Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Interdisciplinary Extension of Requirements Engineering for Sustainable Software Design

Dissertation with the aim of achieving a doctoral degree **(Dr. rer. nat.)**
at the Faculty of Mathematics, Informatics, and Natural Sciences
Department of Informatics
of Universität Hamburg

submitted by

Yen Dieu Pham
from Brandenburg an der Havel

Hamburg, 2022

Day of oral defense:	22.12.2022
Chair of the examination board:	Prof. Dr. Tilo Böhmann
	Prof. Dr. Janick Edinger (deputy)
First reviewer:	Prof. Dr. Walid Maalej
Second reviewer:	Dr. Kim Lauenroth

To those who dare to expand their comfort zone.

Acknowledgements

After completing my architectural studies, I assumed I had overcome my biggest academic challenge. This turned out not to be true. While pursuing a doctorate, I engaged in one of the most challenging and rewarding endeavors of my life so far. This journey has allowed me to expand my perspective to an incredible extent. My being able to rise to this challenge is primarily a function of all the people who have supported me along the way. You have been and will continue to be the most significant source of my motivation!

However, facing this challenge would not have been possible without my first supervisor, Walid Maalej. Your courage to accept me as your doctoral student despite my different academic background has my deepest gratitude and respect. In response to your high demands, I was able to sharpen my thinking and enhance my research. It was my privilege to discover a new world through your efforts, enabling me to perceive myself as an individual with an intercultural and now interdisciplinary perspective. Also, thanks to Kim Lauenroth, my second supervisor. Thank you for your valuable feedback and the inspiring discussions about software design. Through your input, I gained a greater appreciation for my architectural background. Furthermore, I would like to thank all the researchers, practitioners, and study participants who have made interdisciplinary research projects such as mine possible.

I met my supervisor Walid Maalej thanks to my Schotstek community. Their honorary engagement creates encounters that may not otherwise occur. Schotstek has provided me with many precious opportunities. I hope to reciprocate by facilitating similar encounters and opportunities in the future.

Having an opportunity is one thing, but seizing it is another. Thanks to euroterra, I could continue to work as an architect while pursuing my doctoral degree. My euroterra colleagues enabled me to return regularly to my architecture “comfort zone” while following my curiosity to explore new horizons.

In addition, I express my gratitude to the Hans Böckler Foundation for its scholarship during my architecture studies. As a result of these studies, I gained many skills and insights, which ultimately prepared me well for my doctoral studies.

Furthermore, I am grateful to my MAST colleagues, our assistants, and my co-authors for the valuable discussions and collaborations. Special thanks to Davide Fucci, Christoph Stanik, and Carina Volkmer for their patience and the many pieces of advice. I thank Marleen Hillen for her support and involvement. I am deeply indebted to Abir Bouraffa and Clara Marie Lüders, who have made this time particularly worthwhile for me.

I also thank my Love, Peace, and Rock’n’Roll gang, my “adoptive” family, my Việt siblings, my in-laws, my ukulele group, as well as many other people who have accompanied and supported me during this time.

I am so grateful to my mother for the courage she displayed when she bravely ventured into the unknown and learned how to make it our home. Through your perseverance and sacrifices, I have received all these opportunities. Vietnamese Translation: *Tôi vô cùng biết ơn mẹ vì bà sự dũng cảm của bà khi dấn thân vào những điều mới mẻ để tạo ra ngôi nhà của chúng tôi. Nhờ sự kiên trì và hi sinh của mẹ, tôi đã nhận được những cơ hội tốt đẹp này.*

Finally, I would like to express my deep appreciation for my husband, Alex. Thanks to your belief in me and your sincere interest in seeing me grow, I learned and achieved more than I could have ever imagined.

Abstract

In recent years, consumer demand for sustainable products has increased, and political pressure has intensified. Thus, it might become of market relevance for software companies to design and offer sustainable products. Furthermore, the design of future sustainable software products allows us to leverage the already prevalent influence of software. Doing so might increase the chances of our next generation having a worthwhile future. In light of this, we propose an interdisciplinary extension of requirements engineering to promote sustainable software design. Our focus was on requirements engineering since requirements are integral to software design. They can have a crucial impact on whether a sustainable software product emerges.

However, some problems or goals may not yet have sustainable solutions. Hence, software companies may be required to create novel ideas to design sustainable software products. In this regard, the search for new solutions has the potential to drive innovation. As a means of capitalizing on this potential innovation driver, we have applied a psychologist's theory regarding *inspiration, communication,* and *perspiration* as relevant factors for innovation. According to these factors, we present three proposals, Dim_{SUST}, *Sustainability Poker*, and *Sustainability-Centered Design*, as an extension to requirements engineering.

To suggest this extension, we applied an interdisciplinary theoretical foundation. Along with requirements engineering, we also drew on insights from building architecture and linguistics. Using insights from linguistics, we investigated whether requirements engineering is affected by using natural language. We have chosen building architecture to benefit from its insights into the design of long-lasting products. In addition to the theoretical findings, we also conducted various studies to gain empirical insights. We learned that there is a lack of awareness of sustainable software design in the software industry. This unawareness might have, for instance, a negative impact on requirements identification.

Therefore, the first extension proposal includes our dimension set, Dim_{SUST}, which should assist project participants in identifying requirements and considering sustainable aspects. There are nine dimensions represented in this set: *ecological, social, individual, economic, technical, integrative, design-aesthetic, legal, and purpose*. We conducted several experiments and observed that our subjects could identify more requirements with our dimension set Dim_{SUST} than without any dimensions and consider sustainability-related aspects. The second proposal is our card-based estimation and communication procedure, *Sustainability Poker*. Applying our procedure, our study participants could estimate and present their views regarding the impact of a given requirement on a software product's sustainability. We created and evaluated our procedure, *Sustainability Poker*, in an iterative manner with four focus group discussions. The third suggestion consists of our framework,

Sustainability-Centered Design. We aimed to provide a framework that enables stakeholders to consider sustainability throughout the entire development cycle. We use our design framework *Sustainability-Centered Design* to illustrate how we could apply *Dim*$_{SUST}$ and *Sustainability Poker* in an iterative and five-level-based development process. Moreover, we describe the framework in terms of activities, artifacts, and goals which adapt according to the project's progress. In addition, we propose the role of the *Sustainability Design Master*, who is responsible for ensuring that the stakeholders remain mindful of the sustainability orientation of the product throughout the development process.

Kurzfassung

In den letzten Jahren hat die Nachfrage der Verbraucher nach nachhaltigen Produkten zugenommen, und der politische Druck hat sich verstärkt. Daher könnte es für Softwareunternehmen marktrelevant werden, nachhaltige Produkte zu gestalten und anzubieten. Darüber hinaus ermöglicht uns die nachhaltige Gestaltung zukünftiger Softwareprodukte, dass wir mit dem bereits heute omnipräsenten Einfluss von Software die Chance unserer nachfolgenden Generation auf eine lebenswerte Zukunft erhöhen können. Dementsprechend haben wir Vorschläge erarbeitet, wie wir Requirements Engineering erweitern können, um Softwareherstellende beim Design von nachhaltigen Softwareprodukten zu unterstützen. Wir haben uns auf Requirements Engineering fokussiert, da Anforderungen weiterhin das Softwaredesign maßgeblich mitbestimmen und damit einen wesentlichen Einfluss darauf haben, ob ein nachhaltiges Produkt hervorgeht.

Jedoch gibt es für einige Probleme oder Ziele möglicherweise noch keine nachhaltigen Lösungen. Daher müssen Softwareherstellende möglicherweise neue Ideen entwickeln, um nachhaltige Softwareprodukte zu gestalten. Die damit einhergehende Suche nach neuen Lösungsansätzen birgt somit ein innovationsantreibendes Potenzial. Um diesen potenziellen Innovationsantrieb zu nutzen, haben wir unserem Erweiterungsvorschlag die aus der Psychologie entnommene These zugrunde gelegt, dass für die Entstehung von Innovationen die Faktoren *Inspiration, Kommunikation und Perspiration* relevant sind. Gemäß diesen Faktoren schlagen wir vor Requirements Engineering um die drei Ansätze *Dim_{SUST}*, *Sustainability Poker* und *Sustainability-Centered Design*, zu erweitern.

Für die Erarbeitung dieser Erweiterungsvorschläge haben wir eine interdisziplinäre Theoriegrundlage angewendet. Diese bestand neben Erkenntnissen aus Requirements Engineering auch aus Beiträgen aus der Linguistik und Architektur (Hochbauplanung). Wir haben Erkenntnisse aus der Linguistik verwendet, um zu untersuchen, ob Requirements Engineering den Einflüssen der noch überwiegend verwendeten natürlichen Sprache unterliegt. Als weitere Disziplin haben wir die Architektur hinzugezogen, um von ihren Erkenntnissen in der Gestaltung lang genutzter Produkte zu profitieren. Zusätzlich zu den theoretischen Erkenntnissen haben wir auch verschiedene Untersuchungen durchgeführt, um auch empirische Erkenntnisse zu gewinnen. Wir haben anhand unserer empirischen Untersuchungen erfahren, dass in der Softwareindustrie möglicherweise ein mangelndes Bewusstsein für nachhaltiges Softwaredesign herrscht, was sich z.B. wiederum negativ auf die Identifizierung von Anforderungen auswirken könnte. Daher schlagen wir als erste Erweiterung unser Dimensionen-Set *Dim_{SUST}* vor, das Projektteilnehmenden bei der Ermittlung von Anforderungen und der Berücksichtigung nachhaltiger Aspekte unterstützen soll. Das Set umfasst folgende neun Dimensionen: *ökologisch, sozial, individuell, wirtschaftlich, technisch, integrativ, gestalterisch-ästhetisch, rechtlich*

und *zweckdienlich*. Wir haben mehrere Experimente durchgeführt und beobachten können, dass unsere Studienteilnehmenden mit unserem Set *Dim*$_{SUST}$ mehr Anforderungen identifizieren konnten als ohne Dimensionen und nachhaltige Aspekte berücksichtigt haben. Der zweite Erweiterungsvorschlag ist unser kartenbasiertes Einschätzungs- und Kommunikationsverfahren, das *Sustainability Poker*. Durch die Anwendung unseres Verfahrens konnten unsere Studienteilnehmenden ihre Ansichten über die Auswirkungen einer bestimmten Anforderung auf die Nachhaltigkeit eines Softwareprodukts einschätzen und präsentieren. Wir haben unser Verfahren *Sustainability Poker* in vier Fokusgruppendiskussionen iterativ entwickelt und evaluiert. Der dritte Vorschlag besteht aus unserem Framework, *Sustainability-Centered Design*. Unser Ziel war es, ein Framework zu schaffen, das es den Beteiligten ermöglicht nachhaltigkeitsrelevante Aspekte den gesamten Entwicklungsprozess zu berücksichtigen. Anhand unseres Frameworks *Sustainability-Centered Design* veranschaulichen wir, wie *Dim*$_{SUST}$ und *Sustainability Poker* in einem iterativen und fünfstufigen Entwicklungsprozess angewendet werden können. Darüber hinaus beschreiben wir *Sustainability-Centered Design* anhand von Aktivitäten, Artefakten und Zielen, die sich je nach Projektfortschritt anpassen. Des Weiteren schlagen wir die Rolle des *Sustainability Design Masters* vor. Diese dezidierte Rolle soll mitverantworten, dass die Projektteilnehmenden die nachhaltige Ausrichtung eines Softwareprodukts über den gesamten Entwicklungszeitraum nicht außer Acht lassen.

Table of Contents

1 Introduction

This thesis aims to extend requirements engineering to promote sustainable software design using interdisciplinary insights. We begin this chapter by establishing the importance of investigating the topic of sustainable software design and its relationship to innovation in Section 1.1. In Section 1.2, we describe the obstacles we identified that could impact the design of sustainable software products. We continue with our objectives and contribution to approaching these obstacles in Section 1.3. Lastly, we present the scope in Section 1.4 and the outline of this thesis in Section 1.5.

1.1 Motivation

Software products have become an integral part of our everyday lives. Due to their ubiquity, they have a significant impact on our environment and society [113]. According to Charter and Tischner [35], we can influence this impact by paying particular attention to the product design. They report that "The product design and development phase influences more than 80% of the economic cost connected with a product, as well as 80% of the environmental and social impacts of a product, incurred throughout its whole life-cycle." They argue that the product design has to play a crucial role since "it significantly influences the ways that products are produced and consumed [35]." Hence, our software design can influence, for example, how much energy an application consumes [204] and how people behave [101], [13].

If we encourage software companies to consider sustainability in their design decisions, we may benefit from the far-reaching effects of their products. Furthermore, by providing sustainable products, software companies can enhance their chance to remain competitive in the market for an extended period. After all, companies can experience existential consequences if they do not consider sustainability in their production. According to a report by the Economist Intelligence Unit (EIU) [54], commissioned by the World Wide Fund for Nature (WWF), the online search for sustainable goods increased by 71% between 2016 and 2020. Whelan and Kronthal-Sacco [209], from the NYU Stern's Center for Sustainable Business, reported that in the US, 50% of consumer packaged goods (CPG) growth from 2013 to 2018 was derived from sustainability-marketed products [209]. In addition, their findings show that the market for sustainability-declared products has grown 5.6 times faster than that for conventional products. The results suggest that consumers move from conventional products toward sustainable ones. Legacy companies might be able to cope with this shift if they consider sustainable aspects of their product [209].

Along with private consumers, public institutions also demand that companies consider sustainability when producing goods. For example, the United Nations (UN) emphasizes in their *Sustainable Development Goals* that we should strive for "responsible consumption and production" [196]. In the context of the *European Green Deal*, the European Commission is developing proposals and regulations for the economy to achieve its goal of making Europe the first continent to be carbon neutral [58]. Such external factors have likely encouraged companies in several industries to adapt and design alternative products [167].

As an example, sustainability has gained increasing relevance in the financial sector [106]. Investors today have the option of choosing financial products that emphasize sustainability. These products are labeled *Socially Responsible Investing (SRI)* or *Environmental, Social, Governance (ESG)* to indicate that the respective financial product has taken into account non-financial criteria, such as social, environmental, and governance aspects [207].

In the building sector, the demand for sustainability has encouraged the industry to explore, for example, alternative insulation materials [9]. Furthermore, research is concerned with alternative housing concepts, which, for example, enable intergenerational living to ensure the most sustainable and long-term use of a building [130].

Similarly, the software industry is not spared from the demand for sustainable products, as demonstrated by the search engine Ecosia. The company was founded in 2009 and invested 80% of the profits generated from its advertising business in tree planting initiatives [176]. In addition, the company runs its servers on electricity generated from renewable energy sources [131]. Nevertheless, Google still dominates over 90% market share worldwide [131]. Despite this, Ecosia has managed to attract enough consumers with its sustainable product to become the largest search engine in Europe [131].

These examples indicate that companies can contribute positively to our future and be inspired to design new products by considering sustainable aspects. Thus, sustainability also implies innovation potential, as discussed by Hargroves and Smith [85]. According to the authors, technological discoveries, the need to reduce costs, and the need to increase productivity were key drivers of the last five waves of innovation. As part of the current sixth wave of innovation, Hargroves and Smith identified another need, which is the need for sustainable products, as shown in Figure 2 in Subsection 2.2.2. Consequently, software companies should consider sustainable aspects in their product design to stay ahead of the current sixth wave of innovation and avoid being dragged down by it.

In summary, software companies can actively take responsibility for their impact by integrating sustainable aspects into their product design. By considering sustainable aspects, software companies may come up with new and innovative products that better meet the customers' future needs. Sustainable software design may enable them to remain competitive for an extended period. Accordingly, this thesis aims to assist software creators in designing sustainable software products.

To this end, we followed the approach of Duboc et al. [50] and Becker et al. [12]. According to Becker et al. [12], requirements are the "key leverage point" for sustainable software design. Similarly, Mahaux et al. argue [125] that we made crucial decisions during requirements engineering. As a result, these decisions affect how we design software, which in turn influences the behavior of humans.

Therefore, we suggest an extension of requirements engineering using interdisciplinary insights. However, to propose this extension, we first had to consider the following problems:

1.2 Problem Statement

The following section introduces the problems which we took into account. These issues are part of the software engineering field, focusing on requirements engineering.

Neglected Sustainability-Related Requirements

According to Méndez et al. [140], the most frequently mentioned problem in requirements engineering is incomplete requirements. Furthermore, they reported that incomplete requirements are among the main reasons for project failures. Eckhardt et al. [31] define incomplete requirements in two ways: "incomplete requirements specifications as a whole or incomplete requirements, i.e., lack of details for single requirements." Many previous scholars have tackled this problem of incompleteness. DeVries and Cheng [48] introduced their approach *Ares*, which they designed to automatically detect specific instances of incomplete requirements by using expressions from utility functions. Eckhardt et al. [31] focused on the completeness of performance requirements. The authors developed a framework based on a unified classification model for performance requirements from existing approaches, a content model, and sentence patterns to uncover incomplete performance requirements in industry specifications. Ferrari et al. [62] introduced a natural language processing tool, which supports the discovery of relevant requirements and their relationship to each other from input documents. Menzel et al. [141] presented a model-based approach for measuring the completeness of requirements specifications. They combined an information model with assignment rules and an analysis guideline.

In addition to developing approaches to deal with the problem of incompleteness, Albayrak et al. [2] have been looking at how developers deal with incomplete requirements. In their study, they observed that, on average, software engineers with a non-computer-related education made more explicit assumptions (in contrast to implicit assumptions) than engineers with a computer-related education while managing incomplete requirements and their respective stakeholders. The authors defined explicit assumptions as assumptions shared with and then reflected on by the respective users before implementing the requirements.

Alrajeh et al. [6] suggest that one reason for incomplete requirements is that project participants assume ideal project conditions during requirements identification. Consequently, the stakeholders do not consider possible obstacles and have no or just a few measures available when the project does not run as desired. Méndez et al. [140] report that due to the inexperience of the responsible requirements engineering teams, requirements are incomplete or identified later in the course of the project.

We can see from the previous research that the problem of incomplete requirements is very complex. Nevertheless, we need to address this problem because incompleteness, especially incomplete requirements as a whole [31], could also affect the identification of sustainability-related requirements. Since requirements are still predominantly formulated in natural language [124], the linguistic relativity hypothesis [108] could become applicable. According to this hypothesis, the applied set of requirements dimensions, also known as requirement categories, might affect which requirements, e.g., stakeholders can identify and which not. At present, the most common set of requirements dimensions assigns requirements into the dimensions functional and non-functional [27]. However, Curcio et al. reported that in practice, non-functional requirements are ignored [46]. This neglection is problematic as sustainability is often considered a non-functional requirement [157], [128]. Thus, the dimensions applied may be one factor among many others influencing the completeness of the requirements and the consideration of sustainable aspects during software design.

Lack of Awareness and Estimation Methods

Another common issue is communication, whether within the development team or between the development team and their clients [140]. Poor communication can be the lack of a shared understanding and awareness of the problem, different levels of knowledge, or different ideas about the possible consequences of solution proposals. In order to overcome those or similar obstacles [36] concerning sustainable software development, previous research endeavors have developed a wide variety of approaches.

Duboc et al. [50] found that engineers lack the knowledge, experience, and methods to engage in discussions on sustainability effects meaningfully. To resolve this, the authors developed a “question-based framework for raising awareness of the potential effects of software systems on sustainability.”

Lago et al. [112] developed a framework based on the dimensions *environmental, social, technical,* and *economic* to identify software qualities and their interdependencies within and across the dimensions. This process makes potential conflicts apparent, enabling stakeholders to discuss trade-offs. Furthermore, the framework facilitates consideration of the potential influence of stakeholders on the identified software qualities.

Seyff et al. [183] developed a similar approach. They proposed the *WinWin* negotiation model. The model should enable negotiations focusing on the effects of requirements on the sustainability of a software system.

Cabot et al. [31] have visualized interrelationships of different goals using the GORE modeling language i*. Their proposal demonstrates the potential conflicts and interrelationships between business-relevant goals and sustainability-related requirements. With the visualization, the authors support decision-making processes in which stakeholders become aware of the impact of their goals and can discuss trade-offs.

Brito et al. [26] pursue a similar goal with their concern-oriented requirements approach. They model sustainability concepts and present their interrelationships and potential conflicts with sustainability dimensions as well as with other system-relevant aspects.

All these approaches aim to create awareness and foster a discussion on how a system to be built relates to sustainability. Becker et al. [12] reported, "Making these effects visible is the first step to understanding and considering them in system design decisions." Even though a stimulating and broad discussion is an important prerequisite for designing sustainable software, only a few approaches support the estimation and prioritization of requirements effects on software sustainability, as Alharthi et al. [5] have suggested. By doing so, it may be possible to communicate better and manage the design of sustainable software products. This activity could have the advantage that project participants would better understand the possible effects of requirements and pay even more attention to designing sustainable software products.

Lack of Iterative Sustainability-Centered Design Processes

Besides effective communication, an appropriate development process is also crucial to sustainable software design. There are multiple approaches, such as the *waterfall*, the *V-*, the *spiral* or *agile* processes (e.g., Scrum) [173]. Since agile processes became the most popular software development processes today [95], we have focused on agile or rather iterative development processes. However, existing agile approaches have deficits. One way to compensate for these deficits is by expanding the processes, as seen in the agile-user-centered design approach [66]. Practitioners have established this integration since agile development approaches do not necessarily focus on the development of usable products [24], [63] [19], [41]. Sustainability may also be affected by this lack of guidance. According to Shamshiri [184], just a few studies have examined combining agile processes with sustainability objectives. Another issue is the presentation of the identified requirements. There is little guidance on representing requirements at the necessary level of detail according to the project's progress [182]. One of the most commonly cited problems is that developers perceive the representation of requirements as too abstract and underspecified [140]. This could cause developers to incorrectly implement insufficiently represented requirements, which may include sustainable requirements, or to not implement these requirements at all.

1.3 Objectives and Contribution

The thesis aims to qualitatively study requirements engineering approaches and investigate means to extend them for sustainable software design. To this end, we conducted an interdisciplinary research approach based on requirements engineering, psychology, linguistics, and building architecture insights. Using insights from other disciplines for software development is neither new nor unusual. Synergies between, e.g., software engineering and psychology, have long been widely accepted, as evident in the discipline of human computer interaction. Another example of a significant contribution from a different discipline is the pattern approach of the architect Christopher Alexander [3], [117]. His work had a substantial influence on the design of programming languages. In line with this history of integrating different disciplines, we summarize our contributions in the following paragraphs:

Interdisciplinary Insights on Designing Sustainable Products

We introduced an interdisciplinary knowledge base for extending requirements engineering to design sustainable software products. Since designing sustainable products harbors the potential to create novel ideas, we studied drivers and factors for innovation. We applied a theory from psychology that innovation needs *inspiration, communication* and *perspiration* to emerge [179], [81]. We conducted different exploratory studies based on these factors and the interdisciplinary insights. The objective of these studies was to evaluate the potential of common approaches for considering sustainability. Furthermore, we derived aspects we can consider for our requirements engineering extension to promote sustainable software design. One of these common approaches we investigated was design thinking. We conducted a workshop without any explicit instruction on sustainability. The workshop showed that participants hardly or never considered the issue of sustainability on their own. We compared the roles of a requirements engineer, product owner, and building architect by their liabilities, activities, self-portrayal, and artifacts. The analysis revealed similarities and differences between these roles. We found that, e.g., building architects incorporate sustainability dimensions like social and ecological into their set of requirements dimensions. According to our analysis results, we formulated hypotheses. One hypothesis was that stakeholders are more inclined to consider sustainable aspects if they address sustainability dimensions explicitly. Furthermore, we conducted semi-guided interviews with practitioners to learn about their concerns and chances associated with sustainable software design. Our interviewees reported about everyday challenges and other obstacles, like customer preferences. Nevertheless, they also suggested how we could overcome these obstacles to promote sustainable software design. Based on these theoretical and empirical insights, we prepared our requirements engineering extension. In Figure 1, we present our extension proposals to foster novel ideas and enable sustainable software design.

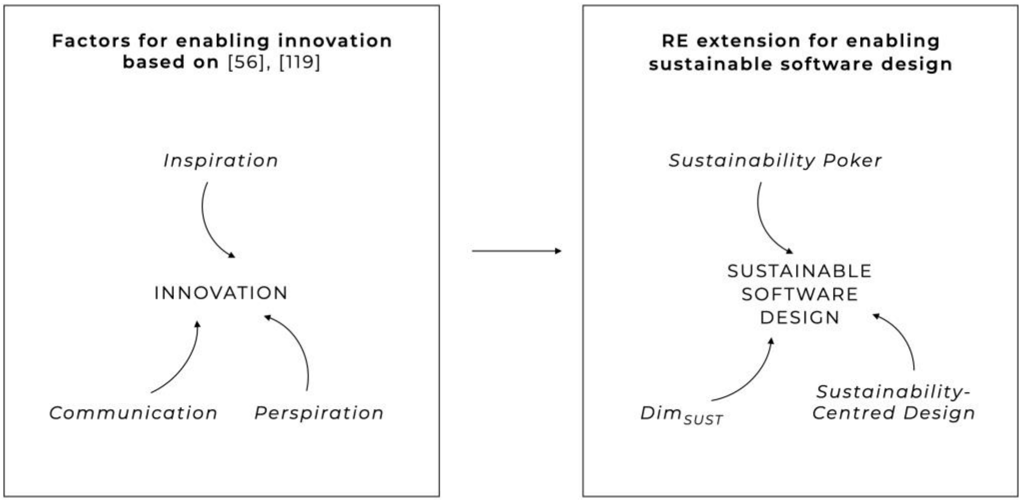


Figure 1: Proposed RE (requirements engineering) extension

Influencing Requirements Identification with Dimensions

Following the linguistic relativity hypothesis, we investigated the influence of requirements dimensions on identifying requirements. To this end, we evaluated established sets and our set of requirements dimensions, which includes the sustainable dimensions from the Karlskrona manifesto. In several experiments, we observed that our subjects, whom we provided with a set of dimensions, identified significantly more requirements than the subjects without any dimensions. We found indications that subjects found more sustainability-related requirements when applying requirements dimensions.

Estimating and Communicating Impact on Software Sustainability

Based on our semi-guided interviews, the most reported issue concerning sustainable software design is the lack of awareness and shared understanding. This issue could lead to communication difficulties. To address this concern, we prepared a card-based communication and estimation approach. The cards contain questions to estimate a requirement's impact on a product's sustainability and create a shared understanding. We iteratively evaluated our approach using focus group discussions. The final set consists of 41 cards. The study results show that our participants could estimate potential impacts and follow the other discussants' particular viewpoints.

Designing Process for Sustainability-Centered Software

Since the proponents of building architecture consider the field a durability-oriented discipline, we explored the design process of building architects. We derived insights and transferred them into our *Sustainability-Centered Design* framework. The resulting framework includes a description of an additional role and an iterative and level-based process. The process illustrates how the additional role can support the design and implementation of sustainable software design. The process consists of five levels. We aimed to represent a particular progress stage with each of these levels. We intended to support the additional role by providing this framework to assess the project's progress. According to the assessment, the role can initiate appropriate actions to advance the project. Possible actions may include using our requirements dimensions and our card-based communication method. To increase its likelihood of adoption into current development frameworks, we suggest the framework as an expansion of iterative development processes.

1.4 Scope

Solution Approaches

We can only determine retrospectively whether a product turns out to be sustainable or innovative [111]. Accordingly, we have suggested approaches that can at most increase the chances for sustainable and novel software products to emerge. We focused on examining what we could do from a requirements engineering perspective to enhance software design rather than what needs to happen during implementation. To this end, we investigated how existing approaches, such as the application of requirements dimensions, influence the design of software products. We intended to examine their impact on requirements identification to gain insights for promoting sustainable software design. Furthermore, we need to apply additional means to foster sustainable software design. For example, we focused our card-based communication approach only on the internal communication within an organization. At the same time, external communication is also essential to spread the product widely. We describe the potential application of our design framework but do not address how current education concepts would need to adapt to enable future software engineers to implement it.

Software Product

Our proposed extension addresses software as a product but does not discuss it as a service. We have not investigated our extension's effects on designing hybrid products, combining soft- and hardware components. We suggested our extension for designing software applications with graphical user interfaces. Furthermore, we aimed to suggest an extension that users who do not necessarily have a technical background can apply.

Designing Process

We only investigated how we can extend iterative development processes. We intended to propose inspiration and communication approaches for planning, designing, and testing related activities. We aimed to suggest a design framework project participants can integrate as a parallel strand to implementation, which also precedes one to two iterations.

1.5 Outline

The remainder of this thesis is structured in the following three parts:

PART I: PROBLEM

Part I consists of four chapters. In Chapter 2, we introduce our interdisciplinary foundation. We describe the socio-historical context of sustainability and its status today. Furthermore, we provide an overview of sustainable product design approaches in software engineering, requirements engineering, and building architecture. We present drivers and factors of innovation. We conducted exploratory studies based on these factors and the interdisciplinary insights regarding software design. In Chapter 3, we report on a design thinking workshop to evaluate whether design thinking approaches consider sustainable aspects. In Chapter 4, we performed a comparative analysis between the roles of product owners, requirements engineers, and building architects to identify potential synergies for sustainable software design. In Chapter 5, we report on semi-guided interviews with sixteen practitioners. We investigated practitioners' perceptions regarding sustainable software design and gained insights from their working experience to consider them in our extension proposal.

PART II: SOLUTION

Part II consists of three chapters. In the first two chapters, we explore and test the effects of requirements dimensions on identifying requirements. In Chapter 6, we introduce our investigated sets of requirements dimensions. We report on an exploratory study with two experiments. We aimed to determine whether the sets or individual participants' traits affect identifying requirements. In Chapter 7, we applied the lessons learned from Chapter 6 and report on our study with two other experiments. We evaluated the potential of dimensions for guiding requirements' identification and considering sustainable aspects. In the third chapter, we introduce our card-based communication and estimation approach. In Chapter 8, we describe our approach and the evaluation results of our focus group discussions.

PART III: SYNOPSIS

Part III consists of three chapters. In Chapter 9, we propose our framework for sustainable-centered software design. We present an iterative design process, including an additional role and its responsibilities. We suggest the framework as an expansion of iterative development processes. In Chapter 10, we discuss the limitations of this thesis. We conclude with Chapter 11 by summarizing our contributions and discussing future work.

PART I: PROBLEM

2 Foundation

In this chapter, we present our interdisciplinary foundation on sustainability in Section 2.1 and on designing sustainable software in Section 2.2. In Section 2.3, we conclude with the summary of this chapter.

2.1 Sustainability

In this section, we first describe the socio-historical context of sustainability in Subsection 2.1.1. We provide an overview of sustainability in software engineering in Subsection 2.1.2, requirements engineering in Subsection 2.1.3, and building architecture in Subsection 2.1.4.

2.1.1 Socio-Historical Context

The term *sustainable* was first introduced publicly by Carl von Carlowitz in his work *Sylvicultura oeconomica* on forestry in 1713 [206]. Von Carlowitz describes that we should cut down a forest only to the extent that the forest can regenerate naturally within a certain time. According to Pufé [167], von Carlowitz thus laid the foundation for understanding sustainability as a resource-economic concept. Even though the term sustainability was not officially used until about 300 years ago, it does not mean that this kind of sustainable action had not been practiced by people much earlier [84]. However, several centuries had to pass before the term sustainability became a political and thus social issue. Environmental problems in the northern part of the world and development crises in the south led to political debates on sustainability [55]. A famous result of such a debate is the *Brundtland Report*, also known as *Our Common Future*, published by the United Nations (UN) in 1987 [198]. The report introduces the following definition of sustainable development: "development that meets the needs of the present without compromising the ability of future generations to meet their own needs." Further debates followed, such as the United Nations Conference on Environment and Development in Rio in 1992, two other conferences in Johannesburg in 2002, and Rio in 2012. The latter was the starting point for providing development goals, resulting in the 2030 Agenda in 2015, including the *UN Sustainable Development Goals (SDGs)* [197]. The agenda expand the traditional view on the sustainability dimensions *social, environmental,* and *economic* by the dimensions: *peace* and *partnership*. The UN referred to these dimensions as the 5 P's: *People (social), Planet (environmental), Prosperity (economic), Peace* and *Partnership* [197]. The UN derived the seventeen *Sustainable Development Goals (SDGs)* based on these five dimensions. These goals should guide governments and companies on how to achieve sustainability. Furthermore, they should demonstrate the interdependencies of the dimensions, as we present in Table 1 based on Sow [42].

Despite the numerous efforts to define sustainability, there is still no agreed understanding [167]. It may be since different fields require specific definitions to operate effectively. According to Pufé [167] e.g., sustainable production in the economic context means more than just investing profits in environmental and social projects, but also generating them in a sustainable manner.

Table 1: UN Sustainable Development Goals [42]

	GOALS	DIMENSIONS
1	No poverty	Social + Economic
2	Zero hunger	Social
3	Good health	Social
4	Quality education	Social
5	Gender equality	Social
6	Clean water	Social + Environmental
7	Affordable clean energy	Environmental + Economic
8	Decent work and economic growth	Economic
9	Industry, innovation, and infrastructure	Economic
10	Reduced inequalities	Economic + Social + Environmental
11	Sustainable cities and communities	Economic
12	Responsible consumption and production	Economic
13	Climate action	Environmental
14	Life below water	Environmental
15	Life on land	Environmental
16	Peace, justice, and strong institutions	Peace
17	Partnerships for the goals	Partnerships

Conclusion 1. Sustainability has been a concern of humanity for centuries. Only in the last few decades have political leaders taken international measures to make our actions more sustainable. We can define sustainability with descriptions, dimensions, and goals. The definition of sustainability can vary according to the specifics of each discipline.

2.1.2 Sustainability in Software Engineering

Among the leading sectors within the industry is information technology [125]. Thus, the sector may significantly impact the environment [149] and society [113]. Consequently, software sustainability has been a subject of increasing interest in recent decades [203]. However, a generally accepted definition has yet to emerge [50]. Nevertheless, Venters et al. [203] distinguish the topic of software and sustainability into *software for sustainability* and *sustainable software*. The authors describe that *software for sustainability* covers *system production*, which focuses on the production resources, and *system usage*, which considers the environmental impact. Meanwhile, *sustainable software* includes all the aspects of the development process, which considers the "ecological, human and financial resources." Additionally, it also covers the *maintenance process*, which refers to continuous monitoring activities. Penzenstadler et al. [156] suggest a similar approach: "The term sustainable software can be interpreted in two ways: (1) the software code being sustainable, agnostic of purpose, or (2) the software purpose being to support sustainability goals." Koziolek [110] follows the approach of defining software sustainability based on its life span. The author suggests that a software system should run for over fifteen years. This approach is similar to the proposal of Calero and Piattini [33], who analyzed various definitions of software sustainability. They report that all their selected definitions include "the capacity of something to last a long time" and "the resources used." The Karlskrona manifesto [13] aims to create a shared understanding of software sustainability. It argues considering both the development process of a software product and its impact on our society and the environment. We follow this suggestion and argue that:

> *Sustainable software products emerge from a sustainable production and their impact contributes to a sustainable future [13].*

Following Venters et al. [203], software engineering research has primarily focused on providing reference models. The reference models should serve as a guide for software development and product evaluation to ensure sustainability. Penzenstadler and Femmer [156] suggest such a reference model. The model presents five dimensions, which, compared to other approaches, extend beyond technical and ecological considerations [113]. The Karlskrona manifesto [13] includes the dimensions of Penzenstadler and Femmer [156], as listed in Table 2.

To consider these dimensions, stakeholders should determine how to address them as part of their software design. Software design is essential in defining the product's purpose and requirements [193], [28], [213]. Based on our definition [163] and these research endeavors, we suggest the following definition of sustainable software design:

> *Sustainable software design pursues the development of software products which are useable for a long period [110] and are resource-conserving in their production and through their entire lifecycle [33] by considering different dimensions [202].*

Usually, requirements engineering can support this definition process [125], making it a crucial part of whether the software design includes relevant aspects of sustainability. Previous research endeavors have demonstrated how requirements engineering can contribute to sustainable software design, as described in the following section.

Table 2: Adopted Karlskrona manifesto dimensions [13]

DIMENSION	DESCRIPTION
Environmental	Concerned with the long-term effects of human activities on natural systems. This dimension includes ecosystems, raw resources, climate change, food production, water, pollution, waste, etc.
Social	Concerned with societal communities (groups of people, organizations) and the factors that erode trust in society. This dimension includes social equity, justice, employment, democracy, etc.
Economic	Focused on assets, capital and added value. This includes wealth creation, prosperity, profitability, capital investment, income, etc.
Technical	Refers to longevity of information, systems, and infrastructure and their adequate evolution with changing surrounding conditions. It includes maintenance, innovation, obsolescence, data integrity, etc.
Individual	Refers to the well-being of humans as individuals. This includes mental and physical well-being, education, self-respect, skills, mobility, etc.

Conclusion 2. Software engineering has been concerned with sustainability for decades as well. Different research endeavors propose definitions, goals, and dimensions according to their specifications. We define sustainable software design as both considering how to develop software and influence its future impact. Software design and requirements engineering are intertwined and crucial for considering sustainability in software development projects.

2.1.3 Sustainability in Requirements Engineering

According to Becker et al. [12], requirements are the “key leverage point” for sustainable software design. However, traditional requirements engineering has not focused on incorporating sustainability into software design [50]. In some instances, sustainability has been viewed as a non-functional requirement [71], [157] or as a quality attribute [125], [112]. Therefore, there is no official consensus regarding the relationship between requirements engineering and sustainable software design [50]. Nevertheless, several research endeavors indicate that requirements engineering can establish the aspect of sustainability at an early stage of a software design process [168].

Duboc et al. [50] suggest a question-based framework to initiate and guide discussions on the potential impacts of software systems on sustainability.

Lago et al. [112] introduce a framework to identify software qualities regarding sustainability and to reveal potential conflicts between them to enable stakeholders to discuss trade-offs.

Seyff et al. [183] follow a similar approach. They propose the *WinWin* negotiation model to enable negotiations that focus on the effects of requirements on the sustainability of a software system.

Cabot et al. [31] propose another negotiation method, which visualizes interrelationships and conflicts of business-relevant goals and sustainable requirements.

Using a concern-oriented requirements approach, Brito et al. [26] propose a way to model sustainability concepts and manage conflict situations. These situations can result from interactions between sustainability dimensions or interactions with other system concerns.

Using requirements engineering techniques may prove to be a promising approach to explicitly incorporating sustainability in software design. Besides software engineering, other areas, such as building architecture, are also concerned with designing sustainable products, as outlined in the following section.

Conclusion 3. Since traditional requirements engineering approaches do not explicitly consider sustainability, scholars propose to adjust these approaches and introduce new ones. These proposals aim to raise awareness of potential impacts and identify potential conflicts when designing sustainable software.

2.1.4 Sustainability in Building Architecture

Applying building architecture principles to software development is not novel. As an example, Christoph Alexanders' "language pattern" is still used as a reference for designing programming languages. Hence, we are studying building architecture to gain insights for designing sustainable software products. The discipline of building architecture seems especially appropriate since its proponents consider it "as a durability-oriented discipline, which has a holistic perspective of the end-product and the environment in which it evolves [150]" [163]. This description corresponds with Duboc et al. [49], who state that sustainable software design should consider "the holistic system within which the software-to-be will function." From these statements, we can conclude that both building architecture and software engineering are concerned with designing products holistically. The German Federal Ministry of the Interior, Building and Community published a *Guideline for Sustainable Buildings* [61] which defines the holistic perspective on buildings as follows:

> *"Buildings are complex systems that fulfil defined tasks and functions. They also provide both living space and a work environment, and they affect their users' well-being, health and satisfaction as well as the quality of social life. While they represent commercial and economic values and help to create added value, they also trigger energy and substance flows that affect the global and local environment" [61].*

As the previous definition indicates, designing sustainable solutions in building architecture is also a complex endeavor [200]. This discipline has to meet various conflicting needs between science and art [154]. Therefore, many research projects on sustainable buildings focus on a few specific aspects. Consequently, most publications on this research topic, similar to those in software engineering, are concerned with energy conservation or other ecological issues [208]. However, building architecture is committed to pursuing holistic solutions that are "well built, easy to use and beautiful" [105]. Therefore Keitsch [105] points out that building architects face the following challenges in designing sustainable buildings:

> "(1) Minimizing the negative environmental impact of buildings by enhancing efficiency and moderating the use of materials, energy and development space.
>
> (2) Developing measures to relate form and adapt the design to the site, the region and the climate.
>
> (3) Establishing a harmonious, long lasting relationship between the inhabitants and their surroundings by addressing the essence of good form-giving [1]" [105].

According to Blutstein and Rodger, designing sustainable buildings requires changing processes, systems, attitudes, and paradigms [20]. As a result, building architecture is also subject to constant change [150] and explores different approaches simultaneously. In terms of dimensions from which building architects can derive requirements, the *Guideline for Sustainable Buildings* [61] proposes three dimensions: *social and cultural, ecological,* and *economic*. However, the guideline divides these three into many smaller subdimensions. In comparison with the guideline, the *Official Scale of Fees for Services by Architects and Engineers (HOAI)* [187] lists eight dimensions, as presented in Table 3. Since the HOAI does not provide further details on the dimensions, we used the descriptions of the guideline to define the dimensions. These eight dimensions encompass four of the five dimensions of the Karlskrona manifesto [13] and include: *urban, creative, functional, technical, economic, ecological, social, public (legal).* As the official scale does not specify the dimensions, we based our definition on the *Guidelines for Sustainable Buildings* [61] and adjusted the wording of the dimensions to enhance comprehension, as presented in Table 3.

Table 3: Building architecture dimensions

DIMENSION	DESCRIPTION
Ecological	Aims to save resources by optimizing construction materials and products, reducing land use, maintaining, and promoting biodiversity, and minimizing energy and water consumption. [187]
Economic	"goes beyond the procurement and erection costs and focuses especially on the follow-up costs of a building. The emphasis is on building-related life cycle costs, economic efficiency, and value stability." [187]
Social	Considers "factors that influence the social and cultural identity of people and how they assess their environment", such as "health, mobility, and quality of life, as well as equal opportunities, participation, education, and cultural diversity." [187]
Technical	Focuses on the quality of the "technical features of the building and its equipment", e.g., for cleaning and maintenance issues, user and maintenance friendliness of building systems, possibility to demolish the building, and resistance to natural risks. [187]
Legal	Takes care that the building's design and the fulfillment of the customers' requirements consider standards and legal boundaries. [187]
Urban-Integrative	Covers requirements, which ensures that the building, its purpose, and its facilities "reflect the respective location and [...] the quality of its integration into the urban space." [187]
Design-aesthetic	Assures that the external and internal perception of the building ensure a high degree of user satisfaction, which shall have a positive impact on "the building's sustainability [...] and long-term value of the building." [187]
Purpose	Aims that the purpose of the building is thoroughly defined and sufficiently considered in the design. The purpose serves as a starting point and reference for any decisions. [187]

Conclusion 4. The design of sustainable products in building architecture is also a complex endeavor. The discipline pursues a holistic design approach between art and science. In this thesis, we investigate their eight requirements dimensions: *ecological, economic, social, technical, legal, urban-integrative, design-aesthetic,* and *purpose*. We chose these dimensions since they substantially overlap with the Karlskrona manifesto dimensions. Moreover, they include legal documents, indicating that they are essential to the design of sustainable buildings.

2.2 Designing Sustainable Software

In this section, we first introduce our software design fundamentals in Subsection 2.2.1. Since implementing sustainable software design requires novel ideas, we describe drivers and factors of innovation in Subsection 2.2.2. According to these factors, we present insights on how to inspire requirements identification according to the linguistic relativity hypothesis in Subsection 2.2.3. Continuing with card-based approaches in Subsection 2.2.4, we discuss how they can enhance communication and describe the relevance of adequate design processes to facilitate sustainable software design in Subsection 2.2.5.

2.2.1 Software Design Fundamentals

Software design can determine the outcome of a software development project. [213], [28], [193]. It can influence the functions the future system will provide and how it will perform [186]. The influential role of software design suggests that we should approach its importance for sustainable software from different perspectives. We can view software design as an activity [193], an intersection [213], a project phase [186] or result [23].

According to Tang et al. [193], software design is a highly complex and demanding activity. Software designers face the challenge of designing new systems that may have no predecessors and therefore cannot draw on previous experiences. Furthermore, according to Goldschmidt and Weil [77], software designers must move between the future and the past. On the one hand, they have to anticipate the future so they can decide on how to promote the progress of their product development. On the other hand, they must evaluate past decisions to determine whether the chosen direction will lead to a coherent product. Due to the lack of evidence pertaining to new designs, it can be difficult for software designers to predict whether a new design will be successful in the future. Therefore, Simon [185] suggests that there can be no one true solution. Instead, software designers provide several alternatives they may combine to solve one problem [142]. Proposing such design alternatives can be challenging. Software designers often cannot straightforwardly apply an existing theory to an existing problem [142].

To perform these complex activities, software designers must deal with various disciplines. Therefore, Winograd [213] describes software design as an intersection and cites the Association for Software Design as follows:

> *"Software design sits at the crossroads of all the computer disciplines: hardware and software engineering, programming, human factors research, ergonomics. It is the study of the intersection of human, machine, and the various interfaces – physical , sensory, psychological – that connect them" [213].*

According to Nadin [146], software design is not just an intersection. Additionally, it is an interdisciplinary field, as it draws insights from "psychology, sociology, communication theory, graphic design, and linguistics, among others." It is essential for software design to draw upon other disciplines to provide design proposals that can address "how people experience the software, what they do with it, and the larger situation in which they encounter it" [214].

Sonnentag et al. [186] take a similar position. Although the authors describe software design as an activity, they also assign it as a part of a sequence. According to the authors, software design receives requirements from preliminary analysis and then "translates" them so that a computer can convert them. However, as the authors point out, this "translation" is not so straightforward. Stakeholders may need to create requirements [96], or requirements may emerge as the project progresses. Furthermore, even if stakeholders have identified a reliable number of requirements, their sum may not result in a coherent software product without any additional effort. However, human beings must perform this additional effort to “translate” the requirements from the “human world” to the “machine world.” Therefore software design is “intensely personal, emotional, cognitive, social, and contingent,” according to Mishra [142].

Software design can be a complex and subjective endeavor. Depending on who executes it and what requirements it deals with, it can significantly contribute to sustainable software. To make this contribution, Charter and Tischner [35] suggest using software design to generate novel ideas as follows:

> *“To create sustainable products and services that increase stakeholders’ ‘quality of life’, while at the same time achieving major reductions in resource and energy use, will require a significant emphasis on stimulating new ideas through higher levels of creativity and innovation” [35].*

While designing sustainable software products, designers may be unable to draw on previous approaches. Appropriate approaches may not exist, or past approaches may cause harm to the environment and society. Thus, software designers may have to come up with original ideas that may eventually lead to innovative and sustainable software products. We can cultivate these novel ideas and this innovation potential by leveraging the profound influence requirements engineering has on software design. To this end, we describe what drivers and factors promote innovation in the following section. Based on these drivers and factors, we propose our requirements engineering extension for sustainable software design.

Conclusion 5. Due to the complexity of software design, it requires a variety of perspectives to describe it. Despite different approaches to defining software design, we can acknowledge the crucial impact it has on the outcome of software development projects. For sustainable software design, designers may need to create new ideas to compensate for the lack of previous examples.

2.2.2 Drivers and Factors of Innovation

There are various approaches to defining innovation [81]. Since we are interested in designing sustainable software products, we followed the approach that a novel idea is not yet an innovation. For example, Maier et al. [127] state that the simple creation of a new idea does not qualify as an innovation. Instead, they argue that it is essential that the idea is both usable and implementable. Guldin and Gelléri [82] agree that an idea is not yet a product that other people can use. Consequently, companies must invest in promising ideas without knowing whether their implementations will result in an innovation one day, as Hauschildt et al. [88] report.

According to Kremer et al. [111], it is crucial for companies to innovate to remain competitive. Thus, identifying the external drivers of innovation can be essential for companies, as Hargroves and Smith [85] describe. The authors report that innovation occurs in waves, as we illustrate in Figure 2. They state: “In order for a wave of innovation to occur, there needs to be a significant array of relatively new and emerging technologies and a recognized genuine need in the market that is leading to a market expansion” [85]. Hargroves and Smith [85] refer to the steam engine, semiconductors, and computers as examples of important inventions and discoveries in the past. Pressure to increase productivity and reduce costs were the driving needs. As shown in Figure 2, we are in the sixth wave of innovation. Unlike previous waves, the sixth wave is driven by both the need to increase productivity and provide sustainable solutions, as Hargroves and Smith [85] emphasize:

> “The past neglect of the importance of resource productivity offers significant opportunities to innovate and gain competitive advantage for those firms and nations that lead. The resource productivity gains, and product differentiation possibilities for firms through sustainable development, will complement and drive the next cycle of innovation” [85].

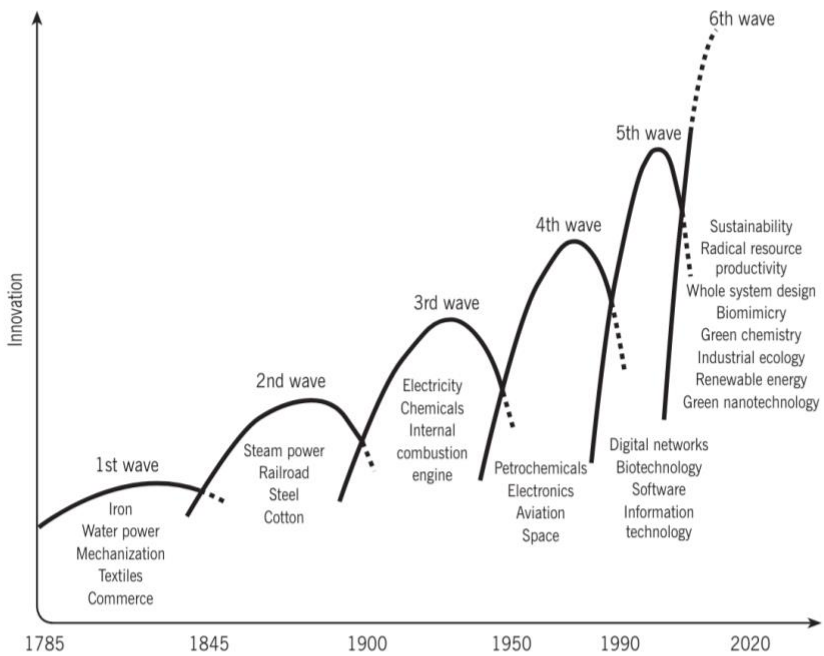


Figure 2: “Waves of innovation of the first and the next industrial revolution” adopted from Hargroves and Smith [85]

Software companies can benefit from these external drivers by influencing internal factors within their organizations. Schuler [178] observed: “Creative performance comes from inspiration to a small amount, from perspiration to a moderate amount, and to a high amount from self-confidence and social skills.” Following Schuler and Görlich [179], Guldin [81] summarizes that *inspiration, perspiration,* and *communication* are those internal factors for innovation. These factors extend the statement attributed to Thomas Edison: “Genius is one percent inspiration, ninety-nine percent perspiration” [171]. The extension of the factor *communication* suggests that mere implementation of an idea is insufficient. A critical mass of people must use the implemented idea before considering it an innovation. Until consumers use a product widely, it is necessary to inform and convince them of its benefits, which, in turn, requires effective communication. We adopt Guldin’s description [81] and summarize that in order for software companies to innovate, they require original ideas (*inspiration*), the ability to communicate both within and outside their organizations so that other people will support and adapt their ideas (*communication*), as well as perseverance to work on the feasibility and applicability of their ideas for an extended period (*perspiration*).

Overall, we can only determine whether an implemented novel idea turns out to be an innovation in retrospect [88]. Therefore, pursuing innovation may seem inefficient, but it can increase the likelihood of software companies remaining competitive. According to Hargroves and Smith [85], consumers are more likely to accept novel products that consider sustainable aspects. We suggest extending requirements engineering for software design to adapt to this sustainability-oriented need. We sought to leverage the profound impact of requirements engineering to enable sustainable software design. We proposed the extension according to the three factors *inspiration, communication*, and *perspiration*. We describe the insights that guided our extension proposal in the following sections.

Conclusion 6. The ability to innovate can be crucial for software companies to stay competitive. External drivers of innovation may include the need for productivity enhancement and sustainable solutions. Software companies can strive for innovative products by influencing their software design according to the internal factors: *inspiration, communication, and perspiration*.

2.2.3 Inspiration: Linguistic Relativity

The approach we have adopted states that innovation requires *inspiration*. However, it raises the question of what kind of *inspiration* enables sustainable software design. According to the Cambridge Dictionary [34], *inspiration* can be: “someone or something that gives you ideas for doing something.” This definition suggests that *inspiration* results from an external stimulus. Following the linguistic relativity hypothesis [108], our natural language can be an external stimulus.

Natural language is still the most widely used form of communicating software requirements. The linguistic relativity hypothesis [108], often attributed to the linguists Sapir [175] and Whorf [210], states that words and their associated mental representation may have a crucial impact on how people perceive their environment and reason about it [7]. The philosopher Ludwig Wittgenstein has already recognized in his work *Tractatus Logico-Philosophicus* [216] from 1922 the relation between language and perception: “The limits of my language mean the limits of my world.” However, this deterministic portrayal of language could not be proven so far, but according to Thierry [194] "cognitive neuroscience suggests that language and thought are intrinsically bound together, and thus that likely language influences thought." In this way, Amici et al. [7] conducted a study, in which they predicted how a participants' memory worked based on the word order of the participants' language. Boroditsky et al. [22] report that the grammatical gender of a word influences perception. They observed that the word *bridge*, which has a feminine gender in German and a masculine gender in Spanish, evoked corresponding associations in the participants. The German participants associated giving more "feminine" attributes, for example, elegant, while the Spanish participants tended to give more "masculine" attributes, such as strength. Similarly, the cognitive psychologist Boroditsky presumes that people can change their thoughts by changing how they speak [21].

The manifestations of the linguistic relativity hypothesis are diverse. They include the view that "language acts as a spotlight," according to which "thinking is directed towards properties highlighted by language," as outlined by Wolff and Holmes [218]. In the context of requirements engineering, linguistic relativity hypothesis would imply that structuring the "requirements language," e.g., based on requirements dimensions could influence the outcome of the requirements identification process. For instance, stakeholders might neglect certain sustainability requirements because they are unaware of the sustainability terminology in the requirements engineering context. The basic assumption is that requirements dimensions represent structures of the language used during the requirements identification, and resulting requirements represent the "worldview and behavior" of the stakeholders involved in this task.

Considering these findings, it appears that language can influence our thoughts, feelings, and values. According to the linguistic relativity hypothesis, the applied languages in the past may have influenced stakeholders to neglect requirements, as we outline in Section 1.2. However, Enfield [57] suggests that "a language, then, does not imprison you; it equips you. A language is not a straitjacket; it is an action suit." We have taken this approach and propose to structure the "requirements language" based on a set of requirements dimensions. As a result, stakeholders may be able to identify more sustainability-related requirements and reduce the risk of incomplete requirements.

Conclusion 7. The linguistic relativity hypothesis states that natural languages can affect our perceptions and thoughts. Since requirements engineering operates primarily in natural languages, it may limit or enhance requirements engineering activities. Depending on the language used, stakeholders may be able to identify more or fewer requirements, which may influence the consideration of sustainability-related aspects.

2.2.4 Communication: Creating Shared Perspectives by Cards

Communication is both important and complex. We use our communication, for instance, to avoid misunderstandings, ensure that others are aware of our needs, or satisfy our desire for recognition [191]. Thus, communication may be equally multifaceted when designing new and sustainable software products. Software design may require communication strategies to convey the potentials or consequences of those new products and their sustainability-related requirements, as they may not be able to rely on accumulated experience. We suggest a card-based approach to communicate and create a shared view on sustainability-related requirements. Lucero et al. [122] summarized the advantages of cards in three aspects: "cards are tangible idea containers, cards trigger combinatorial creativity and cards enable collaboration."

Examples of cards are the *Envisioning Cards* [67] which stakeholders can use "to raise awareness of long-term and systemic issues in design [67]" [147]. The *Behavior Change Design Cards* [109] is another example of cards that assists designers in choosing behavior-changing techniques in order to achieve their design objectives. As these cards assist in evaluating the impact of a design, others assist in organizing the implementation of that design, such as *Planning Poker*.

Grenning [79] introduces *Planning Poker* to assist expert teams in estimating a software task's effort [79], [4]. Cohn [39] describes the process of *Planning Poker* as follows: "At the start of Planning Poker, each estimator is given a deck of cards. Each card has written on it one of the valid estimates. Each estimator may, for example, be given a deck of cards that read 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100. The cards should be prepared prior to the planning poker meeting and the numbers should be large enough to see across a table." After discussing the respective software task, each team member has to select a value individually. After the decision, all team members reveal their chosen value simultaneously. Depending on the results, the team members agree on a shared value. This value as a basis can aid in estimating the number of resources needed to complete the task at hand. Although finding a compromise can be beneficial to the team, it can also take a long time to get to that compromise [68]. *Planning Poker* may require much time and expert knowledge, which ties up resources that an organization may not be able to give. Even though relying on experts' opinions seems to be risky, Moløkken-Østvold et al. [144] observed that *Planning Poker* could be more accurate than the combination of individual expert estimations. Furthermore, according to Mahnič and Hovelja [126], *Planning Poker* structures communication so that all team members can contribute equally while dominant characters have less influence on the outcome.

Conclusion 8. Designing sustainable software might require the evaluation of potential scenarios with little or no evidence. Therefore, stakeholders might have more difficulties sharing their views. However, a shared understanding can be crucial for the designing of sustainable software. Practitioners have positively experienced using card-based approaches to communicate their views and ideas.

2.2.5 Perspiration: Design and Implementation Processes

The factor “*perspiration*” refers to the long and continuous effort involved in designing novel products [178]. Implementing promising ideas can take a great deal of time and energy, and people require specific conditions and processes to continue working on feasible solutions. Designing sustainable software products requires a process that facilitates iterative work and explicitly considers sustainability throughout the life cycle [148].

There are already numerous proposals for iterative development processes. Contrary to popular narratives, the sequential approach *waterfall* is one of them. Introduced by Benington [14] in 1956 and adjusted by Winston Royce [172] in 1970. Neither Benington nor Royce call their processes *waterfall*. Royce describes the process according to the following seven stages: *system requirements, software requirements, analysis, program design, coding, testing, and operation*. However, the author also includes feedback loops to evaluate whether specific stages need to be repeated depending on the feedback result.

Another approach is the *V-model* process [64]. Like the letter V, this process consists of two streams that come together in the end. Both streams have stages and share the development stage at the end. Along the left stream are the stages: *define requirements, system architecture, detailed design* that deal with the requirements and design of the product. The right stream provides the stages: *build and test, system integration and test, deployment* and *verification*. Feedback loops connect the left and right stream stages to enable iterations if required.

Boehm [14] introduces the *spiral model* that presents the development process as a spiral in contrast to linear representations. However, this process also provides repeating stages that the project participants should pass through. These stages are: *determine objectives, alternatives, constraints; evaluate alternatives, identify, resolve risks; develop, verify next-level product; plan the next phases*. Furthermore, Boehm [14] suggests that prototypes should be available after each iteration to analyze potential risks.

In recent years, *agile software development* has become increasingly popular [95]. It includes various approaches, such as *Extreme Programming (XP), dynamic system development method*, and *Scrum* [95]. The advantages of agile approaches are their ability to react quickly to changes [95] and to offer a continuous development process [65]. For example, the approach *Scrum* achieves quick responses by breaking the development process down into short time intervals. *Scrum* teams carry out their development process in two-week sprints. A sprint consists of the activities: *sprint planning, creating a sprint backlog, conducting daily reviews*. After two weeks, the stakeholders discuss the results in the *sprint review*. Depending on the review, the development team must repeat the previous sprint to make improvements, or it can start a new sprint. There is, however, a risk that existing agile approaches will become overly technical. Consequently, there are suggestions to extend agile approaches, such as in *agile user-centered design* [66] or *agile requirements engineering* [177]. Extensions of this kind should compensate for the lack of user-friendly designs and inadequate requirements that result from agile approaches. The propagated minimal documentation [189] might have caused these issues. In addition, the widely used requirements documentation approach *user stories* is mainly textual [46]. As Wiegers and Beatty [211] based on Davis [47] point out, it is necessary to have "a combination of textual and visual requirements representations at different levels of abstraction to paint a full picture of the intended system." There are many approaches to supplement *user stories* [46]. However, there is little guidance regarding how to represent requirements at the appropriate level of detail and combine graphical and textual representations according to the project's current state [160]. Sustainable software design can benefit from an adequate representation guideline for iterative processes. According to Budgen [28], software designs' primary task is "to specify the best solution to a problem and produce a description of how this is to be organized. This description then forms a 'plan' that can be used by the eventual implementors of the system."

Conclusion 9. Iterative development processes, such as agile development, have become increasingly popular due to their ability to adapt quickly to changes. However, they have difficulties considering aspects that go beyond technical issues and do not provide guidance on how to adapt to the progress of a project. Thus, iterative processes require extensions, which ensure the consideration of sustainability during the design and implementation of software products.

2.3 Summary

The concept of sustainability has evolved from a special forestry topic to an internationally relevant principle. Sustainability should be incorporated into all aspects of life to achieve the greatest possible social change. As software products have a crucial impact on the world, they can contribute to a more sustainable future. Towards this end, we aim to promote sustainable software design and propose a requirements engineering extension based on the following summary of our foundation:

- Sustainability has become a global concern. It is possible to examine it from both a general perspective as well as from a discipline-specific perspective.
- There have been two approaches to studying sustainability and software design in software engineering. The studies discuss how we can sustainably produce software products or what impact software can have on sustainability. However, software design must consider both approaches so that software products can contribute to a sustainable future. To this end, the Karlskrona manifesto proposes to adapt software design according to the following five dimensions: *environmental, economic, social, individual, technical*.
- The use of requirements engineering significantly contributes to the design of sustainable software products. Previous researchers suggest several

approaches, e.g., to raise awareness and identify potential conflicts regarding the design of sustainable software products.

- Incorporating building architectural insights into software development has already produced positive synergies. This field strives to design long-lasting products. To this end, they design products based on requirements derived from eight dimensions. These dimensions coincide to a considerable extent with the Karlskrona dimensions: *ecological, economic, social, technical, legal, urban-integrative, design-aesthetic, purpose*.
- Software design is a complex undertaking that we can view as an activity, a process phase, and an interface. The design of sustainable software presents a particular challenge. As it can draw on little or no prior experience, novel ideas may be necessary. Nevertheless, these novel ideas may yield innovative products in the future.
- The ability to innovate can be crucial to a company's survival. We have taken the approach of defining innovation as an implemented and widely used product. External drivers, such as the need for increased productivity and sustainable solutions, stimulate the emergence of innovations. Software companies can leverage these drivers by internally influencing their design towards sustainability according to the factors: *inspiration, communication*, and *perspiration*. To promote the design of sustainable software products, we propose a requirements engineering extension based on these factors.
- The linguistic relativity hypothesis suggests that language can influence our perceptions and thoughts. Language can limit as well as expand our world of thoughts. By utilizing this influence, we may inspire stakeholders to identify requirements more effectively and consider sustainable aspects.
- The communication of sustainable software design may prove challenging. No prior experience may be available in some cases. To make design decisions, stakeholders must speculate based on their ideas and perspectives. If an interdisciplinary team is also involved, the perspectives may differ, making it difficult for the members to find common ground. Researchers observed that card-based approaches could establish a common communication framework.
- Iterative processes enable a rapid response to requests and continuous development. Nevertheless, it can become somewhat technical and provides very little guidance regarding how to adapt to the progress of a project. Current iterative development approaches may not consider sustainable aspects. Extending these development approaches with an additional framework could ensure that sustainable aspects guide the design and implementation of software products.

In the following chapters, we examine current approaches and the awareness of sustainable software in practice. We continue to explore what and how to inspire the requirements identification according to the linguistic relativity hypothesis. We propose and evaluate a card-based approach to communicate sustainability-related viewpoints among interdisciplinary teams. We suggest expanding iterative development processes with an additional and preceding sustainability-oriented design track.

3 A Design Thinking Study

Publication. This chapter is based on and expands our experience report *A First Implementation of a Design Thinking Workshop During a Mobile App Development Course Project* [164]. We focused on gaining insights for future creativity workshops with software engineering students. We examined the potential for such workshops to generate new ideas and consider sustainability-related aspects while designing software products. My contributions to this study comprise the literature study, the workshop design, facilitating the workshop, the analysis, and discussion of our findings, as well as leading the writing of the paper.

Contribution. We suggested a procedure lecturers can use to conduct a design thinking workshop with software engineering students. The analysis of our workshop provided us with lessons learned and suggestions for improvements. The results indicate that some guidance may be necessary to ensure that software project participants take the topic of sustainability into account.

3.1 Motivation

The design thinking approach has become increasingly popular in recent years. Proponents refer to it as an approach that supports solving “wicked problems” that are difficult to comprehend, have constantly changing requirements, are contradictory, and incomplete [169]. Lecturers have applied design thinking as a teaching approach in design schools and engineering [51]. According to Lindberg et al. [120], design thinking promotes the ideation of solutions, like services or products, that are “viable and novel for a particular group of users.” In order to meet our societal challenges, it is no longer sufficient just to create novel ideas. Additionally, these new ideas should consider sustainable aspects relevant to the future of our society and environment. This study examines how students use design thinking to develop new ideas and how they incorporate sustainable aspects into their ideas.

To this end, we conducted a design thinking workshop during M-Lab – a project-based course focused on mobile app development at the Universität Hamburg. In this course, students could develop apps for real customers from the industry. We intended the workshop to assist a students’ team that had difficulties finding a solution to their customers’ problems. Therefore, the workshop focused on generating ideas for a telecommunication provider.

We describe the design thinking approach of the d.school in Section 3.2 and report on the preparation and execution of the workshop in Section 3.3. In Section 3.4, we present our results and discuss lessons learned in Section 3.5. We present our related works in Section 3.6 and conclude with a summary in Section 3.7.

3.2 Background

In preparing for the workshop, we followed the design thinking approach proposed by the Hasso Plattner Institute of Design (also known as d.school) at Stanford University in 2005. Scholars have tested this approach in an engineering environment rather than a pure design setting [139]. We can understand design thinking as a framework with a human-centered approach to problem-solving [89]. Depending on the context, we can interpret design thinking as an innovation method, a working procedure, an attitude towards life, a mindset, or a tool [102]. The d.school defines the design thinking approach as a “constant-work-in-progress” framework of working modes and mindsets [86]. We describe the d.school working modes in Table 4 and the mindsets in Table 5. The working modes consist of five iterative steps [87], as presented in Table 4.

Table 4: d.school working modes based on [86]

MODES	DESCRIPTION
Empathize	In this mode, the designers should try to understand their users. The theory is that the problems of the users are not often related to the designers. Designers need to empathize with the users to design appropriate solutions that fit their needs.
Define	In this mode, the designers need a deep understanding of their users to identify their problems. Based on these insights, designers can “scope a specific and meaningful challenge” [86].
Ideate	The challenge defined in the previous phase represents the starting point for the designers to look for a solution. This process supports the designers in generating new ideas, they will have an explicit problem to solve and know where to start. Moreover, while searching for a solution, the designers will come up with new ideas.
Prototype	A prototype is defined as “anything that takes a physical form” [86]. It is useful to test functionalities, deepen the users understanding, inspire teammates, and explore more solutions.
Test	The testing mode gives the designer the opportunity to get feedback from the users. The latter will improve the idea and lead to new insights.

Besides an appropriate working procedure, applying the following seven mindsets are also crucial for a successful outcome. As presented in Table 5, these mindsets describe designers’ attitudes toward practicing design thinking [86].

Table 5: Adopted d.school mindsets [86]

MINDSET	DESCRIPTION
Show, don't tell	Communicate your vision in an impactful and meaningful way by creating experience, using illustrative visuals, and telling good stories.
Focus on human values	Empathy for the people you are designing for and feedback from these users is fundamental to good design.
Be mindful of the process	Know where you are in the design process, the methods to use in that stage, and what are your goals.
Bias towards action	The name design thinking is a misnomer; it is more about doing than thinking. Be biased toward doing and making over thinking and meeting.
Radical collaboration	Bring together innovators with diverse backgrounds and viewpoints. Enable breakthrough insights and solutions to emerge from diversity.
Embrace experimentation	Prototyping is not simply a way to validate your idea, but an integral part of the innovation process. Prototypes are built to think and learn.
Craft clarity	Produce a coherent vision out of complex problems, the problem needs to be framed to inspire others and fuel ideation.

3.3 Workshop

We describe in Subsection 3.3.1 our objectives, what we prepared for the workshop in Subsection 3.3.2 and how we collected our data in Subsection 3.3.3.

3.3.1 Objectives

We implemented the d.school approach as an intervention workshop within M-Lab, a semester-long project-based course at the department of informatics, Universität Hamburg (Germany). During the project, each of the five teams, consisting of five students, developed a mobile app for a real customer (e.g., the local university hospital or a telecommunication provider). According to the syllabus, the students should have generated *novel* ideas (to be later implemented) two months after the beginning of the course. Two teams were struggling to come up with original solution proposals. Hence, the teaching staff decided to intervene with a design thinking workshop. A mix of 11 bachelor's and master's students – four from the informatics curricula, five from information systems, one from software system development, and one from human-computer interaction – attended the workshop. They were all German native speakers. We interviewed the teaching staff before the workshop. They reported that the students had issues coming up with original ideas and that rather than trying to create something new or remarkable, they were more concerned with meeting all the formal criteria to pass the course (e.g., writing a problem statement, developing a clickable prototype). According to the teaching staff, the team facing the most difficulties was the one that did not get any specific requirements from their customer, a telecommunication provider. The customer gave them complete freedom, as long as they would deliver an innovative mobile app. Instead of being inspired by the possibilities, the students felt overstrained and clueless. Based on these insights, we wanted to achieve two types of goals, internal (not communicated to the students) and external (presented to the students), as shown in Table 6.

Table 6: Workshop objectives

GOALS	DESCRIPTION
Internal	• Inspire creativity of the students • Improve their confidence and enable them to come up with ideas in a limited amount of time • Help the group struggling the most without exposing them as "weak" to their peers • Provide each team with valuable experiences from their progress • Do not mention sustainability to observe whether the students consider it by themselves
External	• Reflect on the current state of their project • Create new ideas as well as concrete suggestions for their implementation

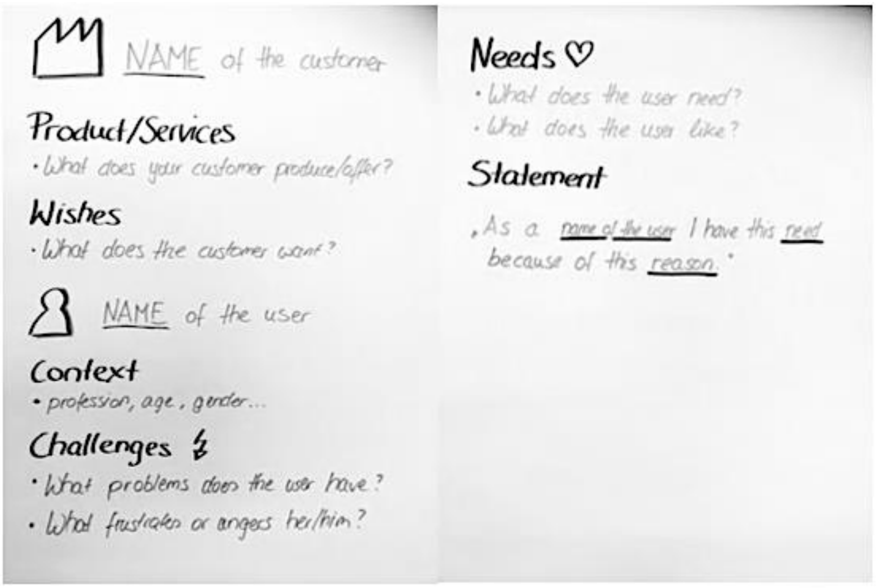


Figure 3: Empathize mode template

3.3.2 Preparation

The workshop took place at the Universität Hamburg. We conducted the workshop in English. We implemented it in a room with space for 30 people. The materials used were pinboards, a flip chart, sticky notes, markers, DIN A3, and DIN A0 sheets. The author of this thesis acted as the facilitator and put the workshop into practice. The workshop lasted two hours and consisted of four phases and two transition phases, as shown in Table 9.

Introduction. The facilitator and students introduce themselves and tell each other about their backgrounds.

1. **Empathize.** Every team reflects their previous progress by using a template, shown in Figure 3, and should realize if they understood and covered the needs and problems of their users. This phase should be used as a starting point to build empathy for the users.

   Transition Phase I. The facilitator chooses an app project in agreement with the students. In accordance with the teaching staff, the students will focus on the telecommunications app.
2. **Define.** The facilitator collects the students' experiences as smartphone users and customers of a telecommunications provider. The facilitator clusters the experiences and closes this phase with a clear scope focusing on three topics.

   Transition from Define to Ideate. The facilitator prepares students to develop an open-minded mindset with an adjusted version of the improvisation game "Yes, but…/ Yes, then let's…", as presented in Table 7.
3. **Ideate.** The students create ideas with the *World Café* method reported in Table 3. During each round, they fill in the template shown in Figure 5.
4. **Prototype and Test.** The students transform their theoretical ideas into paper prototypes. These prototypes should serve as inspirations and suggestions on how to implement the ideas.

   Debriefing. Summary of the results, workshops, and learnings. The facilitator asks for feedback from the students.

3.3.3 Analysis

We based our analysis on observations we made during the workshop, feedback after the workshop, as well as an evaluation of the artifacts produced by the students. To conduct our data collection and analysis, we loosely followed the guidelines by Creswell and Creswell [45].

During the workshop, we took notes on parts of dialogues, accounts of particular activities, or unusual settings. We had a feedback discussion with the students of the telecommunication provider project. We examined the students' understanding and application of our instructions using the artifacts. We counted the number of ideas and discussed their content. We excluded ideas that, e.g., were not readable.

We compared our observations and reported the ones we could agree on. We derived *lessons learned* from the workshop *observation* for future design thinking workshops.

Table 7: Instructions of improvisation game "Yes, but…/ Yes and then let's…"

PHASE	DESCRIPTION
Preparations	• Each student chooses a partner whom she or he never really interact before. • The students should get used to work with unknown partners.
Round 1: "Yes, but…"	Student A thinks about a destination, where she or he would like to travel. Student B has to find arguments against the ideas of Student A and should start her/his sentences with: "Yes, but…" The dialogue should last about 2-3 minutes. Example: • Student A: "Let's travel to Hawaii" • Student B: "Yes, but it is so far away" • Student A: "Maybe, but it is sunny, and we can relax at the beach." • Student B: "Yes, but I will probably get a sunburn and have to stay in the shadow the rest of the journey" • Student A: …
Round 2: "Yes, and then let's…"	Student A starts with the same destination. Student B has to build on the ideas of Student A, and should start her/his sentence with: "Yes, and then let's…" The dialogue should last about 2-3 minutes. Example: • Student A: "Let's travel to Hawaii" • Student B: "Yes and then let's spend some time on the beach" • Student A: "Exactly and go swimming" • Student B: "Yes and then let's rent a boat to visit all the small islands" • Student A: …

Table 8: Procedure *World Café*

PHASE	DESCRIPTION
Preparations	• The students spread equally according to the number of tables
Round 1:	• Each table team creates ideas or solutions based on the topics of the "Definition" phase
Transition I:	• Each team chooses a "table master." • The "table master" has to stay • The other students spread individually to a new topic/ table
Round 2:	• The "table master" presents the results of the previous group to the new members • The new team creates ideas or solutions based on the results of the previous group
Transition II:	• The first "table master" has to leave the table and the team decides on a new one • The other students spread individually to a new table again
Round 3:	• The new "table master" presents the previous ideas to the new team • Last round of ideation based on the ideas of the previous teams

Table 9: Design thinking workshop procedure

PROCESS	GOALS	MATERIALS	GROUP SIZE	TIME
Introduction				
• Facilitator and students introduce themselves (name, background, app team) • Flipchart presentation of the goals and schedule (displayed during the duration of the entire workshop)	• Creating a trustful atmosphere • Involve students, so that they are more likely to express their ideas by knowing each other's name and background. • Being transparent about the procedure	• Tape name tag for the participant • Flipchart and -paper	• 1 group, all students	• 5 min.
1. Empathize				
• Facilitator presents a template with questions about their customer and users (see Figure 4)	• The students should recap their current progress and what they might have missed	• Flipchart-paper • Markers	• 5 groups or according to number of projects	• 10 min. incl. presen-tation
Transition Phase I				
• In agreement with the students, the facilitator decides to focus on the telecommunication team	• Focusing on the telecommunication team without putting them in a difficult spot • Creating a clear scope	• -	• 1 group, all students	• 5 min.
2. Define				
• Brainstorming: The facilitator asks the students about their experiences with their telecommunication provider. • The facilitator writes down each experience on a sticky note	• Gathering needs and challenges, which should inspire students to create solutions/new ideas	• A0 sheets • Pinboard • Sticky notes • Markers • Sticker dots for voting	• 1 group, all students	• 8 min.
• Clustering: The facilitator arranges the sticky notes in clusters on the board while brainstorming with them	• Getting an overview of the topics and preparing for the voting			• 4 min.
• Each student votes for their most interesting topic/cluster	• Defining a clear goal			• 3 min.
Transition Phase II				
• Improvisation Game: "Yes, but..." / "Yes and then..." (see Table 2)	• Making students realize how important it is to be open-minded and supportive towards teammates to create new ideas	• -	• Dividing group in pairs	• 5 min.

3. Ideate				
• Apply World Café [12] (see Table 3): • Facilitator puts the top three topics on different tables with a template (Figure 5) • Students divide equally onto the tables • Students rotate topics and teammates three times	• Creating ideas based on other people ideas • Collaboration with different people • Students realize that too much discussion in the ideation phase prevents new ideas	• DIN A0 sheets • Markers	• Dividing group by topics	• 30 min.
4. Prototype and Test				
• First round of prototyping • The facilitator hands out a paper with smartphone outlines • The students suggest wireframes based on the results of the World Café	• Creating suggestions and inspiration for concrete implementations	• DIN A3 sheets with smart-phone outlines	• Previous groups	• 10 min.
• Apply feedback method "I wish/I like": • Teams present their prototypes • Students comment on prototype what they like/what they wish to change • Facilitator writes down the feedback.	• Getting constructive feedback from the other groups	• sticky notes	• 1 group, all students	• 15 min. (each prototype 5 min.)
• Second round of prototyping: • Adapting prototype to the feedback • Repeat presentation	• Realizing that early feedback can be very helpful	• DIN A3 sheets with smart-phone outlines	• Groups of first round	• 10 Min.
• Second round of feedback • Repeat feedback method	• Getting constructive feedback from the other groups	• sticky notes	• 1 group, all students	• 15 min. (each prototype 5 min.)
Debriefing				
• Facilitator sums up the results of the workshop	• Building up students' self-confidence by stressing that they accomplished a lot in a short amount of time • Inform students that their mindset is important to be creative	• All the results on Pinboard • Flipchart	• 1 group, all students	• 3-5 min.
• Feedback: • The facilitator gives students a small ball • The students tossing the ball to one another and giving feedback	• Collecting suggestions to improve future workshops			• 10 min.

3.4 Results

The workshop had the effect of influencing the students' perspective toward a human-centered approach. At last, they based their final ideas on the needs of a potential user. From the educator's perspective, the biggest challenge was deciding on methods aligned with the internal and external goals, the mindsets of the d.school, and the available amount of time. We report our most salient observations gathered during and after the workshop in Subsection 3.4.1 and the evaluation of the students' artifacts in Subsection 3.4.2.

3.4.1 Workshop

During the Workshop

Observation 1. We observed that the template description was not specific enough, as shown in Figure 5. In particular, some students described their users in general, and others described a personification of the user. The students were confused by the term "name," which meant for some groups a real name like "Jane" and not the general name of their user group. Rather than creating personas, we wanted the students to take the viewpoint of their typical user group. By examining a user group, students should gain a broad understanding of the problems and needs of that group. A persona could be too narrow and misleading if the students did not interview their customers and final users.

Observation 2. At the beginning of the definition phase, we were worried that the students would not engage with the workshop. We were not sure if they could identify needs and problems. The student's academic backgrounds were quite similar in contrast to the mindset of *radical collaboration*. Nevertheless, it was a productive phase. Three students did not contribute, and the students we associate with a more creative background (e.g., in HCI/UX) provided half of the discussion topics.

Observation 3. During the *World Café*, some groups took too long to discuss their ideas without writing them down. In order to prevent the students from talking for too long, the facilitator reminded them about the time limit.

Observation 4. The students were surprised by their productivity in such a limited time. They expressed positive feedback about the strict time frame of the working sessions, which forced them to focus on the essential parts of their ideas. They suggested that the workshop should take place at the beginning of the course rather than halfway through it.

After the Workshop

Since we aimed to support the students with the telecommunication provider, we focused on their feedback in this section.

Observation 5. The team had four members. Only two of them participated in the workshop. It became apparent that their teammates did not want to engage with the ideas that emerged during the workshop. They did not perceive the workshop results as applicable because it did not have a specific goal and was just about creativity. However, after the customer expressed dissatisfaction with the team's technically motivated ideas (e.g., a chatbot to improve customer support), they were motivated to work on the workshop ideas.

Observation 6. The team did some research afterward. They realized that the customer had already implemented most of the ideas generated during the workshop.

Observation 7. The team stated that their final app idea did not originate from the workshop. However, we documented a similar idea from the workshop results.

3.4.2 Artifacts

Observation 8. Using our template, the students could distinguish between the perspective of the user and that of the customer. E.g., they identified the need for a proper signal from the user and the need for a good image from the customer.

Observation 9. During the group brainstorming session, students collected 22 needs and problems they had with their telecommunications provider. We then categorized these into the following groups: *behavior, streaming, device, contract, customer support, internet,* and *feedback*. Because of the limited time, the students decided to work on behavior, streaming, and contract topics.

Observation 10. The students completed each field of the *World Café* template. However, the number of ideas decreased with each round. For example, in the first round of the *contract* cluster, the students came up with seven different ideas, followed by three, and finally, only one. The students did not add any new ideas for the other two clusters in the third round. In Table 10, we summarized the needs and problems of the three clusters and the ideas from the *World Café*-method.

Observation 11. We did not observe the consideration of sustainable aspects during the group work nor in the artifacts. The students mainly focused on features or functional requirements.

Observation 12. In the prototype phase, the students largely adopted the *World Café* ideas and came up with wireframes.

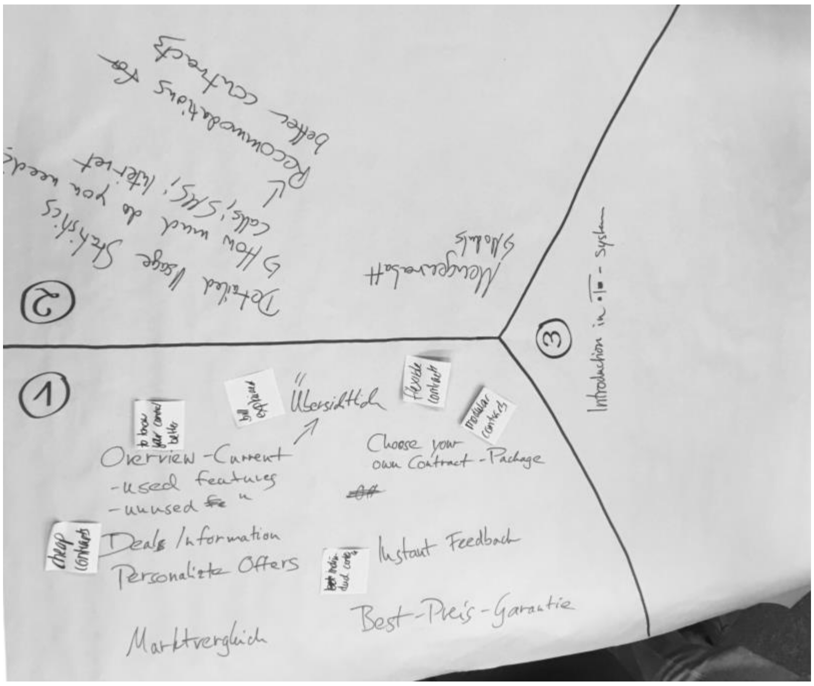


Figure 4: World Café results of the cluster contract

Table 10: Artifacts results

CLUSTER	NEEDS & PROBLEMS	IDEAS
Behavior	• Statistics user behavior • More data on demand	• Historical overview (visualization, via e-mail) • Monthly gifts (birthday, data, streaming) • Recommendations (international, different devices • Gaining awards (gamification, challenges) • Upgrading (payment, modularity data volume)
Streaming	• Movie streaming • Flexibility • Bad quality	• Pre-download • Statistics • Connectivity detection • Information about the partners • Intelligent speed • Automatic hotspot login • Showing hotspots
Contract	• Knowing contract better • Contract expired • Cheap contracts • Individualized contracts • Flexible contracts • Modular contracts	• Overview of used or unused features • Deals information • Personalized offers • Provider comparison • Choosing own contract-packages • Instant feedback • Best-price-guarantee • Detailed usage statistics • Recommendation for better contracts • Volume discount • Introduction provider-ecosystem

3.5 Discussion

The workshop we have applied can be used as an inspiration workshop at most. We conducted the workshop for the first time. Despite our efforts to remain in the background during group work time, we cannot completely exclude the possibility that the students found us disturbing. The students appeared very absorbed in their group work, but we cannot rule out that our observations may have been distorted by their perception of being observed. While creating the artifacts, not all students contributed with their texts or graphics, but only as discussants. Consequently, the artifacts may contain the thoughts of their creators primarily. Students with limited drawing skills or difficulties with the English language may not have been able to incorporate their thoughts into the artifacts. Nevertheless, our observations can serve as a starting point. We have derived lessons learned from our observations and developed suggestions for improvement as follows:

Lessons Learned 1. Be more careful with ambiguous descriptions of the material.

Suggestions 1. The template we used in the empathizing mode, as presented in Figure 3, should explicitly state the characterization of the users. For example, express the naming of the user group as an activity rather than a noun.

Lessons Learned 2. Not every student is used to expressing her or his ideas openly nor able to develop them in a group.

Suggestions 2. Before starting the open discussion, every student should get a few minutes to think and develop their ideas individually.

Lessons Learned 3. It was challenging for the facilitator to keep an eye on every single group, track the time, and ensure that the groups would not get stuck discussing details. Therefore, the students should do some part of the group management.

Suggestions 3. At the beginning of each round, the students should assign roles, like *writer* and *timekeeper*, to remind themselves of their tasks and focus.

Lessons Learned 4. Although the students experienced an unusual approach compared to their previous studies, they were fond of the workshop and engaged with the activities. They even suggested to start the course with a design thinking workshop.

Suggestions 4. Implementing the design thinking workshop not just as an intervention but as a regular activity at the beginning of the course.

Lessons Learned 5. Acceptance is critical to implementing new ideas. The students need to understand and experience the ideation to implement the ideas.

Suggestions 5. All members of a team must be involved in the workshop so that each of them has the chance to contribute with her or his idea(s).

Lessons Learned 6. While the students were halfway through the project, they could still not get an overview of the products/services of their customers. This can be due to improper research by the students or/and due to the big size of the company.

Suggestions 6. Involve the customer in the workshop or ask them for validation (e.g., test mode) to get faster feedback and to avoid redundant ideas. Planning an "icebreaker" – e.g., a game to prevent biased behavior due to the different roles (e.g., customer, student, teacher).

Lessons Learned 7. Students can have difficulty accepting that their project idea originated from a context in which students from other teams were involved. This indicates low acceptance of shared ownership of ideas.

Suggestions 7. In case of an intervention workshop, the facilitator should ask the students beforehand about their ideas and experiences to avoid redundancy. To that end, the facilitator can prepare an ideation phase for every team to increase the acceptance of shared ownership of ideas. The students should acknowledge those side effects of project-based learning (e.g., a team crisis) as learning opportunities.

Lessons Learned 8. The participants do not consider sustainable aspects.

Suggestions 8. The resulting ideas suggest that students do not necessarily take sustainability into account. Therefore, we recommend addressing sustainability explicitly. For example, workshop facilitators could use templates during ideation with sustainability-related dimensions that participants can use to develop ideas.

3.6 Related Works

The literature refers to design thinking as a method for solving *wicked problems* [158], [90], [169]. Thus, researchers have explored how design thinking can assist in resolving wicked problems relevant to sustainability. A case study on efficient energy use by Newman et al. [151] indicate that design thinking can contribute to developing solutions to open-ended and fuzzy problems. In an environmental undergraduate engineering course, Clark et al. [37] observed that students found positive associations with design thinking in terms of creativity and the ability to find solutions for sustainability-related problems. Buhl et al. [29] suggest that design thinking can facilitate the development of sustainable innovations. It helps define the scope and strongly focuses on user needs. Its iterative approach can also ensure the product's positive impacts on sustainability.

Research in requirements engineering has also explored how they can apply design thinking to their wicked problems. Vetterli et al. [205] report that requirements engineering seeks to develop large systems that usually do not have quickly changing demands. As a result, it has difficulty adapting to a rapidly changing world with volatile needs and diverse stakeholders [89]. Hehn and Uebernickel [90] see in the combination of "the strongly human-oriented working mode of DT (design thinking) with the more formal, technology-driven world of RE (requirements engineering)" the potential "to reduce these difficulties and find innovative solutions." Given this perspective, Pereira et al. [158] observed in their study that design thinking can help identify requirements and understand the real user needs. Hehn and Uebernickel [90] report that design thinking contributes to structuring the requirements identification and can be well matched with requirements engineering practices. However, the authors also notice that design thinking is mainly usability-focused and tends to neglect non-functional requirements such as performance or security. Since design thinking has its limits, Lauenroth [114] recommends applying design thinking to small and clearly focused projects. Furthermore, Lauenroth [114] states that design thinking enables the creation of ideas but does not guarantee their implementation. These observations are similar to our workshop results.

In summary, we can say that design thinking can be constructive in generating ideas. Combining it with requirements engineering can produce synergistic effects. However, stakeholders do not consider sustainable aspects per se with design thinking if the problem is not explicitly sustainability related.

3.7 Summary

We conducted a design thinking workshop with eleven bachelor and master students with an informatics-related background. Our goal was to observe how students react to such a workshop and whether they consider sustainability-related aspects in their ideation. We prepared and conducted the workshop using the approaches of the d.school. We provided a detailed description of our procedure. Our results indicate that the students were indeed able to develop ideas from a user-centered approach, but they did not consider sustainability-related aspects. The students' ideas were very feature-oriented. From our observations, we derived lessons learned and made suggestions for improvement for future design thinking workshops. We reported on related works about design thinking and sustainability, as well as requirements engineering and its integration of design thinking approaches.

4 A Comparison of Requirements Engineering and Building Architecture

Publication. This chapter is based on our exploratory comparison, *Renovating Requirements Engineering* [165]. We focused on gaining insights from an interdisciplinary role comparison of requirements engineers, product owners, and building architects. My contributions to this study comprise the research design, the literature study, the analysis, and discussion of our findings, as well as leading the writing of the paper.

Contribution. The comparative study is one of the first of its kind. It reveals similarities as well as differences between the disciplines, from which we were able to develop hypotheses inductively. The hypotheses served two purposes. First, they provide guidance on how requirements engineering could be adapted to improve software design. Second, they served as starting points for the following studies of this thesis.

4.1 Motivation

The application of other disciplines to software development has proven beneficial in the past [117], [92]. Therefore, we compared requirements engineering and building architecture. To this end, we compared the roles responsible for managing requirements in these disciplines since researchers consider requirements the “key leverage point” [12] for implementing sustainable software.

As a discipline that emphasizes durability, we chose building architecture [150]. Additionally, it takes a holistic view of the end product and its surrounding environment [150]. This perspective coincides with the idea that sustainable software design should take into account “the holistic system within which the software will operate” [49]. The Federal Association of German Architects [60] reports that each country defines the profession, scope of work, liability, and training differently. For instance, German building architects provide construction management, cost control, bidding, and design services. In contrast, American building architects are typically only responsible for the design of the building.

For requirements engineering, we selected two roles: requirements engineer and product owner. Ideally, requirements engineers are responsible for monitoring, analyzing, and identifying stakeholder needs and constraints and initiating system updates when these requirements change [166]. Since agile development processes, like *Scrum* [181], have become one of the most popular software development processes today [75], we decided to examine the *Scrum* product owner role. A product owner is solely responsible for maintaining the backlog, which consists of action items representing the project's requirements [181]. Nevertheless, it is unclear how the Scrum team acquires the action items and whether they comply with previous decisions or consider sustainability-related aspects. Although agile practices were initially designed for small, co-located, well-educated, and empowered teams, they are now widely popular in every context [8] – even for large or legacy projects [9]. We analyzed and compared regulatory documents for requirements engineers, building architects, and agile product owners to stimulate a conversation around possible synergies.

In this chapter, we describe our comparison procedure in Section 4.2. We report our results in Section 4.3. Then we discuss our findings in Section 4.4. We present our related works in Section 4.5 and summarize this chapter in Section 4.6.

4.2 Comparison Procedure

In this section, we present the theoretical base of our comparison in Subsection 4.2.1 and our comparison strategy in Subsection 4.2.2.

4.2.1 Documents

This comparison examines the roles of requirements engineers, building architects, and agile product owners. We based our analysis on the following documents:

Requirements Engineers

We selected the study guide *Requirements Engineering Fundamentals* [166] for the *CPRE Foundation Level* of the International Requirements Engineering Board (IREB). We chose these documents since they serve as a knowledge base for becoming a certified requirements engineer.

Building Architects

Our primary references for building architects were the *Neufert - Architects Data* [150] and the *HOAI: Fee Regulations for Services by Architects and Engineers* [187]. *Neufert* is a standard reference for planning and designing buildings. *HOAI* sets out general definitions and provisions for engineering and architectural services. These regulations specify responsibilities, tasks, and processes, including the task of gathering requirements. In addition, we included statements from the Federal Chamber of German Architects [60].

Agile Product Owners

Due to the popularity of *Scrum* in industry and academia, we have chosen its definition of a product owner for our comparison. To this end, we based our analysis on *The Scrum Guide* [181]. The guide defines "Scrum's roles, events, artifacts, and the rules that bind them together" [181].

4.2.2 Analysis

Based on our analysis, we identified four main themes: *liability, self-portrayal, core activities*, and *artifacts*. We depict in Figure 5 an extracted representation of the various *liabilities* within a project and the associated duties and responsibilities of the respective roles. Regarding the theme *self-portrayal*, we summarize the *values* and *characteristics* for each role. IREB defines categories for the main activities of requirements engineers, which we have used to match the *core activities* of the other roles. Finally, we provide an overview of the artifacts utilized by each role to communicate and document their requirements. As a result of this comparison, we identified initial differences and similarities. Based on these results, we derived hypotheses on how requirements engineering could evolve to enable sustainable software design.

4.3 Results

We present our results according to the topics *liability* in Subsection 4.3.1 and Figure 5, *self-portrayal* in Subsection 4.3.2 and Table 11, *core activities* in Subsection 4.3.3 and Table 12, and *artifacts* in Subsection 4.3.4 and Table 13.

4.3.1 Liability

Similarities. All three roles operate as an interface between the involved project participants. They communicate the needs of their respective client or customer to other contractors or development teams. Accordingly, one of their core duties is to determine the requirements for the product to be developed and to communicate them among the project partners.

Differences. Agile product owners and requirements engineers are usually not directly hired by their clients. They often work at the same company as the development teams. However, the building architect acts as the client's trustee towards the other contractors, who, e.g., construct the building.

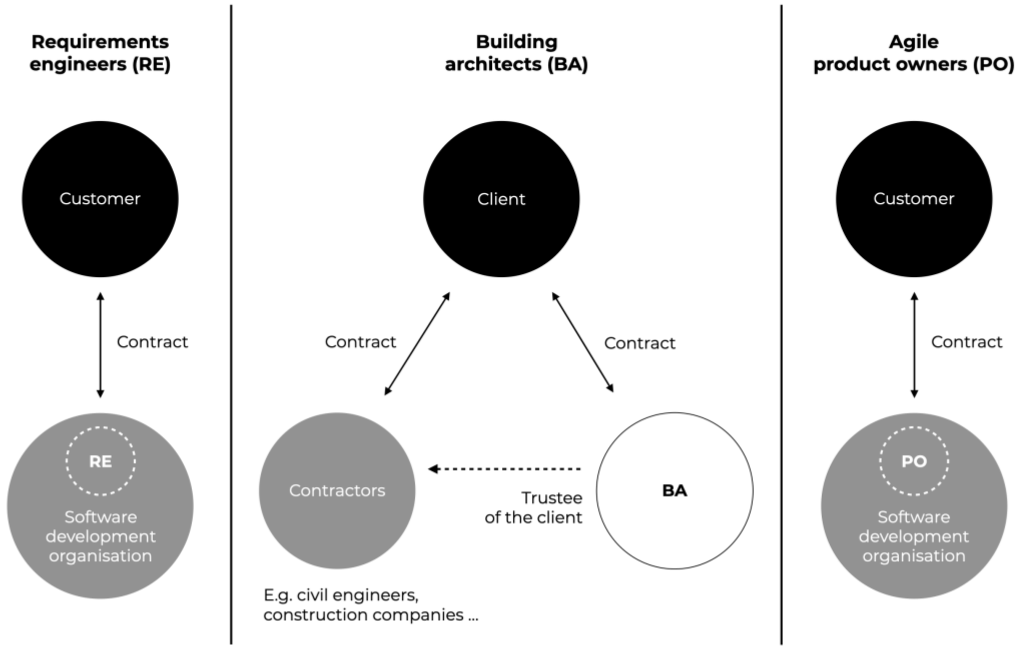


Figure 5: Liabilities of BA adopted from [150], RE based on [166], PO based on [181]

4.3.2 Self-Portrayal

Similarities. We found similar character traits between the roles, although the descriptions in the references differed slightly in their wording. For example, the IREB guidelines expect requirements engineers to be empathetic, similar to product owners, who should be open-minded, committed to the project, and respectful. These qualities are also supposed to be shared by trustees, such as building architects.

Differences. Building architects should take a holistic view of their projects. This means that they should be able to grasp the project in its entirety and keep an eye on the individual details, which are interconnected and cannot exist independently [155]. In this sense, building architects try to balance different interests while enabling innovative solutions. The balancing part sounds somewhat similar to requirements engineers who should be "persuasive to act as an attorney for the requirements of the stakeholders" [166]. On the other hand, product owners should focus "on the work of the Sprint and the goals of the Scrum Team" like everyone else [181]. This focus might create a culture of scripted communication that may neglect important contributions from external sources.

4.3.3 Core Activities

Similarities. All three roles share the understanding that it is crucial to involve stakeholders. They put this understanding into action by creating artifacts that they use to document requirements changes and distribute the latest information among all stakeholders.

Differences. Requirements engineers and building architects have to ensure that the requirements are traceable. Furthermore, building architects have to design, plan and supervise the implementation of the requirements and monitor the costs. