			•	•	13
Technical	•	•	•	•	•	•		•	•	•			•	•	•				•	13
Economic		•	•	•	•	•		•	•	•			•	•				•	•	12
Individual									•	•			•		•				•	5
Human														•						1
Purpose													•							1
Legal													•							1
Integrative													•							1
Design-aesthetics													•							1
Types of support tool																				
Models		•	•	•	•	•										•		•	•	8
Awareness									•	•			•							3
Maps								•									•			2
Catalog	•														•					2
Value Pattern											•	•								2
Reports							•							•						2
Empirical validation																				
Industry		•		•	•	•	•	•			•	•		•	•	•	•	•	•	14
Pilot validated			•										•							2
Academic	•								•	•										3

studies on software engineering and sustainability often encompass multiple dimensions of sustainability. 5 out of 19 studies exclusively investigated the social dimension. The category of **Support Tool Type** was created to group studies that present tools with similar or related functionalities. The categorization was based on how the tools were described in the studies. When a clear definition was not provided, we categorized the tools ourselves. In this category, 8 out of 19 papers were classified as models. More detail about the tools description and related studies is explained in the results section 4 for RQ2. The **Empirical Validation** was performed in the industrial setting in 14 of the studies, providing valuable real-world perspectives from developers and companies.

4.2 RQ2 - Proposed tools to address social sustainability

The tools identified in this SLR consist of various resources such as practices, questions, checklists, visualization tools, diagrams, and frameworks that assist developers in understanding, implementing, discussing, or testing sustainability dimensions. We grouped the tools into six main types of tools (Table 3): awareness, catalog, maps, models, values patterns, and reports.

Models simplify complex real-world concepts and can be used for structured decision-making and implementing new concepts within existing processes. The tools categorized as "models" describe guidelines for considering sustainability in the software development phases.

- **How**: Models were divided into organizational models S02, S018 and product-oriented model S03, S04, S05, S06, S16 S19. Organizational models analyze the overall influence of the organization on sustainability, including how system purchases align with sustainability and business strategy. Product-oriented models focus on the product's impact on sustainability, addressing requirements, development stages, and involving stakeholders. Sustainability is considered a quality aspect of the software in some of these models, mapping it to existing software engineering quality attributes.
- **Similarities**: One commonality is that all the tools categorized into models intentionally addressed social aspects in their proposals.
- **Differences**: S02 outlines the dimensions, practices, and stakeholders related to social sustainability in software product lines. S018 is designed to support decisions in procurement systems. S03 focuses on defining a software sustainability model for software-intensive systems and supporting decision-making tasks such as prioritizing requirements and analyzing trade-offs. S04, S05, S06, propose sustainability models for quality requirements prioritization and software sustainability assessment. It covers 17 qualities from ISO/IEC 25010:2011. S16 proposes a model for a rapid prototyping solution that involves older adults as participants in the process. S19 maps stakeholder profiles onto sustainability dimensions.

In table 3, we summarized the remaining support tools due to space limitations in this paper. These tools' focus and key features were extracted from the selected studies.

Table 3: Summary of Support Tools for Social Sustainability

Type	**Focus...**	**Key Features**	Studies
Awareness	On creating awareness towards the social aspects of a software product.	Can be utilized during the conceptual phase and for continuous review of software requirements.	S09, S10, S13
Catalog	On a pre-defined and ready-to-use catalog for software requirements addressing sustainability dimensions.	Useful for eliciting software requirements and reviewing business needs.	S01 S15
Maps	On mapping existing software engineering practices into sustainability and human values.	Map software engineering best practices to integrate sustainability goals into a product. Map software requirements into human values.	S08 S17
Models	On providing a structure to simplify complex concepts.	Approaches to achieving sustainability at the organizational level or product level.	S02, S03, S04, S05, S06, S16, S19
Reports	On giving guidance on how and what information to report	Sustainability indicators measure initiatives and are reported in Corporate Social Responsibilities reports.	S07 S14
Value Patterns	On describing common themes in language related to social aspects. Patterns consist of values, activities, and indicators.	Help software developers understand patterns and relationships of social aspects.	S11 S12

While we have identified promising tools for addressing social sustainability in software development in this SLR, three tools stand out due to their significant practical contributions. One such tool is the Sustainability Quality Model S04, S05, and S06. These studies incorporated feedback from sustainability experts and software developers, enhancing the model's credibility. Another noteworthy tool is the Sustainability Awareness Framework S09 and S010, which falls under the awareness support tool type. The selected studies have demonstrated the validation and applicability of this framework in the industry. The Sustainability Catalog Webtool S01 is a tool that facilitates the visualization and relationship between different dimensions of sustainability, thus categorized as a catalog. Therefore the contribution of RQ2 is to recommend support tools that can assist developers and organizations in integrating social sustainability practices into their software development process.

4.3 RQ3 - Social sustainability in software engineering

The investigation for this question was conducted to identify the software development practices that impact social sustainability by examining primary studies, coding information for social connections, and aligning with the theoretical framework described in section 2. It was identified that social sustainability in software engineering focuses on the social aspects related to development, justice, and personal growth. These social aspects encompass social goals such as equality, human rights, security, etc. (see figure 4). For each goal, we summarized a practice identified in the studies. These synthesized practices are related to hu-

man factors that serve as motivators to address social sustainability in software development.

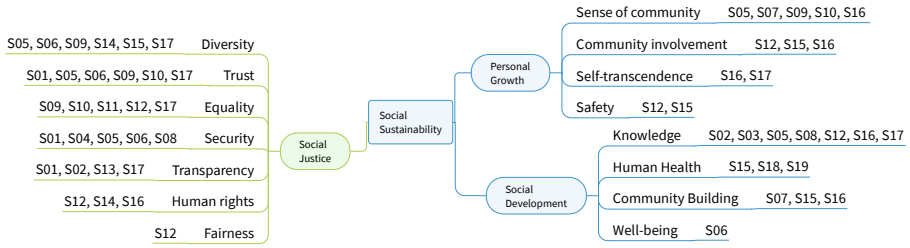

Fig. 4: Social Goals identified in the studies based on the Theoretical Framework.

Social Justice refers to social goals of diversity, trust, equality, security, transparency, human rights and fairness.

Diversity is important for software products and agile teams [19][2]. Practices like designing accessible software (S05 S06) and considering cultural diversity (S15) promote inclusion. In S14, organizations could address the gender gap in software engineering, exploring reasons and proposing initiatives to increase women's participation in computer science. Evaluating software's impact on perception and discriminatory behavior, S09 and involving diverse stakeholders help mitigate negative effects on diversity S17.

Trust is crucial in society, and designing software that instills confidence and meets stakeholders' expectations promotes trust S05, S06. Assessing the software's impact on trust and mitigating potential issues are important practices S09 S10. Security requirements can also enhance trust S01. S17 addresses identifying security options that users can rely on to establish trust.

Equality ensures equal opportunities for individuals to maximize their potential in life and talents [12]. In Information Communication Technology (ICT), we can promote equal access to digital services, infrastructure, hardware, and software. As such, software should provide access to its resources regardless of personal characteristics or beliefs. S05 proposes that during the software requirements, identifying inclusive requirements and implementing accessibility features contributes to addressing equality. In S09 S10, the software's impact on bias and inequality should be assessed throughout development. S11 S12 mentioned that Stakeholders' profiles should be considered to ensure equal opportunities. S17 identifies that using a common language promotes equal participation and collaboration among project stakeholders. Respecting diverse attributes, identities, and capabilities contributes to an equal society.

Software *Security* encompasses technical and social aspects of sustainability. One practice recommended in S08 is to design software that enables quick and easy security updates. The social dimension of security involves preserving user information, identity, privacy, and integrity, with confidentiality, authenticity, accountability, and integrity emphasized in S01, S04, S05 and S06. Considering

these aspects during software requirements ensures they are prioritized alongside other requirements.

Transparency empowers end-users, builds trust, and enables informed decision-making. In S01 and S02, designing software that provides timely feedback and guides users might enhance transparency. Developing software following regulatory standards with legal backing also increases transparency S13. Transparency can be achieved by ensuring data ownership, transparent data handling, and storage and adopting values-conscious practices for personal data S17.

Human rights, regardless of race or religion, should be respected in software design S16. Identifying and addressing requirements that uphold human rights prevent violations S12. In S14, organizational actions, such as non-discrimination policies, contribute to a safe workplace.

Fairness means treating individuals impartially and equitably, without discrimination or favoritism based on arbitrary factors [25]. In software requirements engineering, a practice in S12 involves identifying software needs through a fair selection of stakeholders without favoritism.

Social Development refers to social goals of knowledge, human health, community building, and well-being.

Knowledge is promoted by access to educational systems. S16 discussed ICT Literacy, which enhances people's knowledge. It helps people overcome their fear of technology and effectively use the software. In S08, participatory design aligns with the goal of designing easy-to-learn and user-friendly software. In the studies S02, S05 and S17, organizations should align technical and social skills to create a shared purpose for software under development. S12 describes the understanding of how technology is accepted and can help developers create accessible functionalities. S03 highlighted a practice to identify stakeholders knowledgeable about sustainability during the software requirements.

Human health refers to the condition of an individual's physical, mental, and social state. S15 indicated designing solutions that promote healthy lifestyles and offer predefined sustainability-related requirements applicable across various domains. According to S18, software should prioritize end-user health and avoid causing harm. S19 provides a stakeholder list for sustainability, including advocates for human health.

Community Building is primarily addressed in open-source communities. In S07, reporting the quality of the code and offering training to the community can be adopted by organizations. S15 proposed implementing solutions that promote social solidarity and are related to community building. S16 recommended promoting inter-generational interaction within software development teams to address bias and foster innovative solutions.

Well-being refers to optimal physical, mental, and emotional health and happiness. S06 revealed that software helps users achieve their goals and can reduce stress and frustration. Adopting this practice in projects from a software design perspective can address wellness concerns.

Personal Growth: refers to social goals of a sense of community, community involvement, self-transcendence, and safety.

Sense of Community refers to the individual's concern regarding the potential negative consequences of software on society. In S5, S6 one of the quality attributes proposed is freedom from risk to mitigate the negative impacts on the environment that can also impact the social dimension. In S07, users can rank their experiences when using a software or service. S09 S10 introduced a question to identify how the system affects a person's sense of belonging. S016 proposed inviting potential users to co-design solutions.

Community Involvement refers to how the software can motivate individuals to participate in the community. S12 discussed the indicators to measure community participation, such as social interaction activities, volunteer work, and decision-making, which help establish social ties and networks. S15 emphasized the importance of social interaction between users and developers as a practice that facilitates personal growth. S16 proposed the integration of individuals into agile processes, enabling their participation throughout the entire lifecycle.

Self-transcendence in human values theory refers to surpassing individual needs and desires to engage in actions that benefit others and the larger community [26]. In S16, practicing self-transcendence can involve assisting older adults in becoming familiar with technology and encouraging their feedback on solutions. Self-transcendence can also be linked to collaborative work practices. S17 recognized the importance of treating stakeholders as peers and fostering non-hierarchical communication to promote harmonious interactions between developers and stakeholders.

Safety enables personal growth by focusing on prevention. From a software perspective, it entails creating requirements that protect end-users against crimes such as cyberattacks. S12 identified secure features to protect user property from crimes. S15 identified features that ensure user safety by respecting their privacy and preventing data breaches.

