Sustainability-Driven Meetings as a Way to Incorporate Sustainability into Software Development Phases

Ana Carolina Moises de Souza[a], Daniela Cruzes[b] and Letizia Jaccheri[c]
Department of Computer Science, Norwegian University of Science and Technology, Høgskoleringen 1, Trondheim, Norway

Keywords: Software Sustainability, Social Sustainability, Sustainable Software Engineering, Agile Software Development.

Abstract: Software sustainability has been a trending topic in the last decade in academia. Studies related to software sustainability propose models, frameworks, or practices that can be applied in the industry. But most of these proposals are still not systematically adopted in the industry. Therefore, there is an opportunity to create a structured meeting to support the concrete adoption of sustainability practices in software development. This paper aims to provide an overview of these frameworks and how they can help facilitate sustainability-driven meetings (SusDM). Seeking this, we present practical examples and a workflow to prepare the meeting by applying the existing sustainability frameworks in SusDM. As a position paper, our hypothesis is that the contributions of this meeting may be related to improving the knowledge of software developers on sustainable software engineering, discovering new sustainability requirements, prioritization, and implementing software sustainability practices.

1 INTRODUCTION

Nowadays, Sustainability in Software Engineering is a trending research topic, and it is contextualized from software process, and software product perspectives (Calero et al., 2022). From the software product perspective (Penzenstadler et al., 2014a), proposed a framework to analyze the software product effects in each sustainability dimension. (Hilty and Aebischer, 2015), argued about the life cycle impact of an Information Technology and Communication (ICT) product, which can be reduced by improving its processes on environmental and social dimensions. Regarding sustainability as part of the software development process, (Lago et al., 2015) proposes that sustainability can be framed into software quality requirements, just as Security and Usability disciplines.

One recurrent problem in today's software development projects is the deprioritization of nonfunctional aspects (Tøndel et al., 2019). There is still a lack of systematic ways to deal with aspects of the software that encompass more than just one functionality. For example, in Security (Tøndel et al., 2019): a) security is often seen as not part of strategic decisions, b) security aspects have a tendency to be sacrificed in favor of implementing more functionality and c) responsibility for software security is often unclear. This previous work proposed such meetings for security as a way to increase and systematize the prioritization of security activities in software development projects (Tøndel et al., 2019) (Tøndel and Cruzes, 2022). Our hypothesis is that Software Sustainability suffers from the same challenges as security. The relationship between these areas is described by (Penzenstadler et al., 2014b) where sustainability is also seen as a non-functional requirement.

One evidence of these challenges was studied by (de Souza et al., 2022) that has performed a multi-case study aiming to understand how sustainability practices were addressed in the software development area of financial institutions. The results from the interviews have identified that some organizations do not apply sustainability in software development in a systematized way. Thus, organizational practices related to social, environmental, technical, and economic dimensions are found outside of the software development scope and mainly at the organizational level. (Calero et al., 2019) confirms through the analysis of ten Corporate Social Responsibility reports that there are more actions related to other areas than to soft-

[a] https://orcid.org/0000-0001-8620-6330
[b] https://orcid.org/0000-0002-2490-902X
[c] https://orcid.org/0000-0002-5547-2270

Moises de Souza, A., Cruzes, D. and Jaccheri, L.
Sustainability-Driven Meetings as a Way to Incorporate Sustainability into Software Development Phases.
DOI: 10.5220/0011984700003464
In *Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2023)*, pages 597-604
ISBN: 978-989-758-647-7; ISSN: 2184-4895

ware sustainability adoption in information technology companies.

Aiming to improve the software sustainability knowledge, organizational strategic view towards software sustainability, and well-defined roles and processes, we propose Sustainability-driven Meetings (SusDM). SusDM is organized into two main parts: preparation for the meeting and facilitation of the meeting. For the preparation part, we suggest a workflow that will allow practitioners to identify the business context, the end user, the internal and external stakeholders, the Software Development Life Cycle (SDLC) phase, and finally the steps to run the SusDM based on the output of the previous stages. The second part uses the existing sustainability frameworks, which will be discussed in Section 4, to facilitate the meetings. At the end of the SusDM, it is expected to have a list of practices, features, and process that reflects sustainability concerns to be mitigated by the software under construction.

