



Lessons Learned from Developing a Sustainability Awareness Framework for Software Engineering Using Design Science

Downloaded from: <https://research.chalmers.se>, 2024-06-26 13:25 UTC

Citation for the original published paper (version of record):

Betz, S., Penzenstadler, B., Duboc, L. et al (2024). Lessons Learned from Developing a Sustainability Awareness Framework for Software Engineering Using Design Science. *ACM Transactions on Software Engineering and Methodology*, 33(5). <http://dx.doi.org/10.1145/3649597>

N.B. When citing this work, cite the original published paper.



Lessons Learned from Developing a Sustainability Awareness Framework for Software Engineering Using Design Science

STEFANIE BETZ, Furtwangen University and LUT University, Furtwangen, Germany

BIRGIT PENZENSTADLER, Chalmers University of Technology and LUT University, Gothenburg, Sweden

LETICIA DUBOC, La Salle, University Ramon Llull, Barcelona, Spain

RUZANNA CHITCHYAN, University of Bristol, Bristol, UK

SEDEF AKINLI KOCAK, Vector Institute for Artificial Intelligence, Toronto, Canada

IAN BROOKS, University of the West of England, Bristol, UK

SHOLA OYEDEJI, LUT University, Lappeenranta, Finland

JARI PORRAS, LUT University, Lappeenranta, Finland

NORBERT SEYFF, University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland

COLIN C. VENTERS, University of Huddersfield, Huddersfield, UK

To foster a sustainable society within a sustainable environment, we must dramatically reshape our work and consumption activities, most of which are facilitated through software. Yet, most software engineers hardly consider the effects on the sustainability of the IT products and services they deliver. This issue is exacerbated by a lack of methods and tools for this purpose. Despite the practical need for methods and tools that explicitly support consideration of the effects that IT products and services have on the sustainability of their intended environments, such methods and tools remain largely unavailable. Thus, urgent research is needed to understand how to design such tools for the IT community properly. In this article, we describe our experience using design science to create the Sustainability Awareness Framework (SusAF), which supports software engineers in anticipating and mitigating the potential sustainability effects during system development. More specifically, we identify and present the challenges faced during this process. The challenges that we have faced and addressed in the development of the SusAF are likely to be relevant to others who aim to create methods and tools to integrate sustainability analysis into their IT products

This research was supported by the Departament de Recerca i Universitats de Catalunya (Ref. 2021 SGR 01396) and EPSRC (project EP/R007373/1).

Authors' addresses: S. Betz, Furtwangen University and LUT University, Robert-Gerwig-Platz 1, Furtwangen, Germany; e-mail: besi@hs-furtwangen.de; B. Penzenstadler, Chalmers University of Technology and LUT University, Chalmersplatsen 4, Gothenburg, Sweden; e-mail: birgitp@chalmers.se; L. Duboc, La Salle, University Ramon Llull, C/ de Sant Joan de la Salle, 42, Barcelona, Spain; e-mail: l.duboc@salle.url.edu; R. Chitchyan, University of Bristol, Beacon House, Queens Road, Bristol, UK; e-mail: r.chitchyan@bristol.ac.uk; S. A. Kocak, Vector Institute for Artificial Intelligence, 661 University Avenue, Suite 710, Toronto, ON M5G 1M1, Toronto, Canada; e-mail: sedef.kocak@vectorinstitute.ai; I. Brooks, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, Bristol, UK; e-mail: ian.brooks@uwe.ac.uk; S. Oyedeji and J. Porras, LUT University, P.O. Box 836, FI-00074, Lappeenranta, Finland; e-mail: Shola.Oyedeji@lut.fi, Jari.Porras@lut.fi; N. Seyff, University of Applied Sciences and Arts Northwestern Switzerland, Freilager-Platz 1, Windisch, Switzerland; e-mail: norbert.seyff@fnw.ch; C. C. Venters, University of Huddersfield, Queensgate, Huddersfield, UK; e-mail: c.venters@hud.ac.uk.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

ACM 1049-331X/2024/06-ART136

<https://doi.org/10.1145/3649597>

and services development. Thus, the lessons learned in SusAF development are shared for the benefit of researchers and other professionals who design tools for that end.

CCS Concepts: • **Software and its engineering** → **Designing software**; • **General and reference** → *Empirical studies*; • **Social and professional topics** → **Sustainability**

Additional Key Words and Phrases: IT products, IT services, sustainability analysis

ACM Reference Format:

Stefanie Betz, Birgit Penzenstadler, Leticia Duboc, Ruzanna Chitchyan, Sedef Akinli Kocak, Ian Brooks, Shola Oyediji, Jari Porras, Norbert Seyff, and Colin C. Venters. 2024. Lessons Learned from Developing a Sustainability Awareness Framework for Software Engineering Using Design Science. *ACM Trans. Softw. Eng. Methodol.* 33, 5, Article 136 (June 2024), 39 pages. <https://doi.org/10.1145/3649597>

1 INTRODUCTION

The earth’s natural resources and capacity to absorb pollutants are limited. We have already overstepped several of the ecological boundaries to maintain a safe and stable ecosystem on our planet [57]. Similarly, we are failing to safeguard the minimum standards for human well-being and social justice [94]. Thus, immediate and pervasive changes are needed in all sectors of all societies [5, 11, 70, 71].

This also applies to the IT sector and not simply because the current CO₂ emissions from the IT sector are already on par with that of the aviation industry (i.e., about 3% of global CO₂ emissions [43]). However, some scientists estimate that these will exceed 14% of the 2016-level worldwide greenhouse gas emissions by 2040 [10]. IT products and services used in our societies also change our business processes and human behaviors (e.g., trade and commerce reshaped with online trading platforms, entertainment with streaming services, communication with messenger services) [100].

The **Software Engineering (SE)** community is responding to this challenge by developing tools, methods, techniques, and approaches that attempt to consider sustainability as part of software and service development—for example, the **Sustainability Awareness Framework (SusAF)** [34], WinWin [105], goal modeling [75], the **Sustainability Assessment Framework (SAF)** [27], and GreenSoft [29]. For simplicity, we will refer to these as “tools.”

Developing such tools is not a straightforward endeavor, as the concept of sustainability is often difficult to grasp, and the effects of IT products and services on sustainability are hard to predict. Therefore, the SE community needs to share experiences and lessons learned on how to develop sustainability tools that can be applied in the IT industry and in the education of software engineers. By drawing on each other’s experiences of developing such tools, the SE community can enhance the quality of their work and avoid common mistakes.

In this article, we describe an in-depth case study of developing a sustainability tool, the SusAF. The goal of the SusAF is to identify potential short-, medium-, and long-term sustainability effects of IT products and services. Using SusAF results, the IT product or service owner companies can decide how to progress with their IT product/service development and evolution. For instance, they may decide not to develop a certain feature which, as per SusAF analysis, would lead to potential negative effects or to implement various features for preventing or mitigating negative impacts. SusAF is best applied in the early **Requirements Engineering (RE)** [7]. However, it can also be applied to evaluate and plan evolutionary changes [84].

The SusAF was created following the design science research paradigm. In this paradigm, knowledge and understanding of a design problem and its solution are acquired through iterative development and evaluation of an artifact [38]. SusAF design iterated through several cycles,

whereby both students and companies evaluated the framework, flagging the challenges in understanding and applying it, and informing its ongoing development and improvements. SusAF is publicly available [13] and has already been presented in previous publications [34, 35]. Therefore, the aim of this article is not to present the framework itself. Instead, it describes the rigorous process of creation and evaluation of the SusAF recounting the 13 design cycles of its development, spanning almost a decade of collaborative work. Moreover, the article discusses the lessons learned from this process and the implications that these lessons could have on SE research, education, and practice.

Thus, the novel contributions of this article are threefold: (i) it presents for the first time the holistic design science based process of the SusAF development; (ii) it documents a number of challenges that are likely to be encountered when designing sustainability analysis tools for IT products and services and discusses their implications for SE research, education, and practice; and (iii) it discusses the strengths and weaknesses of applying design science to this end.

The rest of this article is structured as follows. Section 2 outlines the related work and the background on which this work builds, including notions of sustainability, SE, and design science research. Section 3 describes the methodology used and the artifacts developed. The study design is outlined in Section 4. In Section 5, the article explains in detail how the design science research was conducted, presenting each iteration cycle of design, evaluation, and improvements, including lessons learned. The implications of the lessons learned and challenges expected in designing a sustainability tool for IT products and services development are discussed in Section 6. Finally, the article concludes with a summary and an outlook on future research directions and challenges (Section 7).

2 BACKGROUND AND RELATED WORK

2.1 Sustainability and SE

The Karlskrona Manifesto [8] has served as a focal point for the SE community to engage with the concept of sustainability by advocating a set of fundamental principles and commitments that underpin sustainability design. The manifesto advocates that sustainability must be viewed as a construct across five dimensions: environmental, economic, social, individual, and technical. When considering the sustainability of the sociotechnical system, the effects that software may cause can be categorized across the order of effects, as direct (i.e., caused by the direct function of the system, its development, and its disposal), enabling (i.e., arising from the application of a system over time), or structural (i.e., referring to persistent changes that can be observed in the macro level) [54] (Table 1). Normally these effects are only noticed in hindsight, and producing evidence requires a significant resource investment. The notion of sustainability has been discussed extensively in a number of publications, and readers are directed to these for an in-depth treatment of this topic [55, 62, 81, 108, 112, 113]. We argue that sustainability is a concern independent of the purpose of the system and should be considered even if the primary focus of the system under design is not sustainability [7].

While traditional RE methods and tools do not explicitly facilitate the discussion of sustainability-related concerns, research suggests that existing RE techniques, approaches, and methods can serve as a starting point for practitioners to integrate sustainability into their practice [22, 25, 72, 105]. However, a careful look at the referenced literature suggests that the existing RE approaches are heavily biased toward sustainability goals related to second-order effects [111]. As a result, to properly address the prospective impact of software products and services on sustainability, there is a need to enable assessment of the prospective effects across time and dimensions.

A few approaches have tried to take this cross-time and dimensional perspective [12, 19, 27, 75, 86, 105]. For example, the approach by Condori-Fernandez et al. [27] presents

Cycle	C1	C3				C5			C7			C8			C10		C12		C13
Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Chitchyan et al., 2016	x																		
Penzenstadler et al., 2018		x																	
Duboc et al., 2019							x	x											
Duboc et al., 2020			x	x						x	x	x							
Porras et al., 2021														x					

Fig. 1. Relation of previous work to datasets. All cycles in this figure are relevance cycles.

SAF, a decision-supporting tool for software architects. SAF consists of two parts: the software sustainabilityquality model, and the architectural decision map. Most recently, Fatima and Lago [39] contributed to the idea of a blueprint for software architecture evaluation methods that consists of 11 general steps divided into three phases. Furthermore, Saputri and Lee [99] offer an integrated framework for requirements elicitation, the evaluation of which showed that participants could quickly identify more stakeholders, requirements, and features. However, unlike SusAF, this work does not look at the longer-term impacts of its (intended) IT products and services.

Other solutions have been proposed to tackle the various aspects of the sustainability-related challenges, such as the use of a recommender system to overcome the barriers of incorporating sustainability into the SE process [97], the application of a sustainability requirement pattern to guide the specification of sustainability requirements [98], an economics-driven architectural evaluation method which extends the CBAM (Cost Benefits Analysis Method) and integrates principles of modern portfolio theory to control the risks when linking sustainability concerns to architectural design decisions [76], a tool for requirement engineers to analyze the effects of the requirements on system sustainability [3], and a multi-dimensional framework for identifying requirements combined with a developer-oriented representation guideline [90]. However, there has been no comparative evaluation of the proposed methods to demonstrate their efficacy in identifying sustainability requirements against the various dimensions of sustainability, or in fostering sustainability-inducing designs.

This article discusses the systematic development of the SusAF using design science. It builds on our previous work by carrying out a deep synthesis of the learning obtained through the iterative process of design, use, and evaluation of the SusAF and its components throughout 10 years of SusAF development. While the framework itself has been presented in previous publications [34, 35], this article provides a reflective review of the development process itself, bringing together all available datasets. While the lessons learned in the present work are partially derived from the conclusions of the previous works, they are of a completely different nature; they refer to the process of designing the SusAF and its implication to education and practice and distilling a set of novel lessons learned and their implications for SE. We next detail the relationship between the current work and the previous ones on SusAF.

2.1.1 On the Relationship with Previous SusAF Work. This work synthesizes the lessons learned throughout the development of the work published in five papers [24, 34, 35, 85, 91]. It also presents several new datasets (i.e., information on professionals and students who have been interviewed or who have used SusAF), new theoretical background, and the resultant lessons learned. More specifically, the previously published papers describe in detail 8 of the 13 cycles and 9 of the 19 datasets, as illustrated in Figures 1 and 2.¹

¹For a detailed explanation of how each of these previous works relates to the current paper, please refer to <https://zenodo.org/records/10462231>

Cycle	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
Chitchyan et al., 2016	x												
Penzenstadler et al., 2018		x	x										
Duboc et al., 2019				x	x								
Duboc et al., 2020			x	x		x	x						
Porras et al., 2021								x					

Fig. 2. Relation of previous work to cycles. Blue columns refer to relevance cycles (with datasets), whereas green ones refer to rigor and design cycles.

2.2 Design Science and SE

Design science is a research methodology that takes a systematic, iterative approach to evaluation within research projects to develop a deeper understanding of phenomena under investigation, through observation and reasoning, from which general principles and laws can be deduced [117]. Design science is an established research paradigm adopted in many fields of **Information Systems (IS)** and other engineering disciplines to solve real-world problems [52]. The ultimate goal of design science research is to produce general design knowledge rather than to solve the problems of unique instances [37]. However, design science is not a software development methodology, nor does it mandate any particular development method. For example, it should not be seen as an alternative choice to agile methods.

Bjarnason et al. [14] applied design science to the development of an SE improvement method, Gap Finder, that was designed to increase requirements-test alignment. The method was evaluated through a case study in which it was applied to an ongoing IT development project. The project team found the design science approach to be useful and supported joint reflection and improvement of the method. For example, Goodrum et al [46] applied design science to aid in the discovery of the informational needs of five safety-critical system practitioners and 14 experienced developers as they engaged in software maintenance activities, then proposed and evaluated techniques for presenting and visualizing this information. They argued that this approach is particularly appropriate in investigating visual representations which also requires a highly iterative approach. Their study included two iterations of the process and was initially informed by in-depth interviews with safety-critical system practitioners. The results highlight that the design science approach, even with a limited number of participants, provides useful usability feedback to iteratively refine a design.

Schorr and Hvam [102] explored how IT managers can use design science to define and evaluate requirements for the information content of IT service catalogues in the early phase of the service design process. Their results suggest that their approach led to a refined problem statement and justified design objectives which led to a proactive and successful scope reduction of the information content of the IT service catalogue. In addition, design science has been adopted in the design, development, and evolution of a range of RE approaches, including a perspective-based requirements methods checklist [30], a requirements self-elicitation system [96], an early-stage software quality requirements elicitation process [21], a quality-impact assessment method [41], and a regulator-oriented intelligence framework [1], as well as characterizing the support process and data available to analysts in predicting the risk of escalating support tickets [74], a safety module for the Uni-REPM (Unified Requirements Engineering Process Maturity Model) [114], and a grid-based, service-oriented architecture for portfolio performance measurement [116].

However, Aljafari and Khazanchi [4] highlight that the inability to objectively certify knowledge claims in design science research is a serious communication barrier that hinders the

dissemination of knowledge across the different philosophical perspectives in the IS community and ultimately the advancement of IS knowledge. They argue that it is critical to have a consensus on the meanings of the terms used in designing experiments and presenting the outcomes. Wieringa and Heerkens [119] state that there has been an ongoing debate in the computing sciences about the relative roles and merits of design versus empirical research and about the contribution of design and research methodology to the relevance of research results. Based on a review of the relevant literature, they argue that this is a false dichotomy and demonstrate that while research and design are treated as separate activities, they are complementary, concurrent activities and that the relevance of research results is highly dependent on the problem set rather than solely on rigorous methods. Previous research demonstrates that design science makes a positive theoretical and practical contribution to understanding research challenges that require solutions that must be iteratively proposed, refined, evaluated, and, if necessary, enhanced in the fields of SE, RE, and IS [52].