5 Discussion

This review emphasized the underexplored nature of social sustainability in software engineering and the lack of clear definitions and boundaries in the field [17] [14] [3]. The need for a consensus on the definition of social sustainability and the importance of establishing a common language that resonates with software developers are emphasized in the selected studies. To better handle this definition, we used a theoretical framework to understand social sustainability through the lens of social science.

Throughout the review, we noticed that the studies either focused on one specific social goal or did not investigate a selected one. Therefore, a more holistic approach to investigating social sustainability in software development is needed. By doing so, we can avoid situations where these goals may not be explicitly linked to social sustainability [7]. Moreover, this review identified social aspects and goals that trigger a topic to be investigated: corporate digital responsibility [7].

To gain insight into our study, we examined its implications through two lenses: socially sustainable software and societal issues and threats. While the software is designed to benefit society by solving real-world problems and facilitating tasks, people involved in software development often overlook the potential negative consequences of this software. Therefore, designing socially sustainable software requires knowledge of societal issues, threats, social aspects, and goals, thus encompassing social sustainability. A recent societal issue observed is: the negative impacts of Generative Pre-processed Transformers (GPTs) and the wider field of Artificial Intelligence (AI) [9]. Although the main focus of this discussion is not artificial intelligence per se, we acknowledge that when developing AI tools is necessary to adhere to the software engineering process. Best practices of software engineering cannot be dissociated from AI solutions development. Eventually, the social sustainability practices cannot be dissociated either.

In the search for concrete practices for the software development team, we highlight the following types of support tools as ready to be used: awareness, model and catalog. These support tools offer practical adoption, such as detailed guidelines, online resources, and plugins. For instance, the Sustainability Awareness Framework (S09, S10), classified as an awareness tool, provides online material with questionnaires templates and workshop facilitation guidance [29]. Anyhow, it is essential to acknowledge a general lack of empirical evidence regarding the impact of adopting these tools on the final product and the long-term effects of software on society. Likewise, continuous software development brings to light additional challenges inherent to modern non-functional aspects of software. A key aspect is prioritizing sustainability throughout the software development phases [28].

6 Conclusion

As we move forward to address social issues related to the impact of software on human life, we increasingly recognize the importance of practical and scientific research on socially sustainable software. As software and society evolve over time, what will be the impact of this software if it maintains the same functionalities without paying attention to human factors and societal behavior?

This research aimed to discover relevant approaches to address social issues during software development. The studies in this review have discussed social issues and threats in their papers and suggested addressing them in software development. We have provided a list of recommended support tools and practices to allow software developers to explore the societal issues and threats of the software under construction.

The practices and tools identified in this review require validation through experiments or focus groups with software developers to ensure the identified practices' practicality and effectiveness. Further validation and testing of the identified support tools provide insights into their efficacy, enabling informed decisions on their adoption in software development. Exploring how these tools

assist in the decision-making process for adopting social sustainability practices offers an opportunity to incorporate social sustainability considerations into software development processes effectively.

As with other reviews, this research has limitations. The restriction to searching only on the Web of Science potentially overlooks relevant studies from other sources. Studies need to be done for further empirical validation of the synthesized practices' efficiency, usefulness, and contribution to improving social sustainability. There is a possible theoretical limitation since we did not explore practices that might be found in social science papers, limiting the investigation of social aspects in software engineering.

References

1. Ajmal, M.M., Khan, M., Hussain, M., Helo, P.: Conceptualizing and incorporating social sustainability in the business world. Inter Journal of Sustainable Development & World Ecology **25**(4), 327–339 (2018)
2. Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A., Storey, M.A.: The diversity crisis in software development. IEEE Software **38**(2), 19–25 (2021)
3. Alharthi, A.D., Spichkova, M., Hamilton, M.: Sustainability requirements for elearning systems: a systematic literature review and analysis. Requirements Engineering **24**, 523–543 (2019)
4. Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Mahaux, M., Penzenstadler, B., Rodriguez-Navas, G., Salinesi, C., Seyff, N., Venters, C., et al.: The karlskrona manifesto for sustainability design. arXiv preprint arXiv:1410.6968 (2014)
5. Brundtland, G.H.: Report of the world commission on environment and development: Our common future. https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf (1987), accessed on June 18, 2022
6. Calero, C., Guzmán, I.G.R.D., Moraga, M.A., García, F.: Is software sustainability considered in the csr of software industry? Journal of Sustainable Development & World Ecology pp. 439–459 (2019)
7. Cheng, C., Zhang, M.: Conceptualizing corporate digital responsibility: A digital technology development perspective. Sustainability **15**(3) (2023)
8. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: 2011 Inter symposium on empirical software engineering and measurement. pp. 275–284. IEEE (2011)
9. Dwivedi, Y.K.: Opinion paper: "so what if chatgpt wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. International Journal of Information Management **71** (2023)
10. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and software technology **50**(9-10), 833–859 (2008)
11. Eizenberg, E., Jabareen, Y.: Social sustainability: A new conceptual framework. Sustainability **9**(1), 68 (2017)
12. Government, U.K.: Equality act 2013. https://www.legislation.gov.uk/ukpga/2010/15/part/2 (2013), accessed on June 2, 2023
13. Guldner, A., Kern, E., Kreten, S., Naumann, S.: Criteria for sustainable software products: analyzing software, informing users, and politics. Software Sustainability pp. 17–42 (2021)

14. Gustavsson, J.L., Penzenstadler, B.: Blinded by simplicity: Locating the social dimension in software development process literature. In: 7th Inter Conf on ICT for Sustainability. pp. 116–127 (2020)
15. Hernandez Gonzalez, A., Calero, C., Perez Parra, D., Mancebo, J.: Approaching green bpm characterisation. Journal of Software: Evolution and Process **31**(2), e2145 (2019)
16. Higón, D.A., Gholami, R., Shirazi, F.: Ict and environmental sustainability: A global perspective. Telematics and Informatics **34**(4), 85–95 (2017)
17. Khalifeh, A., Farrell, P., Alrousan, M., Alwardat, S., Faisal, M.: Incorporating sustainability into software projects: a conceptual framework. Inter Journal of Managing Projects in Business **13**(6), 1339–1361 (2020)
18. Kitchenham, B., Charters, S.M.: Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report EBSE-2007-01, Keele University and University of Durham (2007)
19. Kohl Silveira, K., Prikladnicki, R.: A systematic mapping study of diversity in software engineering: A perspective from the agile methodologies. In: 12th Inter Works. on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 7–10 (2019)
20. Lago, P., Koçak, S.A., Crnkovic, I., Penzenstadler, B.: Framing sustainability as a property of software quality. Commun. ACM **58**(10), 70–78
21. McGuire, S., Shultz, E., Ayoola, B., Ralph, P.: Sustainability is stratified: Toward a better theory of sustainable software engineering. arXiv preprint arXiv:2301.11129 (2023)
22. Partridge, E.: Social sustainability': a useful theoretical framework. In: Australasian political science association annual conference. pp. 28–30 (2005)
23. Penzenstadler, B., Femmer, H.: A generic model for sustainability with process-and product-specific instances. In: The Workshop on Green in/by software engineering. pp. 3–8 (2013)
24. Perera, H., Hussain, W., Whittle, J., Nurwidyantoro, A., Mougouei, D., Shams, R.A., Oliver, G.: A study on the prevalence of human values in software engineering publications, 2015–2018. In: 42nd Inter Conf on Software Engineering. pp. 409–420 (2020)
25. Rescher, N.: Fairness. Routledge, 1st edn. (2002)
26. Schwartz, S.H., et al.: An overview of the schwartz theory of basic values. Online readings in Psychology and Culture **2**(1), 2307–0919 (2012)
27. Sen, A., Anand, S.: Sustainable human development: Concepts and priorities. UNDP United Nations Development Programme (1994)
28. de Souza, A.C.M., Cruzes, D.S., Jaccheri, L.: Sustainability-driven meetings as a way to incorporate sustainability into software development phases. In: 18th Inter Conf on Evaluation of Novel Approaches to Software Engineering, ENASE. pp. 597–604 (2023)
29. SUSO: Sustainability awareness framework. https://www.suso.academy/en/sustainability-awareness-framework-susaf/ (2022), accessed on Jun 23, 2023
30. Swacha, J.: Models of sustainable software: A scoping review. Sustainability **14**(1) (2022)
31. UNDP: Human development report 2021-22. UNDP United Nations Development Programme (2022), http://report.hdr.undp.org
32. Winkler, T.: Human values as the basis for sustainable software development. In: 2018 IEEE Inter Symposium on Technology and Society (ISTAS). pp. 37–42. IEEE (2018)

SLR References List

1. S01 - Albuquerque, D., et al.: A sustainability requirements catalog for the social and technical dimensions. In: Conceptual Modeling 40th Inter. Conf. (2021).
2. S02 - Chitchyan, et al.: Uncovering sustainability concerns in software product lines. Journal of Software: Evolution and Process (2017).
3. S03 - Condori-Fernandez, et al.: A nichesourcing framework applied to software sustainability requirements. In: 13th Inter. Conf. on Research Challenges in Information Science (2019).
4. S04 - Condori-Fernandez, N., Lago, P.: Towards a software sustainability-quality model: Insights from a multi-case study. In: 13th Inter. Conf. on Research Challenges in Information Science (2019).
5. S05 - Condori-Fernandez, N., Lago, P.: Characterizing the contribution of quality requirements to software sustainability. Journal of Systems and Software (2018).
6. S06 - Condori-Fernandez, N., et al.: An action research for improving the sustainability assessment framework instruments. Sustainability (2020).
7. S07 - de Magdaleno, M.I.A., Garcia-Garcia, J.: Sustainability and social responsibility reporting in open source software. Inter. Journal of the Commons (2015).
8. S08 - Poth, A., Nunweiler, E.: Develop sustainable software with a lean ISO 14001 setup facilitated by the EFIS® framework. In: Lean and Agile Software Development: 6th Inter. Conf. (2022).
9. S09 - Duboc, L., et al.: Do we really know what we are building? Raising awareness of potential sustainability effects of software systems in requirements engineering. In: 27th Inter. Requirements Engineering Conf. (2019).
10. S10 - Duboc, L., et al.: Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow. Requirements Engineering (2020).
11. S11 - Al Hinai, M., Chitchyan, R.: Building social sustainability into software: Case of equality. In: 2015 IEEE Fifth Inter. Workshop on Requirements Patterns (2015).
12. S12 - Al Hinai, M., Chitchyan, R.: Engineering requirements for social sustainability. In: ICT for Sustainability (2016).
13. S13 - Pham, Y.D., et al.: Towards a multi-dimensional representation for requirements of sustainable software. In: 28th Inter. Requirements Engineering Conf. (2020).
14. S14 - Calero, C., et al.: Is software sustainability considered in the csr of software industry? Inter. Journal of Sustainable Development and World Ecology (2019).
15. S15 - Ouhbi, S., et al.: Sustainability requirements for connected health applications. Journal of Software: Evolution and Process (2018).
16. S16 - Kopeć, W., et al.: Guidelines towards better participation of older adults in software development processes using a new spiral method and participatory approach. In: 11th Inter. Work. on Cooperative and Human Aspects of Software Engineering (2018).
17. S17 - Ferrario, M.A., et al.: Values-first SE: research principles in practice. In: Proceedings of the 38th Inter. Conf. on Software Engineering Companion. (2016).
18. S18 - Bomfim, C., et al.: Modelling sustainability in a procurement system: An experience report. In: 22nd Inter. Requirements Engineering Conf. (2014).
19. S19 - Penzenstadler, et al.: Who is the advocate? Stakeholders for sustainability. In: 2nd Inter. Workshop on Green and Sustainable Software (2013).