As a position paper, we are planning to conduct a focus group as an observatory study to understand people's reactions and perceptions during the meeting (Stewart et al., 2007). Action research will complement the focus group in terms of understanding the current scenario of software sustainability in the context of agile software development (Davison et al., 2004). The meeting will be evaluated and improved by collecting feedback from participants and following up on actions taken after each SusDM session.

The paper structure is as follows. In section 2, we provide the research background. Section 3 contains an explanation of Sustainability-driven meetings. Section 4 includes a discussion about software sustainability frameworks and the possibilities of applying this meeting. Then, in section 5, we conclude and propose future research work.

2 BACKGROUND AND RELATED WORK

Trends in the Green and Sustainable Software research field indicate a growth in publications produced each year in this area (Calero et al., 2020). The contribution of this review reinforces the importance about sustainability research and shows the opportunities to make it valuable for the software development community and society. The contributions of existing frameworks are promising to start the conversation and awareness about sustainability in software (Messnarz et al., 2014), (Hinai and Chitchyan, 2016), (Ahmad et al., 2016), (Kopec et al., 2018), (Lago, 2019), (Duboc et al., 2020), (Condori-Fernandez et al., 2020), (Albuquerque et al., 2021). In the review conducted by (Calero et al., 2020), the studies were classified into knowledge areas of Software Engineering. The results show that Software Process demands research related to sustainability, and the most significant research contributions were in Software Quality studies.

In Software Sustainability, it is necessary to have activities linked to the software development lifecycle (SDLC). In agile software development, meetings are vital to executing the agile software development process. For example, SCRUM is supported by regular events, that are goal-driven meetings such as sprint planning, sprint review, sprint retrospectives, and daily scrum, where each meeting has its purpose (Schwaber and Sutherland, 2020). These meetings have very well-defined goals, and they are structured in terms of the role designated to run it, the timebox of the meeting, the preparation before the meeting, the selection of the right people involved, the definition of a goal, the follow-up actions after the meetings and its assessment and reflection to improve them (Andriyani et al., 2017). Scaled agile methods such as Scaled Agile Framework (SAFe), propose systematic meetings or events to run the method and ensure the process runs smoothly (SAFe, 2021a). The difference between not scaled and scaled methods is the complexity, the number of people involved, the coordination of distributed teams, and the strategies required to keep good enough communication between organizational areas, people, and customers (Uludag et al., 2021) (Dingsøyr et al., 2017) (Alzoubi and Gill, 2014). Kanban is another agile method, in which the purpose is a continuous flow managed by lead times and delivery of value between upstream and downstream areas. Feedback loops are necessary to improve the flow in all areas, therefore ”Kanban defines seven specific feedback opportunities or cadences. Cadences are the cyclical meetings and reviews that drive evolutionary change, and effective service delivery” (Anderson, 2010). On the current approaches for agile methods, there are no meetings with the goal of addressing software sustainability.

In the available literature about Sustainability in Software engineering and Green and Sustainable Software, we did not find related work to what we suggest in this paper - sustainability-driven meetings (SusDM). In most of the agile software development methods mentioned (Scrum, Kanban, and SAFe), the terms ”sustainability” and ”sustainable” are used to sustain or enhance something without wasting resources. For example, SAFe addresses Sustainability to deliver sustainable solutions to the customer and sustainable business agility (SAFe, 2023). In their view, to reach Sustainability, the teams must be able to release on demand. Thus, Sustainability requires continuous delivery from the software architect's perspective (SAFe, 2021b). On the other hand, Kanban and Scrum are concerned with the team's Sustainability in terms of sustainable pace and fair workload (Yeret, 2021). When dealing with software engineering, Sustainability is seen as a quality requirement that requires valuable systematic practices to implement and evaluate sustainability in software (Condori-Fernandez and Lago, 2019).

3 SUSTAINABILITY-DRIVEN MEETINGS

Sustainability-driven meetings aim to introduce the topic to be discussed systematically and reflect in each phase of software development. Figure 1 shows an example of the possible workflow of how the SusDM can be initiated in the software development context.

As mentioned before, the Sustainability-driven meeting was inspired by (Tøndel et al., 2019) (Tøndel and Cruzes, 2022). The authors proposed regular security meetings in three different companies conducted through action research and case study. The goals were to make security visible and tangible, reach key stakeholders, and ensure ongoing security prioritization. Along with these goals, a set of recommended strategies were combined based on the software security literature. In this context, it was possible to identify what influences the adoption of security meetings regarding driving force, visibility, motivation, room to manoeuvre and process match. An example of the positive effects of this meeting is "As meeting participants gained more competence and confidence on security, they improved their ability to act as a driving force for security." Moreover, this study has shown that specific security meeting has increased the security tasks because the team had dedicated time to discuss them, without impact to the project delivery date. The main difference between the Security Intention meeting (Tøndel and Cruzes, 2022) and the SusDM is about the preparation workflow required before SusDM starts (Figure 1). This is because software sustainability is still an emerging topic in the industry and therefore requires guided instruction on how to adopt it in the SDLC (Calero et al., 2019).

It is essential to be prepared before the meeting starts, which will take some time in the first place. However, the following meetings will probably not require much effort in preparation.

The workflow starts on the **01-Discovering** stage, which is divided into two steps. The first step in this stage is Business Domain, which aims to identify the business context. This context is usually the purpose of the software. For example, if the software is developed to manage blood donations, the business context might be Health. The discovery of business context to address sustainability concerns allows us to refine better and create features that will positively impact society. Thus, sustainability is very context-dependent (Miller, 2012). The second step of 01-Dicovering stage is to identify who will benefit from the software and to whom the software is intended to be used, which means the end-user of the software. In this case, the software development area could use techniques from the user experience field to help with this discovery. We selected some examples of user experiences techniques listed in Figure 1: Personal Maps (Team, 2022), Ecosystem Map (UX-republic, 2022), Empathy Map (Brown, 2018), and User Journey Map (Bradley et al., 2021). We argue that user-centered designing contributes to social sustainability and the transfer ability to human values once the designers' and researchers' responsibility is "to think more considerately and diversely toward individuals' varied" (Bradley et al., 2021).

The next stage is **02-Stakeholders Map** which is divided into two steps. The first one is the Internal mapping of the stakeholders inside the organizations, which plays a significant role during software development. The internal stakeholders are responsible for creating the product backlog, management, organization operations, and strategies. These people are essential to contribute to the strategic vision of Sustainability and its short or long-term benefits for the product and organization. They also have the power to influence the adoption of Sustainability by developers and operational levels. We believe software developers are important internal stakeholders because their technical views can drive the decisions. External stakeholders are people who have fundamental knowledge about sustainability, know the regulations or by-laws related to this area, and share experiences in other areas that are not related to software development. For example, an external stakeholder can provide an overview of how to operate a non-organizational institution that fights hunger and poverty.

The **03-SDLC** stage is to situate the software under construction in which phase of the Software Development Life Cycle is happening. The software development process could help identify SDLC phases independently of agile or traditional methods. The difference will be the number of interactions needed

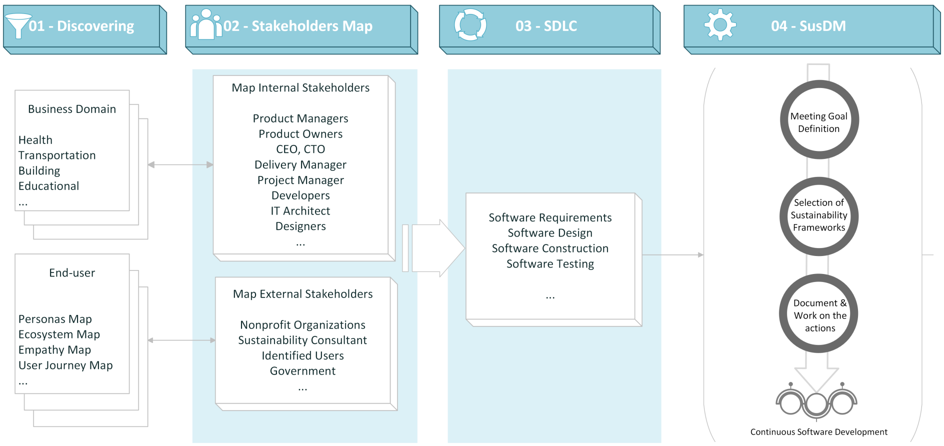

Figure 1: The Sustainability-driven meeting workflow.

and the project's planned schedule. In both approaches, we see room to accommodate the SusDMs. For example, suppose the team is using SCRUM, and the goal of the meeting is to identify sustainability features. In that case, SusDM can be accommodated at the backlog refinement meeting to prepare for the sprint. Of course, at this point, the previous stages were done. Given their importance and intention, the discovery and stakeholder map outputs are required in the SDLC stage.

After finishing all the previous stages, we can run the Sustainability-Driven Meeting **04-SusDM**. It consists of three steps, defining meeting goals, selecting sustainability frameworks, documenting, and working on the actions. The SDLC phase is an input to determine the meeting goals and, therefore, to establish the sustainability framework. Examples of meeting goals and sustainability frameworks are discussed below:

The Sustainability-driven meetings are flexible to be adopted by any business domain and at any phase of software development. We list some examples of the different purposes of this meeting: **Software Requirements:** Derive software features that will contribute to society in one or more sustainability dimensions.

Software Design. Identify sustainability approaches that can be designed or avoided in the software before its development.

Software Architecture. Decide which software architecture approaches will contribute to sustainability.

Software Construction. Identify the efficiency, response time, and use of software resources in order to reduce the environmental impacts of its use or development.

Software Testing. Identify actors from the society that could test and validate the sustainability features. List the sustainability metrics reached by software usage.

Software Development Process. Reflect on the improvement of software development and whether it is sustainable or not to keep from the organization, product, and team perspective.

Once the meeting goal is defined, we can combine the existing sustainability frameworks into the meeting and the software development. The sustainability frameworks are discussed in section 4 based on an ongoing systematic literature review.

The last step is to document the meeting results, which means all the notes taken during the meeting should find their way into any tools used for tracking. The meeting output could be: a) the results and next steps of the discussion, and b) framework usage/feedback for further reference. The meeting outcome can be related to its goals as well. For example, if the goal were to discover new sustainability features of the software, then the outcome would be a list of these features and, perhaps, a detailed user story.

As recommended by (Tøndel et al., 2019), the meeting participants can evaluate the meeting outcomes. In our case, it is also valuable to assess the SusDM as well, so we adapted some of the grades proposed:

- Great: We are doing great during the meeting and the goal will be reached, no further improvement is needed.
- Good: We know we could be better, but we are fairly satisfied with the current practice. We hope to reach the goal.
- Somewhat: We are doing some things, the goal seems ok, but we really should improve the Sustainability-driven meeting.

- Lacking: We are doing close to nothing during the meeting and are far from realizing the goal proposed in the meeting.

4 SOFTWARE SUSTAINABILITY FRAMEWORKS

We identified through an ongoing Systematic Literature Review that Software Requirements are the most explored software development life cycle phase in terms of suggesting models or frameworks. From the software engineering perspective, we have selected the frameworks proposed in the literature that has shown evidence of their applications and could be a tool for the SusDM facilitation to achieve its goals as described in Table 1.

4.1 Software Requirements

Software requirements activities explicitly provide conditions to achieve sustainable software from both perspectives: product and process (Calero et al., 2022). The proposed frameworks in the studies selected provide instructions for their usage by the software development teams allowing us to recommend them as a tool to facilitate Sustainability-driven meetings ((Hinai and Chitchyan, 2016), (Condori-Fernandez and Lago, 2019), (Duboc et al., 2020),(Albuquerque et al., 2021)).

Generic requirements identification was proposed by (Hinai and Chitchyan, 2016) based on core societal values, such as equality and fairness, which are connected with social sustainability goals. The author analyzes six sample studies and creates equity value patterns from them. The patterns created and how they were interpreted provide insights into social sustainability. During the SusDM, it can facilitate the identification of requirements related to social sustainability and work ”as an enabler for these concerns to be integrated into the SE (software engineering) practice” (Hinai and Chitchyan, 2016).

(Condori-Fernandez and Lago, 2019) proposed a software Sustainability-Quality Model(SQM) to support software engineers in identifying relevant quality attributes, such as integrity, user error protection, learnability, interoperability, modifiability, adaptability, capacity, and co-existence. These qualities were categorized into a sustainability dimension, making it easy to connect with real sustainability issues. This model could facilitate addressing non-functional sustainability requirements and evaluating them based on the identified quality attributes.

Table 1: Mapping the existing sustainability tools to facilitate SusDM by SDLC.

SDLC/Sustainability Tool	Dimension
- Software Requirements:	
Values patterns for social sustainability requirements. (Hinai and Chitchyan, 2016)	Social
Software Sustainability Quality Model (SQM) (Condori-Fernandez and Lago, 2019)	Soc., Env., Tech., Eco.
Sustainability Awareness Framework and Diagram (SusAF and SusAD) (Duboc et al., 2020)	Soc., Env., Tech., Eco., Ind.
Sustainability Requirements catalog (Albuquerque et al., 2021)	Soc., Tech.
- Software Design:	
Software Sustainability Assessment Framework (SAF). Sustainability Quality Model (SQ model) and the Architectural Decision Map (DM) (Lago, 2019) (Condori-Fernandez et al., 2020)	Soc., Env., Tech., Eco.
- Software Architecture:	
Architectural Decision Map (DM) (Lago, 2019) (Condori-Fernandez et al., 2020)	Soc., Env., Tech., Eco.
- Software Construction:	
Characteristics and Sub-Characteristics of Software Sustainability Evaluation (Ahmad et al., 2016)	Env., Soc., Eco.
- Software Testing:	
Support for Participant Involvement in Rapid and Agile software development Lab (SPIRAL) user involvement scheme (Kopec et al., 2018)	Social
- Software Process Improvement:	
Mapping between Software Process Improvement (SPI) and Social Responsibility issues based on ISO26000 Evaluation. (Messnarz et al., 2014)	Social

The sustainability Awareness Framework (SusAF) aims to raise awareness of the sustainability effects of software. It is a question-based framework to guide the discussion with stakeholders (Duboc et al., 2020). As part of software requirements, it can engage the stakeholders in software sustainability concerns. The results of its discussion can be represented in a Sustainability Awareness Diagram (SusAD), a visualization tool used to highlight the chains of effects (Duboc et al., 2020). During the SusDM, these frameworks can help in the discovery phase of software requirements and visualize each effect.

The implementation and measure of non-functional requirements are not trivial, as studies regarding this topic confirm (Albuquerque et al., 2021). The challenges seen by the experts are related to technical skills, tooling, testing, and metrics related to non-functional requirements. Aiming to facilitate this meeting, a sustainability catalog can be used by the people involved in implementing this software (Albuquerque et al., 2021). This catalog will also help identify the relationship between the non-functional requirements and how they affect each other, neither positively nor negatively. It will clarify the sustainability concerns and who contributes to enabling sustainability in the software (Albuquerque et al., 2021).

4.2 Software Design and Software Architecture

Social-technical decisions regarding the software structure are made during the software design phase. In this case, the software design can also address the sustainability dimensions, such as social, environmental, and economic. We suggest that a Sustainability-driven meeting can help with the discussion regarding sustainability concerns and help to decide which path to go towards Sustainable software.

In this case, both tools can be combined as proposed by (Condori-Fernandez et al., 2020) (Lago, 2019). First, a tool to create awareness about the aspects of each sustainability dimension based on the domain and software goal. Second, the decision maps can be elaborated to allow software designers to decide the most critical aspect to implement in their software. The benefit of the decision maps is extended to Software Architecture (Lago, 2019).

4.3 Software Construction

During the Software Construction, one checkpoint can be made to assess whether the software's efficiency, response time, and resource utilization aim to reduce the environmental impacts related to its usage or development. (Ahmad et al., 2016) proposed a way to evaluate the sustainability of the software under development based on goal-oriented approaches. An example of defining the goal of software sustainability evaluation for an environmental dimension is (Ahmad et al., 2016) : *”Purposes: To evaluate the utilized resources behavior of computer systems during testing or operation in order to improve it. Perspectives: Examine the input-output resource utilization from the user, developer, maintainers, and SQA point of view. Environmental: the following context of quality of use of resources towards environmental impacts.”* During the SusDM, the list of characteristics and sub-characteristics can be verified in the software construction phase. If not implemented, it can be turned into actions and backlog items to be delivered in the sprint, for example.

4.4 Software Testing

User participation in co-design software was studied by (Kopec et al., 2018) focusing on old adults, but it also can be generalized to other groups of people. The intention of SPIRAL (Support for Participant Involvement in Rapid and Agile software development Lab) is to develop a co-design approach that engages the end-user participation in the phases of defining an idea, analyzing, designing, prototyping, and testing (Kopec et al., 2018). The contribution of this study is a set of design steps to include people and promote participatory software development of a product.

In software testing, having the actual end-users test the Software is essential to guarantee its functional requirements. Once the user is involved, much information can be collected to enhance the software product. In the SPIRAL proposal, ”the least advanced users may participate in the idea generation and testing phases, more advanced can join the design and analysis stages, whereas the most advanced can take part in the full development cycle as experts” (Kopec et al., 2018).

4.5 Software Process Improvement

Social Responsibility concerns can be part of Software Process Improvement (SPI) principles (Messnarz et al., 2014). The mapping between the ISO 26000 - Social Responsibility and SPI Manifesto was created to demonstrate how they are connected. The SPI manifesto can be used in SusDM to reflect what needs to be improved in the software development process toward social responsibility.

One of the SPI principles is ”create a learning organization," which is about facilitating learning and sharing experiences. Influenced by this principle, the SusDM meeting opens the door to learning about software sustainability. The contribution of (Messnarz et al., 2014) is to serve as a checklist where the internal and external stakeholders can help to assess whether these principles and social responsibilities are addressed.

This section discussed in which moment of software development the meeting can be addressed, which frameworks can be used as facilitating tools to promote the discussion, and the argument why use each of them. The software development team can decide when it is the moment to add this meeting. We aim to apply this meeting in action research design to understand how software development companies can adopt SusDM.

5 CONCLUSIONS

In this paper, we argued that a Sustainability-driven meeting is a practical way to address sustainability concerns, create sustainability awareness and realistically adopt sustainability during the software development by using existing evidenced sustainability frameworks. Throughout the paper, we discussed the meeting workflow considering the stages of discovering, mapping stakeholders, identifying SDLC phases, and running the SusDM.

The main idea is to hold this meeting in agile software development teams following the focus group research method (Stewart et al., 2007). We want to observe the participants' engagement and interactions during the application of the sustainability framework chosen for their context based on steps 01, 02 and 03 of the preparation for the SusDM. We plan to conduct this research with the action research method, as we want to understand the problem, propose solutions collaboratively, and run feedback loops (Davison et al., 2004). In the end, we want to assess how valuable these meetings are for developers and how knowledge about software sustainability has improved after starting these meetings.

While other approaches focus mainly on general sustainability awareness or specific phases of software development (Calero et al., 2022), the SusDM is proposed to be added at any phase of software development. The SusDM gives the software development team flexibility regarding when it needs to execute and which tools can be used to reach the meeting goal. The meeting can be assessed to find value in rerunning it or improving it for the next rounds.

The meeting proposed in this paper combines existing sustainability frameworks into software development phases to facilitate the meeting and increase the prioritization of sustainability. In addition, it provides examples of SusDM goals that can be addressed during software development and how they should be initiated in the organization, as stated in the workflow section. Further work will explore how SusDM can be adopted in software development companies. We expect to validate the suggestion, collect feedback and improve it to be operational in any software development context.

REFERENCES

Ahmad, R., Hussain, A., and Baharom, F. (2016). Goal oriented measurement for software sustainable evaluation metric focused on environmental dimension.

Albuquerque, D., Moreira, A., Araujo, J. a., Gralha, C., Goulão, M., and Brito, I. S. (2021). A sustainability requirements catalog for the social and technical dimensions. page 381–394, Berlin, Heidelberg. Springer-Verlag.

Alzoubi, Y. I. and Gill, A. Q. (2014). Agile global software development communication challenges: a systematic review. In Siau, K., Li, Q., and Guo, X., editors, *18th Pacific Asia Conference on Information Systems, PACIS 2014, Chengdu, China, June 24-28, 2014*, page 20.

Anderson, D. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

Andriyani, Y., Hoda, R., and Amor, R. (2017). Reflection in agile retrospectives. In Baumeister, H., Lichter, H., and Riebisch, M., editors, *18th Inter. Conference, XP*, volume 283 of *Lecture Notes in Business Information Processing*, pages 3–19.

Bradley, C., Oliveira, L., Birrell, S., and Cain, R. (2021). A new perspective on personas and customer journey maps: Proposing systemic ux. *Journal of Human-Computer Studies*, 148:102583.

Brown, J. L. (2018). Empathy mapping: A guide to getting inside a user's head.

Calero, C., Guzmán, I. G.-R. D., Moraga, M. A., and García, F. (2019). Is software sustainability considered in the csr of software industry? *Journal of Sustainable Development & World Ecology*, pages 439–459.

Calero, C., Mancebo, J., Garcia, F., Angeles Moraga, M., Garcia Berna, J. A., Luis Fernandez-Aleman, J., and Toval, A. (2020). 5ws of green and sustainable software. *Tsinghua Science and Technology*, pages 401–414.

Calero, C., Ángeles Moraga, M., and García, F. (2022). Software, sustainability, and un sustainable development goals. *IT Professional*, pages 41–48.

Condori-Fernandez, N. and Lago, P. (2019). Towards a software sustainability-quality model: Insights from a multi-case study. In *2019 13th Inter. Conference on Research Challenges in Information Science (RCIS)*, pages 1–11.

Condori-Fernandez, N., Lago, P., Luaces, M. R., and Places, Á. S. (2020). An action research for improving the sustainability assessment framework instruments. *Sustainability*, 12(4).

Davison, R., Martinsons, M. G., and Kock, N. (2004). Principles of canonical action research. *Information Systems Journal*, 14:65–86.

de Souza, A. C. M., Reinehr, S., and Malucelli, A. (2022). Sustainable software engineering: An empirical study of the brazilian financial sector. In Canciglieri Junior, O., Noël, F., Rivest, L., and Bouras, A., editors, *Product Lifecycle Management. Green and Blue Technologies to Support Smart and Sustainable Organizations*, pages 115–129.

Dingsøyr, T., Rolland, K., Moe, N. B., and Seim, E. A. (2017). Coordination in multi-team programmes: An investigation of the group mode in large-scale agile software development. *Procedia Computer Science*, 121:123–128.

Duboc, L., Penzenstadler, B., Porras, J., Akinli Kocak, S., Betz, S., Chitchyan, R., Leifler, O., Seyff, N., and Venters, C. C. (2020). Requirements engineering for sustainability: An awareness framework for designing software systems for a better tomorrow. 25(4):469–492.

Hilty, L. M. and Aebischer, B. (2015). ICT for sustainability: An emerging research field. In Hilty, L. M. and Aebischer, B., editors, *ICT Innovations for Sustainability*, volume 310 of *Advances in Intelligent Systems and Computing*, pages 3–36. Springer.

Hinai, M. A. and Chitchyan, R. (2016). Engineering requirements for social sustainability. In *Proc. of ICT for Sustainability 2016*, pages 79–88. Atlantis Press.

Kopec, W., Nielek, R., and Wierzbicki, A. (2018). Guidelines towards better participation of older adults in software development processes using a new spiral method and participatory approach. In *Proc. of the 11th Inter. Workshop on Cooperative and Human Aspects of Software Engineering*, page 49–56.

Lago, P. (2019). Architecture design decision maps for software sustainability. In *2019 IEEE/ACM 41st Inter. Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 61–64.

Lago, P., Koçak, S. A., Crnkovic, I., and Penzenstadler, B. (2015). Framing sustainability as a property of software quality. *Commun. ACM*, 58(10):70–78.

Messnarz, R., Sicilia, M.-A., Biro, M., Barriocanal, E. G., Garre-Rubio, M., Siakas, K. V., and Clarke, A. (2014). Social responsibility aspects supporting the success of SPI. *J. Softw. Evol. Process.*, 26(3):284–294.

Miller, T. R. (2012). Constructing sustainability science: emerging perspectives and research trajectories. *Sustainability Science*, 8:279–293.

Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H., and Franch, X. (2014a). Systematic mapping study on software engineering for sustainability (se4s). In *Proc. of the 18th Inter. Conference on Evaluation and Assessment in Software Engineering*.

Penzenstadler, B., Raturi, A., Richardson, D., and Tomlinson, B. (2014b). Safety, security, now sustainability: The nonfunctional requirement for the 21st century. *IEEE Software*, 31(3):40–47.

SAFe (2021a). Essential safe.

SAFe (2021b). Solution architect/engineering.

SAFe (2023). Embracing safe in finance to become a data-driven company.

Schwaber, K. and Sutherland, J. (2020). The scrum guide - the definitive guide to scrum: The rules of the game.

Stewart, D., Shamdasani, P., and Rook, D. (2007). *Focus Groups: Theory and Practice*. Applied Social Research Methods. SAGE Publications.

Team, I. E. (2022). What is persona mapping? 4 tips to create your map.

Tøndel, I. A. and Cruzes, D. S. (2022). Continuous software security through security prioritisation meetings. *J. Syst. Softw.*, 194:111477.

Tøndel, I. A., Cruzes, D. S., Jaatun, M. G., and Rindell, K. (2019). The security intention meeting series as a way to increase visibility of software security decisions in agile development projects. In *Proc. of the 14th Inter. Conference on Availability, Reliability and Security, ARES*, pages 59:1–59:8. ACM.

Uludag, Ö., Putta, A., Paasivaara, M., and Matthes, F. (2021). Evolution of the agile scaling frameworks. In Gregory, P., Lassenius, C., Wang, X., and Kruchten, P., editors, *22nd Inter. Conference on Agile Software Development*, volume 419 of *Lecture Notes in Business Information Processing*, pages 123–139. Springer.

UX-republic (2022). Empathy mapping: A guide to getting inside a user's head.

Yeret, Y. (2021). Working toward sustainable pace in scrum and kanban.