2.3 Design Science and Sustainability

Looking at research at the intersection of sustainability frameworks and design science, we find no specific SE work. However, design science has a long-established use in the IS domain, and some IS research has focused on explicitly supporting specific sustainability goals using design science. For instance, Klör et al. [60] consider a decision support system that aids the automobile industry with repurposing electric vehicle batteries. Corbett [28] presents a system for influencing individual behavior (e.g., via designing carbon management systems) so that it persuades employees to engage in ecologically responsible behaviors. Sustainable supply chain research is exemplified with a study investigating how country sustainability risk can support individual sustainable supply chain management [95] and sustainable business models [101].

Another focus area in this body of work is the study of specific sustainability dimensions as part of design science and IS. For instance, vom Brocke and Seidel [20] suggest taking environmental sustainability into account when developing an IS using design science research. However, they consider only the “direct environmental impact due to their [products] physical existence . . . and their indirect impact on business processes” [20, p. 301]. De Leoz and Petter [31] seek to raise awareness of IS design science research when considering the direct and indirect social effects of IS artifacts. Most recently, Monson [73] introduced a socially responsible design science methodology encompassing “the objectives of socially inclusive and environmentally sustainable economic growth” (p. 1). This work combines critical research with the problem definition phase of IS design science.

To summarize, to date, the SE and broader (e.g., IS) communities have struggled to address the impact of IT products and services in all five sustainability dimensions and across time, even where design science is used. The present work shares our experience and discusses the lessons learned from SusAF development.

3 OVERVIEW: METHOD AND ARTIFACT

3.1 Method: Design Science Research

Design science research is a research paradigm that generates knowledge and scientific evidence by means of systematically developing an innovative artifact to solve human problems in a particular context [50, 51, 118]. Wieringa [118], for example, applies it to IS and SE research to develop artifacts through two main activities of design and the investigation of the artifact in context. Here the context is composed of the stakeholders and their goals, the existing scientific and engineering theories, known designs and products, researchers’ experience, and common sense.

In this article, the three-cycle view of design science research defined by Hevner [50] has been used (see Figure 4, presented later):

- (1) The *relevance cycle* defines an application context, the requirements, and the acceptance criteria for the research. More specifically, it determines the people and organizational and technical systems that work together toward a goal (application domain), the problems and opportunities in this domain (the research requirements), and how to measure and evaluate whether the artifact is indeed improving the application context (acceptance criteria). The focus of the relevance cycle lies in the application of the artifact (research results) to the relevant environment (field testing). The results of the application allow for an understanding of whether the desired improvements are achieved. Depending on the outcome, additional iterations of the design science cycles might be needed.
- (2) The *rigor cycle* links the design science activities with the knowledge base of scientific foundations. It draws from experiences and expertise in the state of the art, as well as existing artifacts and processes. This cycle seeks to ensure that the research is creating innovation rather than simply applying conventional design and processes to solve problems. However, as Hevner [50] highlights, it is not always possible to base the creative activities of design science on an existing theory, in which case it is advised to use different sources for creative insights.
- (3) The *design cycle* iterates between the construction of the artifact and its evaluation, evolving the artifact based on the evaluation's feedback. It takes as input both the requirements of the relevance cycle and the theories of the rigor cycle. Alternative designs are often tested several times in experimental settings before a satisfactory design is taken to the field with the relevance cycle or additions to the knowledge base are proposed for the rigor cycle.

3.2 Artifact: SusAF

The main artifact developed through the present research is the SusAF. The SusAF is a systems thinking oriented approach that provides interested stakeholders with a supported process for thinking about and expanding their anticipation of the possible sustainability effects of their IT products and services [34, 35].

The SusAF is composed of a set of elements mapped across five sustainability dimensions and a temporal perspective of cross-dimensional interactions. These artifacts are a visualization diagram (**Sustainability Awareness Diagram (SusAD)**), question sheets, usage and drawing instructions, workbook, and taster [13]—all connected via a guiding process. These artifacts are openly available at the Zenodo repository [13].

3.2.1 SusAF Dimensions and Orders of Effects. The SusAF is based on five dimensions [8] of sustainability, which are essentially an extension of the traditional three-dimensional view set out by the Brundtland Commission [120], which notes that environmental, economic, and societal dimensions together make up sustainability. In the SusAF, the environmental and economic dimensions are directly reused. While common frames of sustainable development [120] integrate the concerns of individuals and society into a *social* dimension, we separate these two categories to highlight the IT product's effects on individual persons as well as society as a whole. In addition, in the SusAF, the technical dimension is considered separately, as the framework is equally concerned with the sustainability of the technology itself.

Furthermore, the temporal and aggregated perspective (framed as direct, enabling, and structural impacts) is also considered in the SusAF as the effects that an IT product has upon the sustainability of its environment transpire cumulatively across time.

Table 1. Orders of Effects according to Hilty and Aebischer [54]

Orders of effect	Definition
1st order (also termed <i>direct, immediate</i>)	First-order effects are direct effects of the production, operation, use, and disposal of IT solutions.
2nd order (also termed <i>indirect, enabling</i>)	Second-order effects are linked with the operation and use of an IT-based system and include any change enabled or induced by it.
3rd order (also termed <i>structural, systemic</i>)	Third-order effects consist of structural changes caused by the ongoing operation and use of the IT-based sociotechnical solution.

3.2.2 SusAF Artifacts. Key SusAF artifacts include a moderator’s slide set, a question set, a visualization diagram template (SusAD), and a workbook. For an industrial application, SusAF artifacts also include an analysis report template. In an educational setting, more artifacts are available for guiding students. These are role-playing guidance, case study examples, and SusAD drawing guidance.

The overall *process of the SusAF* is comprised of the following activities. First, familiarize the stakeholders involved with the IT product and its (the product’s) vision as well as the involved stakeholders’ understanding of the currently understood sustainability effects of the IT products. With our industry partner Partneur,² this meant an introduction to their idea for a business model canvas collaboration platform and its currently perceived sustainability effects. Second, use the SusAF question set to engage stakeholders in a discussion of the potential sustainability effects of the system on the five sustainability dimensions. With Partneur, we spent the larger part of a workshop day on in-depth discussions of the questions and the potential effects related to each key topic—for example, the impact of the discoverability of certain project characteristics on the opportunities for network building as well as inclusive and diverse teams (Figure 3). Third, consider cross-impacts of the identified sustainability effects and their cumulative consequences across time and dimensions. With Partneur, we found relations crossing between almost all dimensions—for example, the impact of affordable online learning access (economic) will lead to more users and comprehensive learning (individual), and then, over time, this leads to active educated communities (social). Fourth, represent the main dependencies identified as chains of effects in the visualization diagram (SusAD) to support the discussion. With Partneur, see the example in Figure 3 exemplifying effects across dimensions and orders. Fifth, compare the identified sustainability effects with the original understanding of the stakeholder to see the changes in stakeholders’ awareness. With Partneur, we noted a significant extension in several dimensions of their understanding of potential sustainability effects—for example, the expansion and networking of the communities triggered according to the perceived quality of user experience. Sixth, discuss, with the help of the workbook, how the sustainability effects and chains of effects can represent threats/opportunities, and identify actions that can mitigate/utilize these. With Partneur, one identified major threat was to underestimate the adequateness of the platform for less technically versed prospective entrepreneurs. And finally, seventh, document main findings by creating an analysis report [13].

4 STUDY DESIGN

4.1 Research Design

In the present research, the artifact that was developed using design science research is the SusAF, which provides a solution to the problem of anticipating sustainability effects from IT products

²The start-up Partneur was our first case of industry collaboration, and we use them as a running example to describe the application process in this subsection. The case is presented in more detail later in the article.

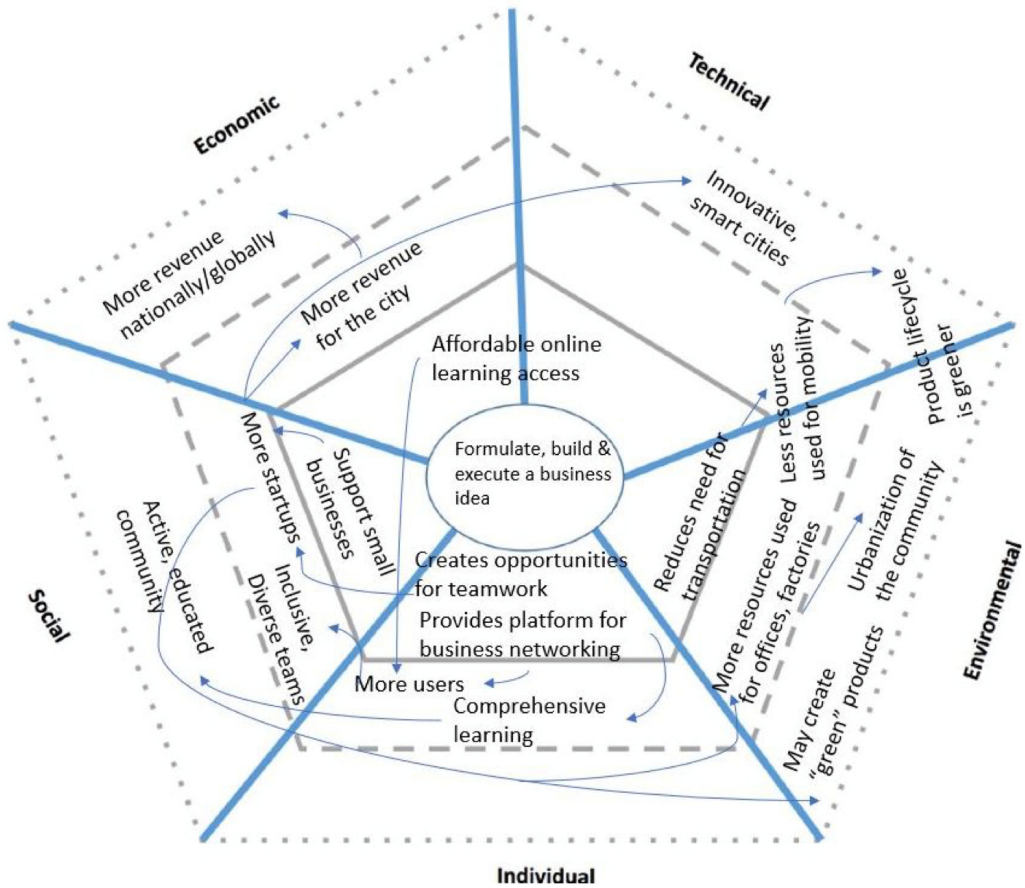


Fig. 3. SusAD: summarizing diagram of the Partneur workshop results.

and services for IT practitioners (software engineers, software architects, etc.) and academia (students and researchers). Figure 4 shows how the three design science research cycles (relevance cycle, rigor cycle, and design cycle) were implemented in this research. A relevance cycle laid the foundation of our work. Following that, the SusAF and its artifacts were incrementally built and evaluated with students during undergraduate and masters’ classes and workshops, as well as industry workshops. Evidence was collected by means of coursework, surveys, and instructor/moderator reflections. Overall, we conducted 13 relevance, design, and rigor cycles, which are discussed in detail in Section 5.³

4.2 Participants

We worked with two main types of participants in relevance cycles: students and IT practitioners.

³We note that this work was carried out as an actual design science project. In other words, it started with a problem identification (i.e., lack of consideration of sustainability in SE process and practice), to which we set out to develop a solution. This solution was evaluated and improved iteratively over several years, with feedback from evaluation informing researchers about what needed to be done next. In short, there was no master plan defined at the start of the project—instead, an open-ended iteration of artifact design-evaluation-improvement was carried out, which is the very spirit of the design science approach.

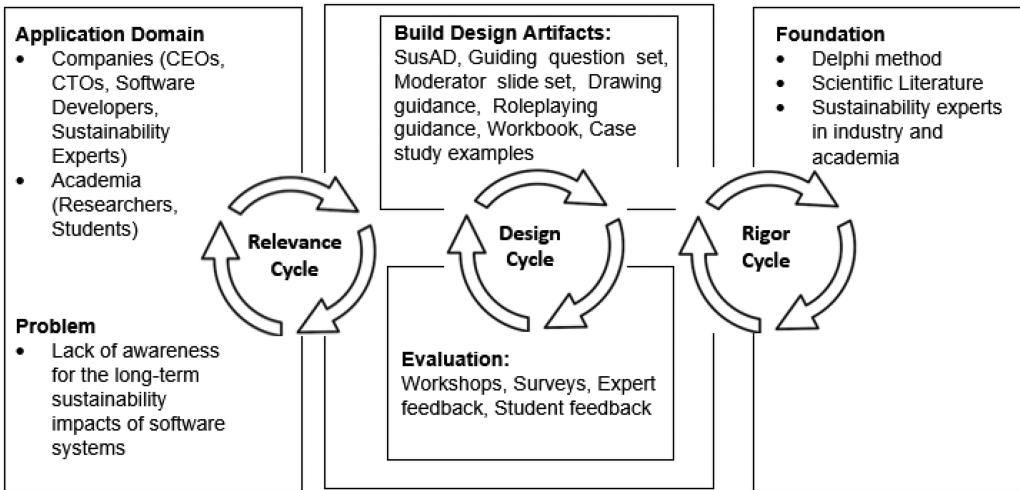


Fig. 4. Application of the design science research cycles to the development of the SusAF, adapted from Hevner and Chatterjee [51].

Overall, we had 208 students in several countries—the United States (CSULB), Finland (LUT), Spain (LaSalle), Germany (HFU), and Sweden (KTH, LiU):

- The students were all in IT-related subject areas, some of them in bachelor’s programs and others in master’s programs. Some of these students were IT professionals with several years of experience. For each implementation of the course, we used the SusAF within a course and collected the created artifacts (SusADs and reports), as well as, in most cases, specific feedback surveys. We also collected instructor reflections. The SusAF was used both on newly to be developed IT products and for the evaluation and evolution of existing products.
- The 64 IT practitioners who participated in our study belonged to 43 companies from Finland, India, the United States, Sweden, Germany, and Spain. For the industry participants, we had access to the developed artifacts (SusADs and reports) and solicited feedback via surveys. In addition, we collected feedback from workshop moderators.

The place, year, participants, timings, relevance cycle, and data collection are depicted in detail in Table 2. We analyzed the data collected separately for every cycle to identify improvements and lessons learned.

4.3 Design Science Cycles Overall

As we progressed through the design science cycles, we developed the different artifacts that compose the SusAF. Some artifacts were targeted at teaching, others at industry, and the remaining at both contexts, as depicted in Figure 5. While the guidance would also be relevant for industry, they are depicted only as teaching artifacts because in the previous industry cycles the researchers performed the analysis and reported back to the companies.

Overall, the development of the SusAF evolved along 13 cycles (Figure 6). The development started from identifying the need to educate IT students and professionals on the potential effects of IT products on sustainability and provide them with tools to make these effects explicit (cycle 1). We iterate between rigor, design, and relevance cycles, where the first two were normally intertwined. Since most of the researchers involved in the design of the SusAF were also university academics, we chose to start developing and applying the solution in the context of university

Table 2. Participants

ID	Location	Year	Participants	Timings	Relevancecycle
1	Interviews, 8 countries	2016	13 RE practitioners Edu level: mixed Background: CS, requirements engineers	Course duration: N/A No. of classes: N/A Time to deliver work: N/A	cycle 1
2	LUT, summer school, Finland	2017	13 students (4 groups) Edu level: BSc, PhD Background: CS, business & marketing, industrial engineering, environmental engineering	Course duration: 1 week No. of classes: 5 Time to deliver work: 1 week	cycle 3
3	LUT, Finland	2017 /2018	21 students (individual) Edu level: BSc, MSc Background: CS, SE	Course duration: N/A. No. of classes: N/A Time to deliver work: 20 weeks (part of thesis)	cycle 3
4	CSULB, USA	2018	26 students (8 groups) Edu level: BSc Background: CS	Course duration: 15 weeks No. of classes: 30 Time to deliver work: 2 weeks	cycle 3
5	HFU, Germany	2018	5 students (2 groups) Edu: BSc Background: IT, business information	Course duration: 15 weeks No. of classes: 30 Time to deliver work: 1 week	cycle 3
6	HFU, Germany	2018 /2019	6 students (3 groups) Edu: BSc Background: IT, business information	Course duration: 15 weeks No. of classes: 30 Time to deliver work: 1 week	cycle 5
7	LUT, Finland	2018	31 students (10 groups) Edu level: BSc, MSc Background: CS, SE	Course duration: 10 weeks No. of classes: 12 Time to deliver work: 2 weeks	cycle 5
8	LUT, Finland	2019a	16 students (individual) Edu level: BSc, MSc Background: CS, SE	Course duration: N/A No. of classes: N/A Time to deliver work: 20 weeks (part of thesis)	cycle 5
9	CSULB, USA	2019	28 students (9 groups) Edu level: 2nd/3rd year Background: CS, 1 environmental engineer	Course duration: 15 weeks No. of classes: 30 Time to deliver work: 2 weeks	cycle 7
10	LUT, Finland	2019b	14 students (7 pairs) Edu level: BSc, MSc Background: CS, SE, environmental engineering	Course duration: 7 weeks No. of classes: 5 Time to deliver work: 3 weeks	cycle 7
11	LUT, Finland	2019c	20 students (6 groups) Edu level: BSc, MSc students Background: CS, SE, business administration	Course duration: 14 weeks No. of classes: 6 Time to deliver work: 3 weeks	cycle 7
12	Partneur, USA	2019	5 IT practitioners (1 group) Edu level: BSc, MSc Background: 3 business, 2 technical	No. of workshops: 1 Duration of workshop: 4 hours Reporting meetings: none Reporting duration: N/A	cycle 8
13	Data matrix, India	2019	3 IT practitioners (1 group) Edu level: MSc, PhD Background: technical, CS	No. of workshops: 1 Duration of workshop: 3 hours Reporting meetings: 1 Reporting duration: 1.5 hours	cycle 8
14	Visma, Finland	2019	2 IT practitioners (1 group) Edu level: MSc Background: CS	No. of workshops: 1 Duration of workshop: 3 hours Reporting meetings: 1 Reporting duration: 1 hour	cycle 8
15	Premier Park, Finland	2019	1 IT practitioner (1 group) Edu level: N/A Background: racing	No. of workshops: 1 Duration of workshop: 6 hours Reporting meetings: 1 Reporting duration: 1 hour	cycle 10
16	Jeppesen, Sweden	2020	4 IT practitioners (1 group) Edu level: MSc Background: engineering	No. of workshops: 1 Duration of workshop: 4 hours Reporting meetings: 1 Duration reporting: 2 hours	cycle 10
17	LaSalle, Spain	2020	19 students (4 groups) Edu level: MSc Background: engineering	Course duration: 8 weeks No. of classes: 8 Time to deliver work: 2 weeks	cycle 12
18	KTH Sweden	2020	9 students (4 groups) Edu level: MSc Background: engineering, CS, sustainability	Course duration: 15 weeks No. of classes: 1 Time to deliver work: day of class	cycle 12
19	HFU, DigiHubs, Germany	2020	49 IT practitioners (individual) Edu level: N/A Background: engineering	Course duration: 90 minutes No. of classes: 3 Time to deliver work: within course	cycle 13

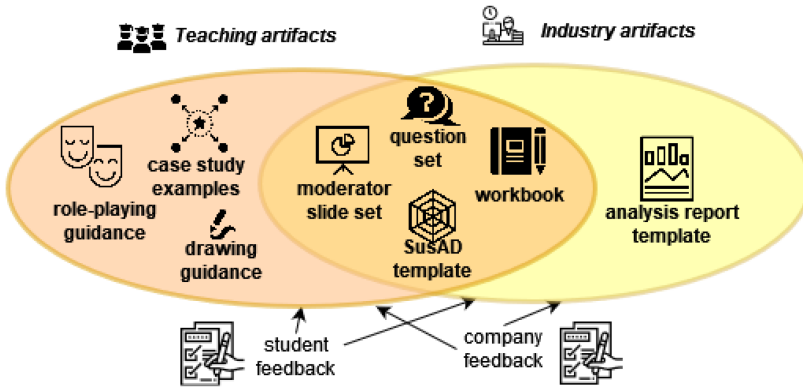


Fig. 5. Teaching and industry artifacts.

teaching (cycles 2–7). Once we felt that the framework was practical and useful in teaching, we extended its application to companies. Therefore, the remaining cycles (8–13) tested the improvements of the framework in both teaching and industry contexts.

5 STUDY EXECUTION

We next explain the work carried out in each one of the cycles depicted in Figure 6, describing how we created, evaluated, and improved SusAF artifacts and why we moved from one cycle to another. The cycles are described as follows.

Relevance cycles discuss the following:

- *Artifacts to be validated*: The artifacts under evaluation, if any;
- *Participants*: Participants in the relevance cycle, if any;
- *Data collected*: The data collected to evaluate the artifacts;
- *Evaluation*: How the artifacts were evaluated and main results.

The first three elements are summarized in a small table in the beginning of each relevant cycle.

Rigor and design cycles discuss the following:

- *State of the art*: The knowledge which has been considered to solve the general challenge in the context of the SusAF.
- *Designed artifacts*: How the new artifacts are created or existing artifacts improved. For simplicity, we will not discuss alternative designs and their evaluation in the design cycles, focusing only on the artifacts taken to the relevance cycles.

The lessons learned in each relevance cycle are highlighted, as these are likely to be relevant to the broader research community.

5.1 [Relevance] Cycle 1: Identified Need: Interview Study

Artifact to be validated	Participants	Data collected
None	Researchers and IT practitioners	None

Evaluation. This cycle identified the need for SusAF. It was inspired by our readings and observations as SE researchers and confirmed by an interview study. In both our teaching and industry

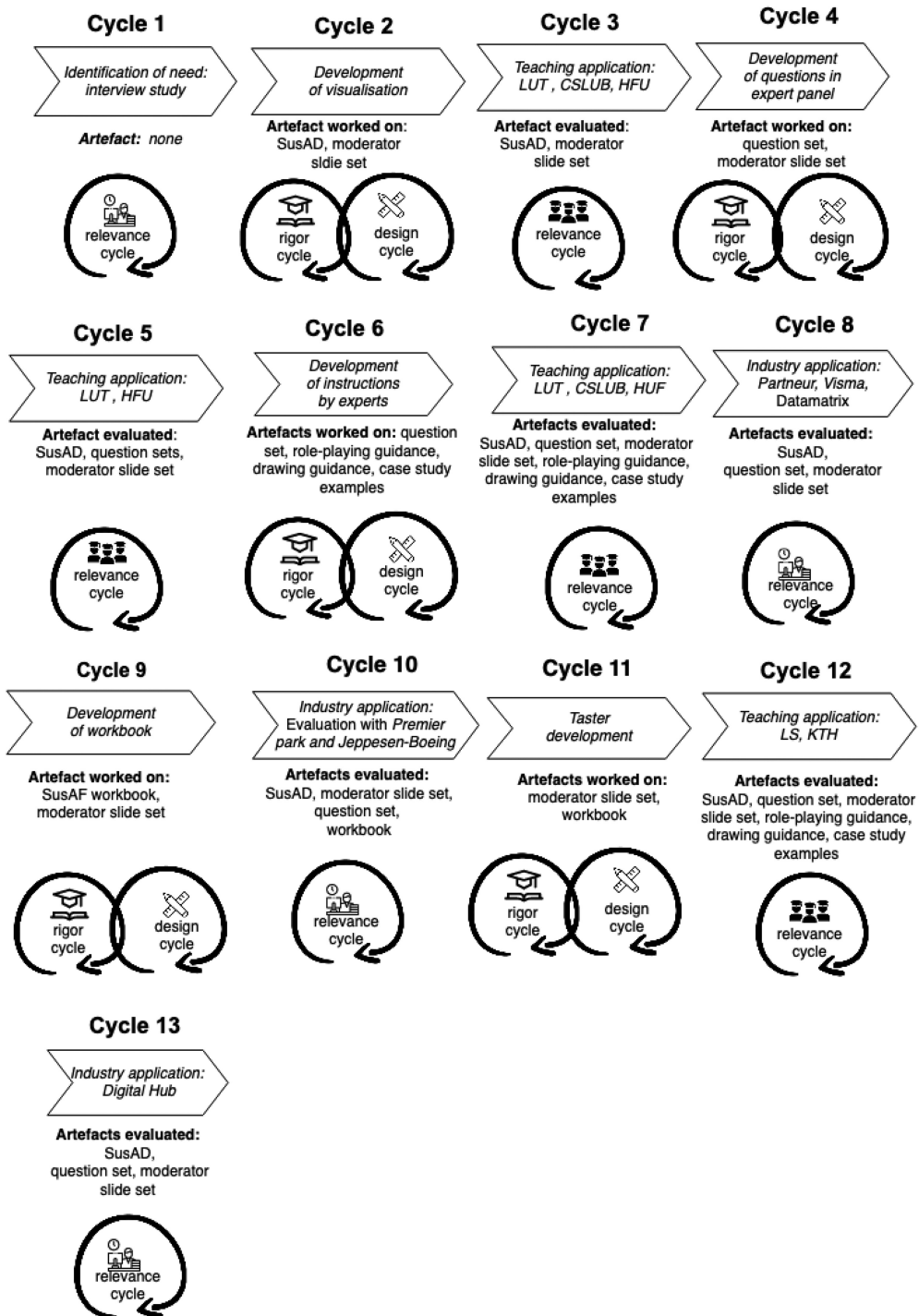


Fig. 6. The design cycles we applied for the SusAF.

collaborations, we had observed that students and IT practitioners lacked an understanding of sustainability and its relationship with IT products. Our observations corroborated other studies (e.g., [45, 48, 58]).

To validate these observations, we carried out a qualitative interview study with 13 RE practitioners from eight countries (Austria, Brazil, Germany, Spain, Switzerland, Turkey, the UK, and the United States) and explored their perceptions and attitudes toward sustainability [24]. In particular, we inquired about their understanding of sustainability, their awareness of the effects that IT products can have on the different sustainability dimensions, and the obstacles and mitigation strategies for the application of sustainable design principles in daily work life. The results revealed that participants felt they lack knowledge, experience, tools, and methodologies to integrate sustainability into their everyday practice, as demonstrated by the following quote:

There must be much much more information and techniques and methods available in order to help the developers, REs, project managers and usability engineers to identify the issues they have to look at when they are trying to realize a sustainable ERP system. [24, Interview N1]

Lessons Learned—L1, On the Need for Methods and Tools: IT practitioners and students (future IT practitioners) need methods, guidelines, and tools to help them consider the potential sustainability effects of novel and existing IT products. More specifically, IT practitioners need to be more aware of (i) the nature of sustainability, (ii) the interdependencies between sustainability and IT products, and (iii) the possible effects that IT products can have over time on different sustainability dimensions.

5.2 [Rigor and Design] Cycle 2: Development of Visualization

State of the Art. In 2015, the Karlskrona Manifesto laid the foundation for this work, by providing a set of principles for sustainability design in SE, yet it did not propose any supporting tools [8]. Easterbrook [36] detected that software engineers and computer scientists, in general, are prone to learning much about computational thinking (i.e., how to divide and conquer challenges) but little about systems thinking (i.e., looking at the integrated big picture), which is crucial for addressing such (wicked) challenges as sustainability. Similarly, Man et al. [67] and Mann and Smith [68] pointed out the need for computing to look at sustainability in a more holistic way.

With respect to how to enable this holistic vision, cognitive sciences confirm that a central visual artifact to summarize insights and results helps understanding and knowledge retention [122]. Thus, one way to provide the initial starting point for such systems' analysis is through visual representation (e.g., via rich picture [2] for a sociotechnical system, or a high-level data flow diagram [92]). Probably the best-known visualization for sustainability effects at the time was the Flourishing Business Canvas model [110], but it focuses on depicting an overview of a business model instead of on the potential effects such a system could cause.

Moreover, there is a class of literature dealing with "Education for Sustainable Development" for models which are useful in teaching about sustainability, but they are not targeted at IT products [66, 93, 109].

Designed Artifact. Based on the preceding literature, the first developed artifact of the SusAF was a central visualization diagram to discover, document, and validate potential sustainability effects [7]. The diagram (an example of which is shown in Figure 3) gives a representation of chains of such effects across five sustainability dimensions and timelines (represented by) order of effects. The diagram was named *SusAD*.

Finally, to support teaching, we also developed a *set of moderator slides*. The slides explained the main concepts of sustainability, their relationship with IT products, and the purpose and use of the SusAD.

5.3 [Relevance] Cycle 3: Teaching Application: LUT, CSULB, and HFU

Artifact to be validated	Participants	Data collected
SusAD Moderator slide set	Researchers Students (LUT, CSULB, HFU)	SusAD instantiations ^a Reports (or thesis works) Facilitator reflection

^aThe SusADs produced by participants.

Evaluation. The second relevance cycle focused on the SusAD, and was performed through teaching in the United States, Finland, and Germany (see Table 2). The experiment consisted of using the SusAD to document the potential effects of existing IT products.

The classroom-based evaluation demonstrated that the visualization tool helped students grasp the key notions of sustainability dimensions and orders of effects at the conceptual level. Due to the visualization, they were able to better understand the tradeoffs between the different dimensions and how one effect can cause other impacts. Additionally, the moderator slide sets helped the students better understand the effects of IT products on sustainability and generate theoretical knowledge on that subject. Details of this evaluation were presented in the work of Duboc et al. [35].

Based on the instructors' feedback and qualitative feedback from the participating students, we observed that the notion of sustainability effects remained somewhat abstract to students. They did not immediately see how to apply these notions to the context of their IT products: "The questions did not fit our system" (German student, quote from survey).

They also often simply reused the example sustainability effects given by the instructors or limited exploration to the effects that reinforced the IT products' purposes. However, the students were not sure as to how to initiate the process of sustainability requirements elicitation [35].

Lessons Learned—L2, On the Need for Dimensional and Temporal Link Up:

IT students need help to (i) link up the abstract notion of sustainability and its dimensions with the specific domain of the sociotechnical system they are working on at the given time, and (ii) link up the potential structural changes from long-term use of the system (which could become notable in a few years) with development decisions they could take at the system development time.

This help is relevant for IT practitioners as well.

5.4 [Rigor and Design] Cycle 4: Development of Questions with Expert Panel

State of the Art. Turning to the literature, we observe that sustainability is also frequently misinterpreted as having to do only with the environmental dimension [25] rather than encompassing the systemic and multidimensional nature of it; however, IT practitioners do not commonly deal with effects that may unfold over time [8]. In addition, when dealing with the complex systems-related information and analyzing systems effects on sustainability, often abstract guidance is proposed [23, 103]. No empirically validated results of such effects or hands-on tutorials to help gather the relevant information and identify possible effects were found at the time.

Furthermore, we noted that where empirical data was available (for other than sustainability-specific RE contexts), the researchers had aggregated empirical results for frequently analyzed

problems into patterns (goal decomposition patterns for security, usability, etc. [121]) or using the GQM (goal-question-metrics) method to break down complex goals into measurable and operationalizable metrics. Moreover, where the empirical data was rather limited, researchers have employed expert panels to help elicit requirements and show that the expert knowledge reliably relates to the reality of the problem domain [9]—for example, from predicting requirements defects [64] to estimating software development effort [59].

The “scenario techniques” widely used in future studies [16, 17] were also identified as a relevant method to help analyze and coherently present the possible future effects. This technique is often used for modeling best case, worst case, and middle or conditional scenarios.

The scenarios technique informed design of the *extreme scenario artifact* for the SusAF. This is a simplified scenario focusing on large-scale long-term use of the intended IT product.

Designed Artifacts. Based on the preceding literature, we designed the *questions set* used in the SusAF.

Given that at the time we had neither empirically validated patterns for sustainability goal decomposition nor a clear perspective on what sustainability must look like for diverse sociotechnical systems, we chose to enlist the help of experts. The experts were the members of the Karlskrona Alliance on Sustainability Design [8], who had worked on sustainability topics for a decade and had investigated various application domains, such as energy, food security, and smart cities. More specifically, we used an adaptation of the Delphi method [63, 77] to get these experts to define a set of questions to help start and guide the discussion on the effects of IT products on their *in situ* sustainability [35].

Additionally, we utilized the current body of knowledge as a further input to the question set, including the Flourishing Business Canvas model [107], the Schwartz model of human values [104], the visualization diagram [8], the Sustainability Goal Reference model [87], and SIA (Societal Impact Assessment) literature. The derivation of questions was carried out in three rounds (as detailed in the work of Duboc et al. [35]):

- (1) In the first (initiation) round, the panel facilitator set out an online document and invited panel members to contribute views on factors that affect the five dimensions of sustainability, and questions that a requirements engineer should consider regarding these factors.
- (2) At the second (review) round, the panel was requested to (asynchronously and in writing) review and comment on all of the results of the first round. This resulted in a number of issues raised with regard to previously expressed views/proposed questions.
- (3) During the third (consensus) round, the panelists reflected on the written feedback given by others and reviewed their inputs. Thereafter, any remaining questions and issues were resolved through online small group meetings, where two to four panelists met to discuss the concerns raised.

This process resulted in identification of five topics per sustainability dimension, considered to be the most relevant by the panel. Each topic was accompanied with a list of questions to initiate discussion and requirements elicitation for it.

These topics are listed in Table 3, and an example of the first set of questions for the social dimension is shown in Figure 7.

While the questions were intended to help start a discussion on the sustainability dimensions, the challenge of representing the systemic (long-term) effects of the IT product on its larger sociotechnical system remained. For this, we turned to future visioning scenario description techniques, providing the students with an “extreme scenario” where students were told to imagine that the system had been a big success and it has been used for a long time and by a very large

Table 3. Topics in Each Dimension

Dimension	Topics
Social	Sense of Community; Trust; Inclusiveness & Diversity; Equality; Participation & Communication
Individual	Health; Lifelong Learning; Privacy; Safety; Agency
Environmental	Materials and Resources; Soil, Atmospheric and Water Pollution; Energy; Biodiversity and Land Use; Logistics and Transportation
Economic	Value; Customer Relationship Management; Supply Chain; Governance and Processes; Innovation and R&D
Technical	Maintainability; Usability; Extensibility and Adaptability; Security; Scalability

SOCIAL DIMENSION	
SENSE OF COMMUNITY <input type="checkbox"/> Often people belong to an organization, to an area or to a group of like-minded people. Can the system affect a person's sense of belonging to these groups?	PARTICIPATION AND COMMUNICATION <input type="checkbox"/> Can the system change the way people participate in an organization or other social groups? <input type="checkbox"/> Does it affect the way people communicate verbally and non-verbally? <input type="checkbox"/> Does it affect the way people create networks? <input type="checkbox"/> Does it affect the way people form bounds? <input type="checkbox"/> Does it affect the way the effort people put in a group work? <input type="checkbox"/> Does it affect the actions people take to achieve the goals, projects and tasks of a group? <input type="checkbox"/> Does it affect the way people engage with others? <input type="checkbox"/> Does it affect the way people support, consider, critique or argue with others?
TRUST <input type="checkbox"/> Can the system change the trust between the users and the business that owns the system? <input type="checkbox"/> What about the trust between the users themselves?	
INCLUSIVENESS AND DIVERSITY <input type="checkbox"/> Can the system impact on how people perceive others? <input type="checkbox"/> Does the system include uses with different backgrounds, age groups, education levels, or other differences? <input type="checkbox"/> Does the system caters for these differences? How?	
EQUALITY <input type="checkbox"/> Can the system make people to be treated differently from each other? For example, because the system carries out data analytics or influences human decisions.	
Is there any other issue that is relevant to the society that the system may affect?	

Fig. 7. Question sheet for the social dimension.

number of users, and to consider what effects will that long-term continuous and large-scale use have on the sociotechnical environment of the system?

5.5 [Relevance] Cycle 5: Teaching Application: LUT, HFU

Artifact to be validated	Participants	Data collected
SusADs Questions sheet	Researchers Students (LUT and HFU)	Students feedback form ^a SusAD instantiations Reports

^aExcept LUT 2019a.

Evaluation. The new version of the SusAF (now including the questions) was used in the teaching of three courses in Finland and Germany (see Table 2). To evaluate the new artifacts, the survey was expanded to address the question set. The students applied the framework on their course projects and were instructed to carry out interviews with experts on the project domain and sustainability dimensions, using the question set provided. They were also told to explore the future effects by asking interviewees to consider the “extreme scenario” with respect to each question.

Students reported that the new version of the SusAF was easy to understand and useful, and that it supported interesting and structured discussions by providing them with new perspectives. For example, the students noted that the SusAF

“[gave] different points of views and new perspectives,”and
 “gave structured analysis of the important challenges concerning sustainability.”

The students also reported that the question set helped them to identify additional effects: “new ideas/opinions I haven’t thought of yet” (student from Finland, survey).

These opinions were confirmed by the SusADs that students produced: compared to the SusADs produced by students from the previous evaluation cycle, these had a significantly higher number of elicited impact effects and the identified effects were also better contextualized. Moreover, the chains of effects (i.e., where one effect causes another) also increased in number and length. It is interesting to highlight that there was an 80% increase in the potential effects identified in LUT diagrams from the academic year 2017–18 to 2018–19 [34, 35]. In particular, we noted that students were better able to clearly relate their previous knowledge on sustainability to the software systems development.

However, the students’ feedback revealed several shortcomings:

- the questions were found to be phrased “too academically and abstractly;”
- students were inexperienced and did not know how to use the questions effectively in an interview context (e.g., “They had to be discussed to understand what is being asked”), and
- asking about the “extreme scenario” after each question was too tedious and repetitive (teacher feedback).

Additionally, a new teaching assistant joined the module delivery team and asked for instructions on how to draw the SusAD.

Lessons Learned—L3, On the Need for Examples: To be practically usable, the methods and artifacts that help analyze sustainability impacts of a sociotechnical solution on its situated environment need to be accompanied with a set of application guidelines and demonstrative application examples.

5.6 [Rigor and Design] Cycle 6: Development of Instructions and Examples with an Expert Panel

State of the Art. In the literature, Ouhbi and Pombo [80] report that role-playing and problem-based learning are among the most effective ways of teaching SE. Along the same line, Anastasiadis et al. [6] identified learning by doing, applying critical thinking, and fostering real-world engagement as crucial factors for teaching sustainability in its complexity. Wamsler [115] argues that a shortcoming of many sustainability education approaches is (i) the neglect of inner dimensions and capacities, and (ii) a limited capacity to facilitate reflection on the cognitive and socioemotional processes underpinning people’s learning and decision making. She concludes that more integral approaches and pedagogies are urgently needed that include adapting contemplative interventions. These research findings motivated the design direction of the artifact application guidelines and examples.

Designed Artifact. Considering the preceding literature and utilizing the expert panel once more, we improved SusAF by doing the following:

- (1) Developing instructions for drawing the diagrams and suggesting alternative designs for the diagrams, depending on the dimensions one wants to emphasize or explicitly reflect on (Figure 8);
- (2) Adding detailed examples for two case studies (namely, for a procurement system and Airbnb) into the teaching materials to ground the explanation of the process, making it less abstract, and familiarizing the students with its application process;
- (3) Adding a brief (10-minute) role-play into the delivery of the SusAF class, where the teacher and an assistant would simulate an interview by a requirement analyst with a system or domain expert; and

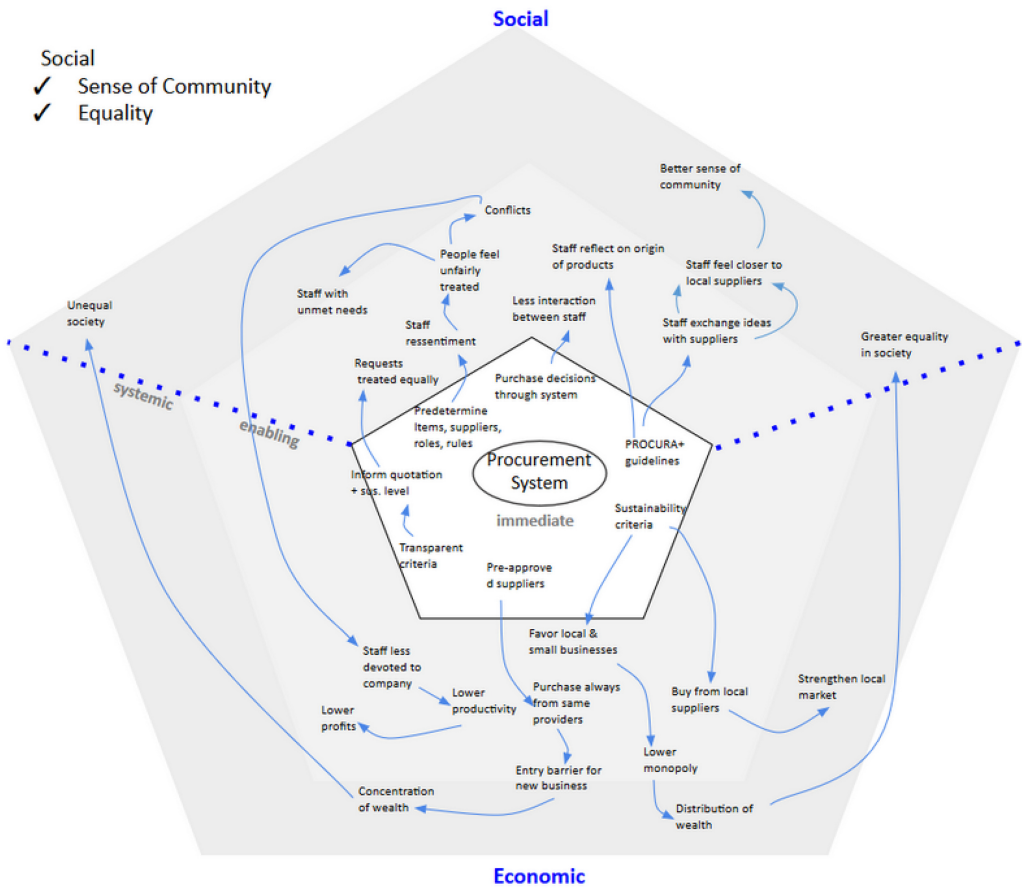


Fig. 8. Alternative way to draw the SusAD (1).

(4) Refining the questions via another round of the expert panel review to make them easier to understand.

5.7 [Relevance] Cycle 7: Teaching Application: LUT, CSULB, and HFU

Artifact to be validated	Participants	Data collected
SusAD Question sheets Drawing instructions Example: Procurement Example: Airbnb	Researchers Students (LUT and CSULB)	Students' feedback form (V2) SusAD instantiations Reports ^a

^aExcept LUT 2019b.

Evaluation. We carried out the evaluation of the new and improved artifacts, once again, through teaching four courses in Finland, the United States, and Germany (see Table 2). The revised questions, drawing instructions, and role-play were integrated into the classes. A survey to collect students' feedback was, once again, administered. The collected assessment submissions and survey data demonstrated that both the quality of the artifacts and the feedback provided by the students had improved—for example, the number of identified effects had risen [35] and the instructors received hardly any requests for clarification.

Moreover, the feedback suggested that the SusAF provided insightful discussions and that the proposed approach was applicable in practice [34, 35].

Therefore, the research team concluded that the SusAF was ready for the relevance cycle with practitioners: to be applied to the problem of lack of awareness for the long-term sustainability effects of IT in the industrial setting (see Table 2).

5.8 [Relevance] Cycle 8: Industry Application: Partneur, Visma, and Datamatrix

Artifact to be validated	Participants	Data collected
SusAD Questions sheets Process Report ^a Presentation	Researchers Representatives of Partneur, Datamatrix, and Visma	Company feedback form ^b SusAD instantiations

^aProcess and reports did not exist at the time of Partneur.

^bFor Partneur, the form (V1) did not have questions about the report, and for Visma and Datamatrix, it did (V2). Visma also gave us oral feedback.

Evaluation. In this cycle, we evaluated the SusAF with three industry partners:

- The first evaluation workshop was carried out with *Partneur*, a start-up company in California. Partneur is a company that supports other start-ups in developing their business plans through a dedicated technology platform. Their open collaboration platform follows the Flourishing Business Canvas model approach [79] and has been rolled out globally, but the marketing effort was mainly targeting the United States at the time of this study [82]. During the workshop with Partneur, the SusAF with its question sheets was used and the feedback was collected through a survey.

The workshop was further repeated with two other companies: Datamatrix and Visma.

- *Datamatrix* is an Indian company that focuses on energy-efficient water pump installations for agricultural irrigation. This includes monitoring and optimization, asset health care and management, and energy conservation.
- *Visma* is a Norway-based company providing software systems that simplify and digitize core business processes in the private and public sector. Among its solutions, Visma offers software solutions for automating accounting processes.

In all workshops, three members of the research team met with companies’ representatives to elicit the effects of their system upon sustainability. All workshops started with an explanation of the SusAF, a quick discussion about the company’s existing perception of the effect of their systems on sustainability, then moving into a moderated group discussion guided by the SusAF question sets. The workshops uncovered a considerable number of potential system effects distributed across all dimensions. Post-workshop, we summarized the discussions, captured the main chains of effects in the SusADs, and identified threats and opportunities for each company using the analysis report template. The results were presented to the companies, and detailed analysis reports were handed into Datamatrix and Visma. The feedback from companies was collected via a short survey. The companies said that the workshop changed their perception of sustainability, bringing insights from perspectives that they had not considered previously. For instance, companies’ representatives said that they had gained

“large amount of insights [and] a new way of understanding how our product can affect the society” (Datamatrix), “a systematic documented approach to sustainability” (Visma), and a way “to really look at the long-term effects of using the platform” (Partneur).

This impression was confirmed by comparing their initial awareness of sustainability effects (Table 4, row 1) with the total number of effects (Table 4, row 2) discovered during the workshop.

Table 4. Metrics on Effects and Chains in Reports/SusADs for Partneur, Visma, and Datamatrix

	Partneur	Visma	Datamatrix
Effects Known before SusAF	N/A	3	4
Total Number of Effects	N/A ⁴	137	153
Number of Effects per Dimension	N/A	7–48	13–51
Number of Effects in SusADs	25	37	34
Number of Chains Crossing Dimensions in SusADs	4	15	25
Number of Chains Crossing Order of Effects in SusADs	12	14	13

For example, Datamatrix’s original perceptions of the sustainability effects of their product was limited to four aspects before the workshop: technical solution for resource monitoring (technical), improved management of water resources (environmental), transparency and communication between stakeholders (social), and subsidy-free farming—that is, economic independence of farmers (economic). During and after the workshop, their perception was broadened significantly to 153 potential sustainability effects identified in the report. Some of the newly identified effects included, for example, enabling new ways for community engagement and responsible sharing, extra clean energy production, and improved drinking water quality.

Additionally, all workshop participants considered that the value they got out of the workshop was worth the time spent. All three companies also reported the intention to take action based on the results of the SusAF and to repeat this exercise in 6 to 12 months. All said they would recommend the SusAF to collaborators, and a handful planned to repeat the analysis on a different company product (see Open Data Package [83]).

While preparing the post-workshop analysis reports, it was observed that it would be helpful to have a workbook where notes could be taken in a structured format throughout the elicitation and analysis process. This issue would become even more relevant when the practitioners start to carry out the workshops by themselves. Not only note taking needed to be better supported, but so did the post-workshop analysis process itself. The analysis process applied in the present evaluation cycle was quite time consuming for the researchers and would not be affordable within a practice setting. Thus, a need to visualize and formalize the data collection and analysis process was identified. The practicality and applicability of research artifacts are common concerns in technology transfer (as discussed in the following subsection).

Lessons Learned—L4, On the Efficiency of Elicitation and Analysis:

Data organization and analysis need to be supported by guidelines which can be applied by practitioners without external help. Support for both full-scale and partial application of the analysis framework is needed. This is because companies may be more interested in certain sustainability dimensions due to their domain or priorities. In particular, focusing on specific effects of dimensions, at a given time, can be critical to some company objectives or priorities.

5.9 [Rigor and Design] Cycle 9: Development of the Workbook

State of the Art. Mazurkiewicz and Poteralska [69] observed that among the main barriers to technology transfer are companies’ focus on easily implemented technologies, difficulties in transmitting technical information from R&D organizations to the technology users, a high-level of tacit knowledge hindering the transference of technology, and problems concerning intellectual property rights over the solution.

Diebold and Vetro [32] carried out a survey with industrial and academic partners on two large research projects in Germany. They observed that transfer mediums are too often human intensive

(e.g., personnel exchanges and meetings) rather than artifact based (e.g., tools, guidelines, and publications), yet it is the guidelines that are more often used by the industry partners [33]. This observation is supported by other studies on IT transfer. For example, a study of 608 IT organizations moving into object-oriented programming language concluded that IT practitioners were willing to try new technologies when knowledge could be acquired easily and cheaply, recommending proper packaging and support [40]. Gorschek et al. [47] propose a seven-step technology transfer mode for ensuring close cooperation and collaboration between researchers and IT practitioners. They note that the user documentation is the most important factor in success of step 7: solution release, in which the documentation should ideally be in form of a few-page quick-reference guide. Finally, Heuer et al. [49] suggest the use of structured guidelines for technology transfer, along with recommendations on writing such artifacts.

Designed Artifacts. Inspired by the preceding guidelines, we developed a workbook and an industry-tailored version of the moderator slide set to document and guide the process of identifying the potential effects of IT products on sustainability. The workbook was developed through an iterative design process of five rounds within the team with one group designing and the other group providing iterative feedback. Figure 9 shows a thumbnail of a few pages from the workbook. It is now released for independent use and replication under the Creative Commons Attribution open source license [13].

5.10 [Relevance] Cycle 10: Evaluation with Premier Park and Jeppesen-Boeing

Artifact to be validated	Participants	Data collected
Workbook with: SusAD	Researchers Students Stakeholders from	Company feedback ^a
Questions sheets Process Report	Premier Park and Jeppesen-Boeing	SusAD instantiations

^aFree-form for Premier Park and form for Jeppesen-Boeing.

Evaluation. Once deemed ready, the SusAF along with the workbook were evaluated in a relevance cycle with two more companies: Premier Park and Jeppesen-Boeing.

Premier Park is a Finnish driver training and conference center company near Helsinki. Jeppesen-Boeing is a company offering flight, crew, and airport systems.

Both evaluations were carried out via case study analysis workshops. Before commencing the workshops, we asked the company representatives to summarize their current perceptions of the sustainability effects that the IT products under consideration could have. After the workshop, we counted the relevant metrics from the workbook notes, including the number of total effects, effects per dimension, the cross-dimensional impacts, and chains of effects across time and dimensions. The summary results of pre-and post workshop data are presented in Table 5. We clearly observe that through the workshop, both companies have substantially (more than 10-fold) broadened their perception of the potential sustainability effects of their products.

Premier Park's workshop was attended by three researchers, four students, and one company representative. The analysis focused on the potential sustainability effects of a new product the company was envisioning. After the workshop, a workbook-based report was prepared and presented to the company. The researchers allowed for a 6-month reflection period, for the company to review and use the report as they wished. After the 6-month period, a feedback session was arranged to see whether the report had led to further consideration or action. The company representative stated that the sustainability analysis and report were "clear and it's pretty easy to continue preparations after that and based on that" (Premier Park).

The company was using the report as a basis for planning the new product and searching for partners to implement the project with sustainability impacts in mind.

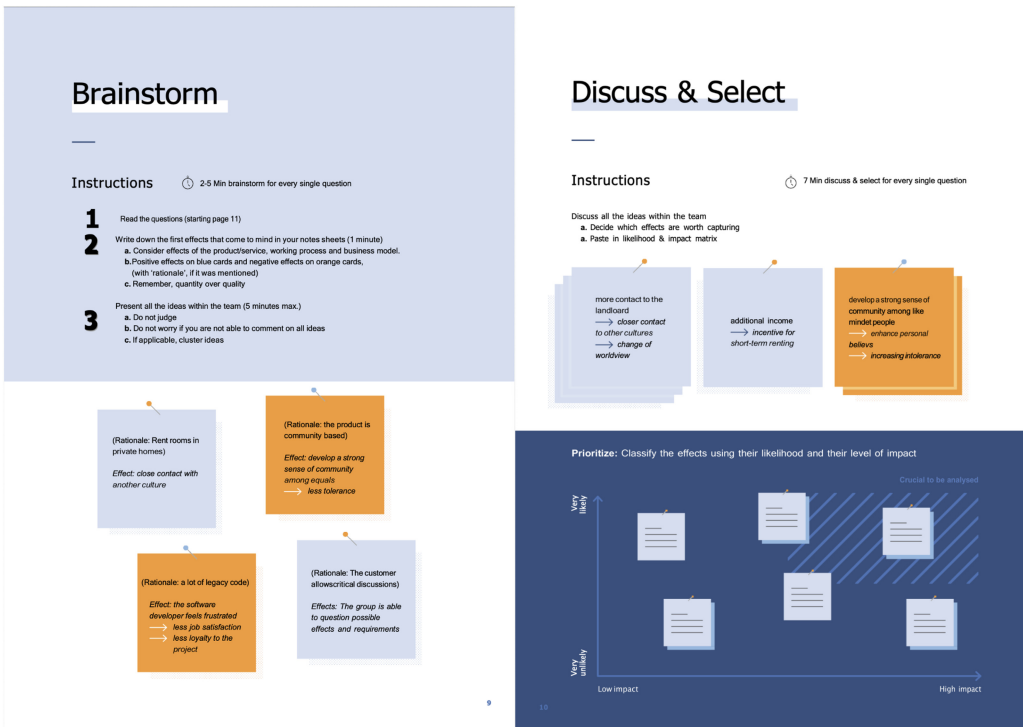


Fig. 9. Two pages from the workbook as thumbnails. The workbook is available for free [13].

Table 5. Metrics on Effects and Chains in Reports/SusADs for Jeppesen Boeing and Premier Park

Metrics	Jeppesen Boeing	Premier Park
Effects known before the SusAF	7	2
Total number of effects with the SusAF	72	144
Number effects per dimension	9–21	20–35
Number effects in the SusADs	31	25
Number of chains crossing dimensions in the SusADs	25	13
Number of chains crossing orders of effects in the SusADs	25	10

Jeppesen-Boeing held a half-day-long workshop attended by four stakeholders and a facilitating researcher. The subsequent workbook-based notes and analysis completed by the researchers were submitted to the company and feedback requested. The company representatives said that this gave them “new useful insights,” and they wanted to continue this line of work in follow-up studies.

However, during the post-workshop debrief, the companies’ representatives noted that a half- or full-day workshop for applying the SusAF is not always within the time budget of a company, especially if the company is not fully convinced of the approach. Therefore, an additional introductory format is desirable, which could be used before a client company is committed to the full extent.

Lessons Learned—L5, On the Introductory Format: Companies and students may lack the time and/or the conviction necessary to carry out a full application of the SusAF when they are first introduced to it. Thus, an “Introduction to” format of SusAF delivery is needed, which would demonstrate the application process and its expected benefits.

5.11 [Rigor and Design] Cycle 11: Development of the Taster

State of the Art. In literature, trialability and compatibility have been identified as key elements for the transfer of new solutions between researchers and IT practitioners in companies [89]. The former refers to when a solution can be experimented with on a limited basis, and the latter to what degree it is consistent with existing values, past experiences, and needs of potential adopters. In this rigor cycle, our objective was to find a time-efficient version to increase trialability.

Design Artifacts. To be able to accommodate more restrictive time frames and more importantly to provide an opportunity to get an introductory view on the method, we packaged the SusAF in a compressed format (taster) for situations where we would only get 90 to 120 minutes to demonstrate the framework. In one instance, we used two topics per dimension, and in the other one two dimensions (social and individual) with all five topics each. As the taster is only a compressed, repackaged version of the SusAF for situations where we are short on time, we do not consider it as an additional artifact.

5.12 [Relevance] Cycle 12: Teaching Application: LS and KTH

Artifact to be validated	Participants	Data collected
Taster version ^a	Students (KTH and La Salle)	Oral feedback Feedback survey SusAD instantiations Instructor reflection

^aV1 (90 minutes) for KTH and V2 (150 minutes) for La Salle.

Evaluation. The taster has been applied twice. The first application of the taster (two topics per dimension) was in a guest lecture for nine master's students at KTH, in Stockholm. During the class, students applied the taster to X (formerly known as Twitter). We received positive oral feedback, in which students said they found the taster useful and insightful. However, the pace was perceived as high.

The second version of the taster (two dimensions with all five topics each) was used with 19 master's students in IT management at La Salle–Ramon Llull University, in Spain, as part of the course "IT Policies and Markets." Students were experienced professionals with 5 to 20 years of experience in technical roles and were aiming to or had just started directive roles (e.g., CTO). During the taster, students applied the framework on their 8-week group projects. After the workshop, they were asked to fill out a Google survey, which 15 students did.

The evaluation showed that the taster changed participants' perception of sustainability, broadening their view and highlighting the importance of the topic in ICT. Most students reported having identified new ideas on their projects and all believed that this kind of analysis should be repeated over time to existing systems. In particular, they commented that the SusAF offered "a greater perception and wider vision" of the effects of systems on sustainability and allowed them to "discover several aspects that they did not consider before."

Finally, while most students felt that the taster worked well and that the results were valuable given the time invested, half of them also reported that the duration was too short.

Similarly, the instructor's experience of the taster, compared to the full workshop delivery, was that it was somewhat rushed and did not give students a chance to get deeper into their discussions.

However, it was also clear that the contradiction in time requested by students versus the time provided by the industry was unsurprising: while the students wanted to learn a full process, the industry wanted to check its relevance before committing to it. Thus, the researchers considered that the taster was ready to be taken for evaluation with an industry partner (see Table 2).

5.13 [Relevance] Cycle 13: Industry Application: Digital Hub

Artifact to be validated	Participants	Data collected
Taster version (V1, 90 minutes)	Researchers and industry stakeholders from various companies in the DigiHub network	Feedback survey Qualitative feedback Instruction reflection

Evaluation. The subsequent relevance cycle took the SusAF taster to industry in Germany in three workshops. A first workshop was conducted in a face-to-face setting, followed by two on-line workshops due to the COVID-19 pandemic. They were organized and advertised by the local government for IT companies. The overall idea was to provide a training platform in the field of digital transformation.

We, as researchers, conducted short workshops providing an insight into our research and tools. At each workshop, approximately 10 representatives from different IT companies took part. The 90-minute workshops started with a short introduction of the SusAF and its relevance, went on with applying the SusAF taster on example systems (e.g., X), and ended with a short feedback round. We took a well-known third-party system as an example, as the workshops had a mix of attendees from different companies. Overall, the participants reported that they got a broader understanding of sustainability and SE and that the mixed audience enriched the discussion. Yet, they felt that there was not much time for a detailed discussion and that the mixed audience led to further time compression, as first they needed to get to know each other. This is the last cycle reported in this article (see Table 2).

Lessons Learned—L6, On Adapting to Need or Context: The delivery format should be adapted per need or context. For example, a longer version can be used for in-depth training, whereas shorter ones can be used for familiarization, or for its practical use when there is no time to apply the tool all at once. Moreover, companies should be able to adapt the tool further according to their needs.

6 DISCUSSION

In this article, we have described 13 cycles in the development of the SusAF. The results from cycles 12 and 13 convinced us that the tool is mature enough to provide useful results in teaching and industry contexts. Clearly, there is always room for improvement, and the SusAF might continue to evolve over the next few years. However, this decade of using design science to develop SusAF has taught us valuable lessons that we share in this section.

6.1 On Lessons Learned from Developing the SusAF

As detailed in previous sections, our experience of SusAF development has resulted in six lessons learned on challenges to be expected in designing a sustainability concerns exploration tool for IT products and services development. In the following, we first discuss how these lessons relate to the development of the SusAF. Thereafter, longer-term implications of each of these lessons are discussed with respect to SE research, education, and practice. Finally, some reflections are presented on the methodological concerns of using design science for sustainability tools development. To ease the review, we have clustered the discussion on the lessons learned into two, addressing (i) the complex nature of sustainability and (ii) the need for efficiency in the application of sustainability analysis.

6.1.1 Conceptual Complexity of Sustainability. The first three lessons learned from our experience (i.e., the need for tools and methods, the need to link up the problem at hand with various

dimensions and temporal aspects of sustainability, and the need for examples) with practitioners and students demonstrate a gap in understanding as to what sustainability is and how it relates to software products and services. We noted that this gap exists because the respective tools and techniques for integrating sustainability into software practice are still lacking. Yet, the lack of such techniques and tools is caused, at least in part, by the conceptual complexity of the notion of sustainability itself.

Not only is sustainability multifaceted (i.e., we have discussed environmental, societal, economic, individual, and technical aspects of it), but its various aspects often conflict with each other (e.g., increasing economic benefits often lead to negative environmental impacts).

While some facets of sustainability have historically remained “invisible” (e.g., environmental or social impacts), others (i.e., economic and technical) have been closely observed, measured, planned for, and publicized. As a result, financial and technical planning, budgeting, monitoring, and maintaining tools have been developed and used for the well-attended facets [15, 78], resulting in so-called “sustainable software practices and processes” [42]. This has led to biased and incomplete perception of sustainability in its own right.

Furthermore, costs and benefits for each of these facets are not always measurable, which makes tradeoff decisions across them difficult to conceive and even harder to quantify/qualify and express (e.g., while environmental impact is measured in CO₂ emissions, social impact is expressed in terms of increased trust and individual impact is expressed in improved self-confidence or well-being, economic impacts can be measured in monetary units, whereas technical ones can be expressed in convenience, longevity of technology, or reduced to financial terms).

Finally, the time-dependent nature of sustainability has made addressing the second and third orders of effects of software solutions very challenging [54].

While the SusAF does not address all of these challenges, it provides (through the visual diagram) the first step in building up a connected picture for understanding the cross-dimensional and cross-temporal impacts of decisions taken in a software solution development. It also provides an introduction to the kinds of questions that consideration of sustainability necessitates (through its dimension-specific question sets). Other topics, such as impact quantification using KPIs and domain-specific refinement of sustainability-focused questions, still remain to be addressed in future research.

Lesson 1: On the Need for Methods and Tools: IT practitioners and students (future IT practitioners) need methods, guidelines, and tools to help them consider the potential sustainability effects of novel and existing IT products. More specifically, IT practitioners need to be more aware of (i) the nature of sustainability, (ii) the interdependencies between sustainability and IT products, and (iii) the possible effects that IT products can have over time on different sustainability dimensions.

Implication for SE research: SE research should recognize sustainability as an SE challenge. Tools and methods need to be developed to enable SE practitioners to identify the potential effects that the software they develop could have on sustainability. Given that, by nature, sustainability is an interdisciplinary concern, SE research on sustainability must be interdisciplinary too. Sustainability is also dynamic (i.e., constantly changing with time). The translation of interdisciplinary and time-dependent characteristics of sustainability into SE processes and tools still remain critical open research challenges today, despite the significant amount of related ongoing work (e.g., [34, 65]).

Implication for SE education: Many universities now offer a general course on sustainability. Unfortunately, such general courses do not establish any relationship with the particular professional practice. So a software engineer, having completed such a course, would not do anything any different in her professional activities. Instead, the specialist SE curriculum must be updated to instruct software engineers on how to integrate sustainability considerations into the requirements, design,

and implementation activities within software projects (which is to be supported with methods and tools from the previous research activities, as was shown in the work of Duboc et al. [35]). In addition, SE education must complement the traditional teaching of the computational thinking, with teaching the holistic (or systems) thinking to help software engineers appreciate the holistic impact of ICT products and processes.

Implication for SE practice: Most SE practitioners have no training on integrating sustainability into their daily practices. To address this, the ICT industry needs to (i) support the on-the-job education for practitioners to fill in this gap, and (ii) make sustainability a part of company culture and trickle that down into development processes, supported by tools and chosen KPIs. While the comprehensive tool and method development remains an open challenge, as discussed earlier, a set of such tools and techniques are now emerging from the academic research (e.g., [34, 65]) and can be trialed and improved by practitioners.

Lesson 2: On the Need for Dimensional and Temporal Link Up. IT students need help to (i) link up the abstract notion of sustainability and its dimensions with the specific domain of the sociotechnical system they are working on at the given time, and (ii) link up the potential structural changes from long-term use of the system (which could become notable in a few years) with development decisions they could take at the system development time. This help is relevant for IT practitioners as well if they have not had previous sustainability training or experience.

Implication for SE research: SE research on interdisciplinary and temporal impacts of IT products has to be carried out on per-domain basis, as the same action from an ICT system will lead to different sustainability impacts in different domains. Moreover, contextual factors (including the local environment, culture, and values) become critical aspects in SE decisions. Given that the temporal dimension is central, the implications of the potential SE decisions (with the respective environmental, social, and individual SE debt) should also be identified and traced.

Implication for SE education: The SE education should include not only techniques for holistic thinking (e.g., systems thinking and systems dynamics) but should also develop specialized SE modules per major industrial/business domains (e.g., SE for telecommunications specialism will include a module on analyzing impacts of use of rare metals in telecommunications infrastructure, impacts of such metals extraction on biodiversity, radio wave interference, and more, as all of this will be relevant in designing systems for the next generation of the telecommunications protocols).

Implication for SE practice: To integrate long-term thinking about the impact of their software solutions into the SE practice of the ICT industry, the responsibility for this impact needs to become part of SE companies business model. Thus, in addition to the mandatory (EU) corporate sustainability reporting directive, the maintenance and upgrade for the post-delivery impact should be accepted as a contractual responsibility of the software delivery contract.

Lesson 3: On the Need for Examples. To be practically usable, the methods and artifacts that help analyze sustainability impacts of a sociotechnical solution on its situated environment need to be accompanied by a set of application guidelines and demonstrative application examples.

Implication for SE research: The SE research community needs to collect a pool of examples of sustainable and unsustainable systems for the major domains (impacts by Compass, Airbnb, Uber, CouchSurfing, etc.). These will then serve as basis for domain-specific pattern development (akin to the design patterns in SE).

Implication for SE education: Case-based teaching is becoming a crucial way forward. On the one hand, it is essential for learning about key characteristics of a specific domain (e.g., sustainability impacts of SE in health vs energy systems). On the other hand, it is necessary as part of accumulating the lived experience in the context of an industrial project, as otherwise the abstract concept of sustainability is difficult to understand.

Implication for SE practice: The examples, necessary for meaningful education on domain-specific sustainability impacts, need to be developed in collaboration with industry. Practitioners can also (i) serve as mentors and/or clients in carrying out case studies with students and (ii) use the educational case studies for up-skilling their current ICT/SE workforce.

6.1.2. Need for Efficiency in Sustainability Analysis.

Lessons 4 through 6 are focused on efficiency constraints within which the businesses must operate. Modern software businesses must be agile and quick to market to be successful [53].

Lesson 4: On the Efficiency of Elicitation and Analysis. Data organization and analysis need to be supported by guidelines which can be applied by practitioners without external help. Support for both full-scale and partial application of the analysis framework is needed. This is because companies may be more interested in certain sustainability dimensions due to their domain or priorities. In particular, focusing on specific effects of dimensions, at a given time, can be critical to some company objectives or priorities.

Implications for SE research: SE research needs to find an efficient way to integrate sustainability analysis (along with the domain-specific knowledge and cross-dimensional and temporal impact assessment) into a tool-supported process. Additionally, the obstacles on the path to such process adoption (organizational, cultural, economic, etc.) need to be researched, and ways to overcome these and assimilate them into everyday industrial practice should be developed.

Implications for SE education: Although time-efficient methods and tools for sustainability in SE are still to be developed, it is critically important to educate students on the conceptual aspects, as well as on the use of the available methods and tools. Thus, SE education must keep up with the most up-to-date research findings so that the research to practice gap is continuously reduced. Additionally (as was shown with the case of the SusAF), working with students on developing and testing methods is both a productive and effective way of moving the research ahead. Thus, the SE educational sphere itself is a tremendous asset for researching the sustainability topics, and the teaching and research aspects of SE should closely collaborate on sustainability.

Implications for SE practice: While integration of sustainability analysis and implementation support in the SE tool chain is not yet in place, SE tool developers can take an opportunity to distinguish themselves from competitors through early delivery of such support. Thus, the more requests for such features are placed by practitioners/companies to tool suppliers, the more likely that such development will be expedited. Additionally, companies would benefit from early up-skilling of their workforce with respect to conceptual and theoretical aspects, as much can be gained already through awareness of RE practices (as discussed with respect to the SusAF). Even this requires initial effort, as any process change is costly. That is why mandatory corporate social responsibility reporting and business model-level integration of sustainability into service delivery are essential.

Lesson 5: On the Introductory Format. Companies and students may lack the time and/or the conviction necessary to carry out a full application of the SusAF when they are first introduced to it. Thus, an “Introduction to” format of SusAF delivery is needed, which would demonstrate the application process and its expected benefits.

Implications for SE research: The utility to try before one buys (or buys into an idea) is well understood in sales. Similarly, researchers need to ensure that they find a way to provide “minimum viable versions” of the sustainability supporting tools and techniques which deliver real value.

Implications for SE education: The education sector needs to find the most appropriate method to deliver both introductory and more advanced methods of integrating sustainability into SE. For instance, the introductory version of a method can be practiced during a lab/seminar session, with the full method expected to be applied during coursework or final year project delivery, and so on.

Implications for SE practice: Where an introductory format is available, companies should test the relevant tools/techniques. It should be clear to companies that the evaluation of the sustainability tools/techniques must be carried out against a different set of KPIs (CO₂ reduction, trust to the company's brand, etc.) than the traditional return on investment.

Lesson 6: Adapting to Need or Context. The delivery format should be adapted per need or context. For example, a longer version can be used for in-depth training, whereas shorter ones can be used for familiarization, or for practical use when there is no time to apply the tool/method fully "all at once." Furthermore, companies should be able to adapt the tool according to their needs.

Implications for SE research: Ordinarily, when researchers develop training materials, this can be a one-size-fits-all affair. We suggest that it is highly beneficial to develop different formats of the training materials for all new tools/techniques. This can, for example, include an in-person workshop, an online course, or a video tutorial followed by a discussion. Additionally, the developed methods and tools should be flexible. For example, support for both full-scale and partial application of a framework could be beneficial. This is because companies may be more interested in certain sustainability dimensions or topics due to their domain or priorities. In particular, focusing on specific effects or dimensions, at a given time, can be critical to some company objectives or priorities.

Implications for SE education: It may not always be possible to carry out the full training with students within a course, so short versions may be useful to give them an overview and serve as a taster for a given methodology/tool. Through appropriately set practical and lab exercises, collaboration projects with industry, case studies, internships and the like, the educational sector can help students gain relevant knowledge and experience with a variety of the (versions of) tools and techniques, as well as appreciate the relevance of contextual factors.

Implications for SE practice: When several versions of a method are available (as shown by the SusAF), the companies can select which formats are most useful at a given time for a given purpose. For instance, the introductory format can be used to get to know a method in general terms, then a partial analysis format can be used to integrate sustainability analysis for a key goal only or for a single dimension of a critical relevance, and so on. We envision that companies would develop their own tailored strategies for choosing and using the various sustainability analysis and development formats for various contexts, clients, and circumstances. The key implication here is that different formats will each have their uses, as long as the general issue of integrating sustainability into own practice is set to be relevant to a company. It is setting this relevance as a priority that requires work as the first step (in communicating the necessity and benefits to shareholders, engaging the top level executives, etc.). This should be done now, irrespective of any further tool/method availability.

However, presently, integration of sustainability analysis into software development is a very time consuming process. This is not surprising, given the previously discussed complexity of five dimensions and three timescales that sustainability analysis requires, aggravated with the lack of previous knowledge of sustainability in companies and unavailability of well-integrated tool chains, supporting software developers [24].

Consequently, even companies that are passionate about sustainability found it difficult to integrate the full SusAF analysis framework into their development processes. This led us to developing several formats of a sustainability analysis method:

- The introductory format helps interested parties with no previous knowledge of sustainability analysis get some familiarity with the topic and process (as per Lesson 5). This is a typical problem in technology transfer [44].

- The partial analysis format, which allows developers to focus on one (or a few) sustainability dimension analysis at a time, helps improve time efficiency in addressing issues of immediate interest, as well as consolidating the experience of the analysis method. At the same time, the workbook developed to accompany the analysis process guides the data collection and recording ‘in workshop real-time,’ removing the need for extensive post-workshop analysis (as per Lesson 4).
- Finally, while the method application within the development context needs to be efficient, the teaching and training contexts will allow much more time for learning. Thus (as per Lesson 6), we suggest differentiating the delivery format per need and context: in teaching and training contexts, full complexity of sustainability needs to be addressed to prevent the simplistic viewpoints of reducing sustainability to one aspect only in the software profession [36]. In practical use, professionals will be able to use simplified versions of the analysis, while also being aware that such simplification leads to building up sustainability debt [12], just as temporarily disregarding good design and development guidelines leads to technical debt [61], which will need to be addressed at a later time.

6.2 On Methodological Concerns of Using Design Science in Sustainability Analysis Framework Development

As noted previously, the main method used in SusAF development was design science. When using design science (as detailed in Section 4), a continuous cycle of problem analysis—solution design—evaluation for a new analysis is iteratively applied until the desired response from the evaluation cycle signals that the development process can be stopped and a satisfactory solution has been delivered. This approach has been reported to perform well in various engineering-related problem contexts [51], which was a key reason for choosing this method for SusAF development. Yet, as the framework development progressed, we observed some incompatibilities in applying the design science method to the challenge of developing a framework for sustainability impact analysis. More specifically, we observed the following:

- *Timescale challenges:* When undertaking an artifact evaluation, the design science method expects that the artifact is tried and evaluated by the prospective users, on basis of whose feedback new improvements for the following iteration cycle are proposed. Indeed, all elements of the SusAF had been tried and evaluated by either students and/or software practitioners who would be the intended end users of the framework either for education or practice. Yet, the analysis conducted with the SusAF refers to the impact of an intended solution on sustainability of its situated environment, which could transpire after long-term use of the system. To fully evaluate correctness and relevance of SusAF analysis results, IT companies should deliver the designed solutions and, after a number of years, evaluate the impacts that transpire with those suggested by the SusAF. However, such long-term design and evaluation cycles are challenging with the usual design science practices and timescales. That would require a project in which data is collected and analyzed over decades within the same context (e.g., company, domain, and system). For the design of novel sustainability tools, this can be challenging, as tool designers usually count with volunteering early adopters with no obligation and are unlikely to commit to decades of data gathering.
- *Scope and measurement challenges:* When analyzing the prospective impacts of the intended IT solution on the social, individual, and environmental dimensions of sustainability, we were unable to identify agreed upon units of measurement which could be integrated with the “normal” design science metrics in the context of tool development (effectiveness, ease of use, application time, etc.) or boundaries for impact evaluation. However, effectiveness can

be difficult to measure to when the system would impact the prospective user's well-being, or cohesion in a society where that system is used, or the impact that the system may have on biodiversity. Similarly, it is often unclear whether the impact should be evaluated for some groups of users or all of them, at the scale of neighborhood, city, country, or worldwide. Some examples of "unmeasurable concerns" from industrial partners case studies are as follows:

- *Scope definition*: The Datamatrix case study identified a threat to wildlife from the fencing of solar panels for water extraction, but this was not quantified or measured, as it was not clear if the company should be responsible for this issue; in other words, it was not clear if concern about wildlife is within the scope of the solution design. Given that sustainability is a systemic concern, it is likely to bring along emergent properties, some of which will require redrawing system boundaries and adding new metrics and measurements at the time when evaluation is in progress. To the best of our knowledge, this consideration of emergent properties and dynamic evolution of measurement criteria has not been considered previously in design science and would require methodology extension to accommodate.
- *Measurement definition*: Partneur was aiming at creating trust across small businesses using their platform but was not sure how to measure such a trust being achieved, and design science had no relevant metrics to draw on. We recognize that, should the SusAF itself develop such metrics, these could be utilized within following applications that draw on design science. However, given that the SusAF was the artifact being constructed with design science, we naturally expected to draw on the metrics arsenal of that methodology. Here too, expansion is needed for the set of design science metrics used for ICT artifact evaluation when sustainability is considered.

To summarize, at present, design science is a method to be applied within certain boundaries, such as direct and indirect effects of an IT product on a company and its direct and indirect stakeholders, whereas sustainability has a wider scope. Sustainability is a systemic concern that brings along the need to cater for evaluating its emergent properties as they emerge—in use and through time.

- *Design science is anthropocentric*: We have been very aware of the limitations of the anthropocentric focus of design science. The foundational literature for design science repeatedly states that it is focused on human problems: "Design science research is a research paradigm in which a designer answers questions relevant to human problems" [51, p. 5]. Indeed, statements such as "1. Focus on the user and all else will follow" [51, p. 85] might be characterized as an anthropocentric obsession with a very limited range of humans. As a poignant example, Holopainen et al. [56] describe the use of design science in the design of virtual reality services for forest management. The stakeholders considered are forest owners and forest management service providers. There is no consideration of non-human stakeholders in the forest or of future human stakeholders who will value highly the function of forests as carbon sinks. We concur with Brendel et al. [18] concerning the limits of design science "due to the lack of environmental beliefs held by the target users of said artifacts" [18, p. 25]. At a time when other disciplines are engaging with the representation and legal rights of non-human entities [26], this narrow anthropocentric approach is clearly inadequate. There is a demand for design approaches that recognize the complementarity of human and non-human agents [88]. Recently, proposals have been made for a socially responsible design science [73], but there is clearly also a need for research and development of ecologically responsible design science.

In summary, given the long-term, multiscale, and multidimensional impacts that IT products will have on the sustainability of their situated environments, the cultural (i.e., long-term

co-responsibility for system development, commissioning, and use) and contractual conditions (i.e., methodologies for multidimensional impact monitoring with consistent measurement units and agreed impact scope) within which such projects are developed need to be updated.

6.3 Threats to Validity

The research team followed the well-established design science research method for the development and execution of this study. For completeness, the threats to the study validity and their mitigation [106] are discussed next.

Threats to internal validity relate to the participant and researcher selection [106]. Given that both the students and companies participating in the study were engaged through a convenience sampling method, they may not be representative compared to the general student and industry population. To redress this potential bias, the student participants were selected from five countries as well as from different educational levels (BSc, MSc, and Ph.D. students). Industrial participants were selected from six countries and from significantly different application domains, namely a collaborative web app (Partneur), an embedded system for water management (Datamatrix), a cloud business application (Visma), a driving and traffic safety facility (Premier Park), and a globally distributed scheduling simulation system (Jeppesen). Similarly, although the membership of the Delphi panel was convened from collaborating researchers, they were all from different universities, from across different countries, and had varied backgrounds and experience from other projects on sustainability and thus were able to provide a diverse set of views for the panel. Another internal threat is the reactive bias, as the students might have felt pressed to answer in a way that conformed with the expectations of their lecturers. This bias was mitigated through reassuring the students that there is no right or wrong response—all answers would be equally valid.

Threats to construct validity relate to the degree to which inferences can legitimately be made from the operationalizations in the study to the theoretical constructs on which those operationalizations were based. Here we consider the specific threats of construct confounding, novelty effects, and experimenter expectancies. We are aware that the SusAF consists of many artifacts, thus it is difficult to tell which ones are working. However, the artifacts were added step by step according to feedback in each cycle of what was missing or suboptimal. Following the design science approach, we are confident that we have concluded with a useful set of artifacts. Nevertheless, construct confounding cannot be ruled out completely. It is difficult to determine if the feedback given by the participants was due to their unfamiliarity with the topic or with the artifacts themselves. While our approach expects a learning curve and an expansion of participants' understanding of sustainability, there is no way to completely exclude novelty effects. However, all participants had basic knowledge of sustainability, and the tools we use for visualization are rather basic templates to fill out. With regard to experimenter expectancies and the extent by which facilitators influence the workshop outcomes and evaluations, we acknowledge the potential researcher bias, as the facilitators of the workshops were SusAF researchers. To mitigate this, we have published the framework available for anyone to conduct a workshop without the researchers being present.

Threats to external validity relate to the generalizability of our findings. What assurance does the research give us that the challenges faced with the SusAF are likely to occur when developing other tools? To mitigate this threat, during the relevance cycles, we referred to literature to see whether the challenges we were facing had occurred to others in similar contexts and how they were approached. Although we had 43 industrial participants from six countries, the number could always be increased to achieve more generalization potential.

Threats to conclusion validity relate to the degree to which conclusions reached about relationships in data are reasonable. This study as described in Section 5 has more than 208 student participants from IT-related undergraduate and postgraduate courses across five countries, which

increases confidence in the generalization of the study findings across countries and cultures. Furthermore, to address the challenges with the unreliability of treatment implementation, all facilitators followed the same process and used the same set of workshop slides during the workshops that belonged to the same cycle. Lessons learned and best practices were shared among all facilitators in regular weekly meetings and implemented in organizing workshop sessions.

Considering data reliability, part of the data is quantitative based on surveys, SusADs, and reports. We relied on simple descriptive statistics, so conclusions are straightforward and the results are repeatable. An open data package, containing all non-NDA or consent-protected materials, is available for replication.⁵

7 CONCLUSION AND OUTLOOK

In this article, we discussed the development of the SusAF using design science as the research methodology. We shared our experiences and presented the lessons learned from our work in the development of a sustainability tool through the application of design science. Additionally, we discussed the implications that these lessons could have on SE research, education, and practice. We are aware that creating such an approach, like the SusAF, deals with many uncertainties and is a difficult and challenging endeavor. Sustainability and its effects are difficult to understand and work with. By identifying the general challenges and lessons learned that are likely to occur when developing a sustainability tool, and discussing their implications, it is our hope to inspire and support other researchers in developing sustainability awareness tools that can be applied in education and industry.

Based on the holistic design science process of SusAF development, we identified six lessons learned regarding the nature of sustainability and the need for efficiency in sustainability analysis. Design science research was adopted because it generated knowledge about the design of an innovative solution to sustainability awareness for software engineers. Having used design science, we were able to synthesize those lessons, thus we found design science useful in this context. However, we cannot compare it to any other method that we have applied as extensively. Additionally, it has some shortcomings, as we have identified methodological concerns such as timescale challenges and scope and measurement challenges. Nevertheless, by using design science research, we were able to successfully create the SusAF.

Currently, we continue evolving the SusAF using design science research. Especially, we are investigating if the state of the company has an influence on how useful the SusAF is for them. For example, does it make a difference if the company is developing a new IT product or is ready to redesign its current version or if the company has a well-established IT product with only small incremental improvement steps planned? Moreover, we are planning to combine the SusAF framework with design thinking approaches and with existing measurement frameworks. Finally, we are planning to map the SusAF to existing SE process methods—specifically, we are currently working on the integration of SusAF and agile methods.

8 CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

ACKNOWLEDGMENTS

We would like to thank all industry partners, students, and workshop participants who have supported this research through their presence and discussion contributions.

⁵<https://figshare.com/s/0bd7430f9aa7fc197eb8>

REFERENCES

- [1] Okhaide Akhigbe. 2016. Towards a regulator-oriented regulatory intelligence framework. In *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference (RE'16)*. 415–420. <https://doi.org/10.1109/RE.2016.26>
- [2] Ian F. Alexander and Ljerka Beus-Dukic. 2009. *Discovering Requirements: How to Specify Products and Services*. Wiley.
- [3] A. D. Alharthi, M. Spichkova, and M. Hamilton. 2018. SuSoftPro: Sustainability profiling for software. In *Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE'18)*. 500–501. <https://doi.org/10.1109/RE.2018.00072>
- [4] Ruba Aljafari and Deepak Khazanchi. 2013. On the veridicality of claims in design science research. In *Proceedings of the 2013 46th Hawaii International Conference on System Sciences*. 3747–3756. <https://doi.org/10.1109/HICSS.2013.427>
- [5] Myles R. Allen, Mustafa Babiker, Yang Chen, Heleen de Coninck, Sarah Connors, Renée van Diemen, Opha Pauline Dube, Kristie L. Ebi, Francois Engelbrecht, Marion Ferrat, James Ford, Piers Forster, Sabine Fuss, Bola Tania Guilén, Jordan Harold, Ove Hoegh-Guldberg, Jean-Charles Hourcade, Daniel Huppmann, Daniela Jacob, Kejun Jiang, Tom Gabriel Johansen, Mikiko Kainuma, Kiane de Kleijne, Elmar Kriegler, Debora Ley, Diana Liverman, Nathalie Mahowald, Valérie Masson-Delmotte, J. B. Robin Matthews, Richard J. Millar, Katja Mintenbeck, Angela Morelli, Wilfran Moufouma-Okia, Luis Mundaca, Maïke Nicolai, Chukwumerije Okereke, Minal Pathak, Antony Payne, Roz Pidcock, Anna Pirani, Elvira Poloczanska, P. Hans-Otto, Aromar Revi, Keywan Riahi, Debra C. Roberts, Joeri Rogelj, Joyashree Roy, Sonia I. Seneviratne, Priyadarshi R. Shukla, James Skea, Raphael Slade, Drew Shindell, Chandni Singh, William Solecki, Linda Steg, Michael Taylor, Petra Tschakert, Henri Waisman, Rachel Warren, Panmao Zhai, and Kirsten Zickfeld. 2018. *Summary for Policymakers*. IPCC.
- [6] Stephanos Anastasiadis, Stephanie Perkiss, Bonnie A. Dean, Leopold Bayerlein, Maria Alejandra Gonzalez-Perez, Alec Wersun, Pilar Acosta, Hannah Jun, and Belinda Gibbons. 2020. Teaching sustainability: Complexity and complexities. *Journal of Applied Research in Higher Education*. Published Online, May 7, 2020.
- [7] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. Venters. 2016. Requirements: The key to sustainability. *IEEE Software* 33, 1 (2016), 56–65.
- [8] C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. Venters. 2015. Sustainability design and software: The Karlskrona manifesto. In *Proceedings of the 37th International Conference on Software Engineering*, Vol. 2. IEEE, 467–476.
- [9] Sarah Beecham, Tracy Hall, Carol Britton, Michaela Cottee, and Austen Rainer. 2005. Using an expert panel to validate a requirements process improvement model. *Journal of Systems and Software* 76, 3 (2005), 251–275. <https://doi.org/10.1016/j.jss.2004.06.004>
- [10] Lotfi Belkhir and Ahmed Elmeli. 2018. Assessing ICT global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production* 177 (2018), 448–463. <https://doi.org/10.1016/j.jclepro.2017.12.239>
- [11] Mike Berners-Lee. 2019. *There Is No Planet B: A Handbook for the Make or Break Years*. Cambridge University Press.
- [12] Stefanie Betz, Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin Venters. 2015. Sustainability debt: A metaphor to support sustainability design decisions. In *Proceedings of the International Workshop on Requirements Engineering for Sustainable Systems*.
- [13] Stefanie Betz, Leticia Duboc, Birgit Penzenstadler, Jari Porras, Ruzanna Chitchyan, Norbert Seyff, Colin C. Venters, and Ian Brooks. 2022. The SusA Workshop—Improving Sustainability Awareness to Inform Future Business Process and Systems Design. Retrieved March 16, 2024 from <https://doi.org/10.5281/zenodo.7342574>
- [14] Elizabeth Bjarnason, Helen Sharp, and Björn Regnell. 2019. Improving requirements-test alignment by prescribing practices that mitigate communication gaps. *Empirical Software Engineering* 24, 4 (2019), 2364–2409. <https://doi.org/10.1007/s10664-019-09698-6>
- [15] Barry W. Boehm. 2006. Value-based software engineering: Overview and agenda. In *Value-Based Software Engineering*. Springer, 3–14.
- [16] Lena Börjeson, Mattias Höjer, Karl-Henrik Dreborg, Tomas Ekvall, and Göran Finnveden. 2006. Scenario types and techniques: towards a user's guide. *Futures* 38, 7 (2006), 723–739.
- [17] Ron Bradfield, George Wright, George Burt, George Cairns, and Kees Van Der Heijden. 2005. The origins and evolution of scenario techniques in long range business planning. *Futures* 37, 8 (2005), 795–812.
- [18] Alfred Benedikt Brendel, Friedrich Chasin, Milad Mirbabaie, Dennis M. Riehle, and Christine Harnischmacher. 2022. Review of design-oriented green information systems research. *Sustainability (Basel, Switzerland)* 14, 8 (2022), 4650. <https://doi.org/10.3390/su14084650>
- [19] I. S. Brito, J. M. Conejero, A. Moreira, and J. Araújo. 2018. A concern-oriented sustainability approach. In *Proceedings of the 2018 12th International Conference on Research Challenges in Information Science (RCIS'18)*. 1–12. <https://doi.org/10.1109/RCIS.2018.8406676>
- [20] Jan vom Brocke and Stefan Seidel. 2012. Environmental sustainability in design science research: Direct and indirect effects of design artifacts. In *Design Science Research in Information Systems: Advances in Theory and Practice*. Lecture Notes in Computer Science, Vol. 7286. Springer, 294–308. https://doi.org/10.1007/978-3-642-29863-9_22

- [21] Sandra L. Buitrón, Francisco J. Pino, and Tulio Rojas Curieux. 2020. Using design-science for the representation of non functional requirements. In *Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI'20)*. 1–6. <https://doi.org/10.23919/CISTI49556.2020.9140876>
- [22] J. Cabot, S. Easterbrook, J. Horkoff, L. Lessard, S. Liaskos, and J. Mazon. 2009. Integrating sustainability in decision-making processes: A modelling strategy. In *Proceedings of the 31st International Conference on Software Engineering—Companion Volume*. 207–210. <https://doi.org/10.1109/ICSE-COMPANION.2009.5070983>
- [23] Peter Checkland. 1999. *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. John Wiley & Sons Ltd.
- [24] Ruzanna Chitchyan, Christoph Becker, Stefanie Betz, Leticia Duboc, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. 2016. Sustainability design in requirements engineering: State of practice. In *Proceedings of the 38th International Conference on Software Engineering Companion (ICSE'16)*. ACM, New York, NY, USA, 533–542. <https://doi.org/10.1145/2889160.2889217>
- [25] R. Chitchyan, S. Betz, L. Duboc, B. Penzenstadler, S. Easterbrook, C. Ponsard, and C. Venters. 2015. Evidencing sustainability design through examples. In *Proceedings of the 4th International Workshop on Requirements Engineering for Sustainable Systems*.
- [26] Cristy Clark, Nia Emmanouil, John Page, and Alessandro Pelizzon. 2019. Can You hear the rivers sing? Legal personhood, ontology, and the nitty-gritty of governance. *Ecology Law Quarterly* 45, 4 (2019), 787–844. <https://doi.org/10.15779/Z388S4JP7M>
- [27] Nelly Condori-Fernandez, Patricia Lago, Miguel R. Luaces, and Ángeles S. Places. 2020. An action research for improving the sustainability assessment framework instruments. *Sustainability* 12, 4 (Feb. 2020), 1–25. <https://doi.org/10.3390/su12041682>
- [28] Jacqueline Corbett. 2013. Designing and using carbon management systems to promote ecologically responsible behaviors. *Journal of the Association for Information Systems* 14, 7 (2013), 2.
- [29] Jacek Dabrowski, Fitsum Meshesha Kifetew, Denisse Munante, Emmanuel Letier, Alberto Siena, and Angelo Susi. 2017. Discovering requirements through goal-driven process mining. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Workshops (REW'17)*. 199–203. <https://doi.org/10.1109/REW.2017.61>
- [30] Maya Daneva, Nelly Condori-Fernandez, Klaas Sikkel, and Andrea Herrmann. 2018. Experiences in using practitioner's checklists to evaluate the industrial relevance of requirements engineering experiments. In *Proceedings of the 2018 IEEE/ACM 6th International Workshop on Conducting Empirical Studies in Industry (CESI'18)*. 5–12.
- [31] Gerard De Leoz and Stacie Petter. 2018. Considering the social impacts of artefacts in information systems design science research. *European Journal of Information Systems* 27, 2 (2018), 154–170.
- [32] Philipp Diebold and Antonio Vetro. 2014. Bridging the Gap: SE technology transfer into practice—Study design and preliminary results. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*. <https://doi.org/10.1145/2652524.2652552>
- [33] Philipp Diebold, Antonio Vetro, and Daniel Mendez Fernandez. 2015. An exploratory study on technology transfer in software engineering. In *Proceedings of the 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'15)*. 1–10. <https://doi.org/10.1109/ESEM.2015.7321189>
- [34] Leticia Duboc, Stefanie Betz, Birgit Penzenstadler, Sedef Akinli Kocak, Ruzanna Chitchyan, Ola Leifler, Jari Porras, Norbert Seyff, and Colin C. Venters. 2019. Do we really know what we are building? Raising awareness of potential sustainability effects of software systems in requirements engineering. In *Proceedings of the 2019 IEEE 27th International Requirements Engineering Conference (RE'19)*. 6–16. <https://doi.org/10.1109/RE.2019.00013>
- [35] Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff, and Colin C. Venters. 2020. Requirements engineering for sustainability: An awareness framework for designing software systems for a better tomorrow. *Requirements Engineering* 25, 4 (2020), 469–492.
- [36] Steve Easterbrook. 2014. From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. In *Proceedings of the 2014 Conference on ICT for Sustainability*. 235–244. <https://doi.org/10.2991/ict4s-14.2014.28>
- [37] Emelie Engström, Margaret-Anne D. Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. 2019. A review of software engineering research from a design science perspective. *CoRR abs/1904.12742* (2019). <http://arxiv.org/abs/1904.12742>
- [38] Robert Farrell and Cliff Hooker. 2013. Design, science and wicked problems. *Design Studies* 34, 6 (2013), 681–705.
- [39] Iffat Fatima and Patricia Lago. 2023. Towards a sustainability-aware software architecture evaluation for cloud-based software services. In *Proceedings of the European Conference on Software Architecture*.
- [40] Robert Fichman and Chris Kemerer. 2002. The assimilation of software process innovations: An organizational learning perspective. *Management Science* 43 (2002), 1345–1363. <https://doi.org/10.1287/mnsc.43.10.1345>
- [41] Farnaz Fotrousi. 2016. Quality-impact assessment of software systems. In *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference (RE'16)*. 427–431. <https://doi.org/10.1109/RE.2016.53>
- [42] Martin Fowler and Jim Highsmith. 2001. The agile manifesto. *Software Development* 9, 8 (2001), 28–35.

- [43] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. 2021. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns* 2, 9 (2021), 100340. <https://doi.org/10.1016/j.patter.2021.100340>
- [44] Vahid Garousi, Kai Petersen, and Baris Ozkan. 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology* 79 (2016), 106–127. <https://doi.org/10.1016/j.infsof.2016.07.006>
- [45] M. L. Gibson, R. Chitchyan, C. C. Venters, M. Palacin-Silva, L. Duboc, B. Penzenstadler, S. Betz, and N. Seyff. 2017. Mind the chasm: A UK FishEye lens view of sustainable software engineering. In *Proceedings of the 6th International Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy'17)*. 15–24. <http://eprints.hud.ac.uk/id/eprint/32429/>
- [46] Micayla Goodrum, Jane Cleland-Huang, Robyn Lutz, Jinghui Cheng, and Ronald Metoyer. 2017. What requirements knowledge do developers need to manage change in safety-critical systems? In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE'17)*. 90–99. <https://doi.org/10.1109/RE.2017.65>
- [47] Tony Gorschek, Per Garre, Stig Larsson, and Claes Wohlin. 2006. A model for technology transfer in practice. *IEEE Software* 23, 6 (2006), 88–95. <https://doi.org/10.1109/MS.2006.147>
- [48] Iris Groher and Rainer Weinreich. 2017. An interview study on sustainability concerns in software development projects. In *Proceedings of the 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA'17)*. 350–358. <https://doi.org/10.1109/SEAA.2017.70>
- [49] André Heuer, Philipp Diebold, and Torsten Bandyszak. 2014. Supporting technology transfer by providing recommendations for writing structured guidelines. *CEUR Workshop Proceedings* 1129 (2014), 47–56.
- [50] Alan Hevner. 2007. A three cycle view of design science research. *Scandinavian Journal of Information Systems* 19 (2007), 87–92.
- [51] Alan Hevner and Samir Chatterjee. 2010. *Design Research in Information Systems: Theory and Practice*. Integrated Series in Information Systems, Vol. 22. Springer. <https://doi.org/10.1007/978-1-4419-5653-8>
- [52] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design science in information systems research. *MIS Quarterly* 28, 1 (2004), 75–105. <http://www.jstor.org/stable/25148625>
- [53] Jim Highsmith and Alistair Cockburn. 2001. Agile software development: The business of innovation. *Computer* 34, 9 (2001), 120–127.
- [54] L. Hilty and B. Aebischer. 2015. ICT for sustainability: An emerging research field. In *ICT Innovations for Sustainability*. Springer, 3–36.
- [55] Lorenz M. Hilty and Bernard Aebischer. 2014. *ICT Innovations for Sustainability*. Springer.
- [56] Jani Holopainen, Osmo Mattila, Essi Pöyry, and Petri Parvinen. 2020. Applying design science research methodology in the development of virtual reality forest management services. *Forest Policy and Economics* 116 (2020), 102190. <https://doi.org/10.1016/j.forpol.2020.102190>
- [57] IPCC. 2022. *Climate Change 2022: Impacts, Adaptation and Vulnerability: Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press.
- [58] Julian Kirchherr, Laura Piscicelli, Ruben Bour, Erica Kostense-Smit, Jennifer Muller, Anne Huibrechtse-Truijens, and Marko Hekkert. 2018. Barriers to the circular economy: Evidence from the European Union (EU). *Ecological Economics* 150 (2018), 264–272. <https://doi.org/10.1016/j.ecolecon.2018.04.028>
- [59] Barbara Kitchenham, Shari Lawrence Pfleeger, Beth McColl, and Suzanne Eagan. 2002. An empirical study of maintenance and development estimation accuracy. *Journal of Systems and Software* 64, 1 (2002), 57–77.
- [60] Benjamin Klör, Markus Monhof, Daniel Beverungen, and Sebastian Bräuer. 2018. Design and evaluation of a model-driven decision support system for repurposing electric vehicle batteries. *European Journal of Information Systems* 27, 2 (2018), 171–188.
- [61] Philippe Kruchten, Robert L. Nord, Ipek Ozkaya, and Davide Falessi. 2013. Technical debt: Towards a crisper definition report on the 4th International Workshop on Managing Technical Debt. *ACM SIGSOFT Software Engineering Notes* 38, 5 (2013), 51–54.
- [62] Patricia Lago, Sedef Akinli Koçak, Ivica Crnkovic, and Birgit Penzenstadler. 2015. Framing sustainability as a property of software quality. *Communications of the ACM* 58, 10 (Sept. 2015), 70–78. <https://doi.org/10.1145/2714560>
- [63] Jon Landeta. 2006. Current validity of the Delphi method in social sciences. *Technological Forecasting and Social Change* 73, 5 (2006), 467–482. <https://doi.org/10.1016/j.techfore.2005.09.002>
- [64] Soren Lauesen and Otto Vinter. 2001. Preventing requirement defects: An experiment in process improvement. *Requirements Engineering* 6, 1 (2001), 37–50.
- [65] Grace Lewis, Patricia Lago, Sebastian Echeverria, and Pieter Simoens. 2019. A tale of three systems: Case studies on the application of architectural tactics for cyber-foraging. *Future Generation Computer Systems* 96 (2019), 119–147. <https://doi.org/10.1016/j.future.2019.01.052>

- [66] Samuel Mann. 2011. *The Green Graduate: Educating Every Student as a Sustainable Practitioner*. NZCER Press, Wellington, New Zealand.
- [67] Samuel Mann, Logan Muller, Janet Davis, Claudia Roda, and Alison Young. 2010. Computing and sustainability: evaluating resources for educators. *ACM SIGCSE Bulletin* 41, 4 (2010), 144–155.
- [68] Samuel Mann and Lesley Smith. 2011. Collaboration in sustainability vision. In *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems (CTS'11)*. IEEE, 404–412.
- [69] Adam Mazurkiewicz and Beata Poteralska. 2017. Technology transfer barriers and challenges faced by R&D organizations. *Procedia Engineering* 182 (2017), 457–465. <https://doi.org/10.1016/j.proeng.2017.03.134>
- [70] Donella H. Meadows, Dennis L. Meadows, and J. Randers. 1992. *Beyond the Limits: Global Collapse or a Sustainable Future*. Earthscan Publications.
- [71] Donella H. Meadows, Jørgen Randers, and Dennis L. Meadows. 2004. *The Limits to Growth: The 30-Year Update*. Chelsea Green Publishing Company.
- [72] Gabriel Alberto Garcia Mireles, Ma Ángeles Moraga, Félix García, and Mario Piattini. 2017. A classification approach of sustainability aware requirements methods. In *Proceedings of the 2017 12th Iberian Conference on Information Systems and Technologies (CISTI'17)*. IEEE, 1–6.
- [73] Mike Monson. 2023. Socially responsible design science in information systems for sustainable development: A critical research methodology. *European Journal of Information Systems* 32, 2 (2023), 207–237. <https://doi.org/10.1080/0960085X.2021.1946442>
- [74] Lloyd Montgomery and Daniela Damian. 2017. What do support analysts know about their customers? On the study and prediction of support ticket escalations in large software organizations. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE'17)*. 362–371. <https://doi.org/10.1109/RE.2017.61>
- [75] G. Mussbacher and D. Nuttall. 2014. Goal modeling for sustainability: The case of time. In *Proceedings of the IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE'14)*. 7–16. <https://doi.org/10.1109/MoDRE.2014.6890821>
- [76] Bendra Ojameruaye, Rami Bahsoon, and Leticia Duboc. 2016. Sustainability Debt: A portfolio-based approach for evaluating sustainability requirements in architectures. In *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. 543–552.
- [77] C. Okoli and S. Pawlowski. 2004. The Delphi method as a research tool: An example, design considerations and applications. *Information & Management* 42, 1 (2004), 15–29. <https://doi.org/10.1016/j.im.2003.11.002>
- [78] Helena Holmström Olsson, Hiva Alahyari, and Jan Bosch. 2012. Climbing the “stairway to heaven”—A multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In *Proceedings of the 2012 38th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 392–399.
- [79] Alexander Osterwalder, Yves Pigneur, and Tim Clark. 2010. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons, Hoboken, NJ.
- [80] Sofia Ouhbi and Nuno Pombo. 2020. Software engineering education: Challenges and perspectives. In *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON'20)*. IEEE, 202–209.
- [81] Birgit Penzenstadler. 2013. Towards a definition of sustainability in and for software engineering. In *Proceedings of the 28th Annual Symposium on Applied Computing*. ACM, 1183–1185. <https://doi.org/10.1145/2480362.2480585>
- [82] Birgit Penzenstadler, Mara Anoushka, Nam Stephanie, and Budzinski Brian. 2019. Exploratory case study on sustainability awareness with a startup for business models. In *Proceedings of the 6th International Conference on ICT for Sustainability (ICT4S'19)*.
- [83] Stefanie Betz and Birgit Penzenstadler. 2024. Open Data Package. <https://zenodo.org/doi/10.5281/zenodo.10894317>
- [84] Birgit Penzenstadler, Stefanie Betz, Leticia Duboc, Norbert Seyff, Jari Porras, Shola Oyediji, Ian Brooks, and Colin C. Venters. 2021. Iterative sustainability impact assessment: When to propose? In *Proceedings of the 2021 IEEE/ACM International Workshop on Body of Knowledge for Software Sustainability (BoKSS'21)*. 5–6. <https://doi.org/10.1109/BoKSS52540.2021.00010>
- [85] Birgit Penzenstadler, Stefanie Betz, Colin C. Venters, Ruzanna Chitchyan, Jari Porras, Norbert Seyff, Leticia Duboc, and Christoph Becker. 2018. Everything is INTERRELATED: Teaching software engineering for sustainability. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET'18)*. ACM, New York, NY, USA, 153–162. <https://doi.org/10.1145/3183377.3183382>
- [86] B. Penzenstadler, L. Duboc, C. C. Venters, S. Betz, N. Seyff, K. Wnuk, R. Chitchyan, S. M. Easterbrook, and C. Becker. 2018. Software engineering for sustainability: Find the leverage points! *IEEE Software* 35, 4 (July 2018), 22–33. <https://doi.org/10.1109/MS.2018.110154908>
- [87] Birgit Penzenstadler and Henning Femmer. 2013. A generic model for sustainability with process- and product-specific instances. In *Proceedings of the 2013 Workshop on Green in/by Software Engineering (GIBSE'13)*. ACM, New York, NY, USA, 3–8. <https://doi.org/10.1145/2451605.2451609>

- [88] Massimo Perriccioli. 2021. The alliance between ecology and cybernetics for a new design science. *TECHNE—Journal of Technology for Architecture and Environment* 21 (2021), 88–93. <https://doi.org/10.13128/techne-9855>
- [89] S. Pfleeger. 1999. Understanding and improving technology transfer in software engineering. *Journal of Systems and Software* 47 (1999), 111–124.
- [90] Yen Dieu Pham, Abir Bouraffa, and Walid Maalej. 2020. ShapeRE: Towards a multi-dimensional representation for requirements of sustainable software. In *Proceedings of the 2020 IEEE 28th International Requirements Engineering Conference (RE'20)*. 358–363. <https://doi.org/10.1109/RE48521.2020.00048>
- [91] Jari Porras, Colin C. Venters, Birgit Penzenstadler, Leticia Duboc, Stefanie Betz, Norbert Seyff, Saeid Heshmatisafa, and Shola Oyediji. 2021. How could we have known? Anticipating sustainability effects of a software product. In *Software Business. Lecture Notes in Business Information Processing*, Vol. 434. Springer, 10–17.
- [92] Roger S. Pressman. 1994. *Software Engineering: A Practitioner's Approach* (European ed.). McGraw-Hill.
- [93] Quality Assurance Agency for Higher Education and Advance HE. 2021. *Education for Sustainable Development Guidance*. Technical Report. QAA and Advance HE, Gloucester. <https://membershipsresources.qaa.ac.uk/s/article/Education-for-Sustainable-Development-Guidance>
- [94] Kate Raworth. 2017. *Doughnut Economics: Seven Ways to Think Like a 21st-Century Economist*. Penguin Random House, London.
- [95] Dagmar Reinert, Christian Busse, and Stephan M. Wagner. 2019. Using country sustainability risk to inform sustainable supply chain management: A design science study. *Journal of Business Logistics* 40, 3 (2019), 241–264.
- [96] Tim Rietz. 2019. Designing a conversational requirements elicitation system for end-users. In *Proceedings of the 2019 IEEE 27th International Requirements Engineering Conference (RE'19)*. 452–457. <https://doi.org/10.1109/RE.2019.00061>
- [97] K. Roher and D. Richardson. 2013. A proposed recommender system for eliciting software sustainability requirements. In *Proceedings of the 2013 2nd International Workshop on User Evaluations for Software Engineering Researchers (USER'13)*. 16–19. <https://doi.org/10.1109/USER.2013.6603080>
- [98] K. Roher and D. Richardson. 2013. Sustainability requirement patterns. In *Proceedings of the 2013 3rd International Workshop on Requirements Patterns (RePa'13)*. 8–11.
- [99] Theresia Ratih Dewi Saputri and Seok-Won Lee. 2021. Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. *Information and Software Technology* 129 (2021), 106407.
- [100] Christoph Schneider and Stefanie Betz. 2022. Transformation²: Making software engineering accountable for sustainability. *Journal of Responsible Technology* 10 (2022), 100027.
- [101] Thorsten Schoormann, Dennis Behrens, and Ralf Knackstedt. 2018. Design principles for leveraging sustainability in business modelling tools. In *Proceedings of the European Conference on Information Systems*. <https://api.semanticscholar.org/CorpusID:56140270>
- [102] F. Schorr and L. Hvam. 2020. Measuring information technology service levels: A literature review. In *Proceedings of the 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM'20)*. 265–269.
- [103] Douglas Schuler. 2021. On beyond wicked: Exploring the uses of “wicked problems.” In *Proceedings of the 7th Workshop on Computing within Limits*.
- [104] S. Schwartz. 2012. An overview of the Schwartz theory of basic values. *Online Readings in Psychology and Culture* 2 (2012), 11.
- [105] N. Seyff, S. Betz, L. Duboc, C. Venters, C. Becker, R. Chitchyan, B. Penzenstadler, and M. Nöbauer. 2018. Tailoring requirements negotiation to sustainability. In *Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE'18)*. 304–314. <https://doi.org/10.1109/RE.2018.00038>
- [106] W. R. Shadish, Thomas D. Cook, and D. T. Campbell. 2002. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Cengage Learning.
- [107] Strongly Sustainable Business Model Group (SSBMG). 2021. The Flourishing Business Canvas. Retrieved March 16, 2024 from <http://flourishingbusiness.org/the-toolkit-flourishing-business-canvas>
- [108] Harald Sverdrup and Mats Svensson. 2005. Defining the concept of sustainability—A matter of systems thinking and applied systems analysis. In *Systems Approaches and Their Applications*. Springer, 143–164. https://doi.org/10.1007/1-4020-2370-7_6
- [109] UNESCO. 2017. *Education for Sustainable Development Goals: Learning Objectives*. Technical Report. UNESCO, Paris. <https://unesdoc.unesco.org/ark:/48223/pf0000247444>
- [110] Antony Upward and Peter Jones. 2015. An ontology for strongly sustainable business models: Defining an enterprise framework compatible with natural and social science. *Organization & Environment* 29 (2015), 97–123. <https://doi.org/10.1177/1086026615592933>
- [111] Colin C. Venters, Norbert Seyff, Christoph Becker, and Stefanie Betz. 2017. Characterising sustainability requirements: A new species red herring or just an odd fish? In *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS'17)*. 3–12. <https://doi.org/10.1109/ICSE-SEIS.2017.2>

- [112] Colin C. Venters, Rafael Capilla, Stefanie Betz, Birgit Penzenstadler, Tom Crick, Steve Crouch, Elisa Yumi Nakagawa, Christoph Becker, and Carlos Carrillo. 2018. Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software* 138 (2018), 174–188. <https://doi.org/10.1016/j.jss.2017.12.026>
- [113] Colin C. Venters, Rafael Capilla, Elisa Yumi Nakagawa, Stefanie Betz, Birgit Penzenstadler, Tom Crick, and Ian Brooks. 2023. Sustainable software engineering: Reflections on advances in research and practice. *Information and Software Technology* 164 (2023), 107316. <https://doi.org/10.1016/j.infsof.2023.107316>
- [114] Jéssyka Vilela, Jaelson Castro, Luiz Eduardo G. Martins, and Tony Gorschek. 2020. Safety practices in requirements engineering: The Uni-REPM safety module. *IEEE Transactions on Software Engineering* 46, 3 (2020), 222–250. <https://doi.org/10.1109/TSE.2018.2846576>
- [115] Christine Wamsler. 2020. Education for sustainability. *International Journal of Sustainability in Higher Education* 21 (2020), 112–130.
- [116] Sven Weber, Roman Beck, Martin Wolf, and Jens Vykoukal. 2010. Portfolio performance measurement based on service-oriented grid computing: Developing a prototype from a design science perspective. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*. 1–10. <https://doi.org/10.1109/HICSS.2010.469>
- [117] Roel Wieringa. 2010. Design science methodology: Principles and practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)*, Vol. 2. ACM, New York, NY, USA, 493–494. <https://doi.org/10.1145/1810295.1810446>
- [118] Roelf J. Wieringa. 2014. *Design Science Methodology for Information Systems and Software Engineering*. Springer, Netherlands. <https://doi.org/10.1007/978-3-662-43839-8> 10.1007/978-3-662-43839-8.
- [119] Roel Wieringa and Hans Heerkens. 2008. Design science, engineering science and requirements engineering. In *Proceedings of the 2008 IEEE 16th International Requirements Engineering Conference (RE'08)*. 310–313. <https://doi.org/10.1109/RE.2008.63>
- [120] World Commission on Environment and Development. 1987. *Our Common Future*. Oxford University Press, Oxford. <http://www.un-documents.net/wced-ocf.htm>
- [121] Eric S. K. Yu, Paolo Giorgini, Neil A. M. Maiden, and John Mylopoulos (Eds.). 2011. *Social Modeling for Requirements Engineering*. MIT Press. <https://mitpress.mit.edu/books/social-modeling-requirements-engineering>
- [122] Jianping Zhang, Da Zhong, and Jiahua Zhang. 2010. Knowledge visualization: An effective way of improving learning. In *Proceedings of the 2010 2nd International Workshop on Education Technology and Computer Science*, Vol. 1. IEEE, 598–601.

Received 22 May 2023; revised 18 January 2024; accepted 25 January 2024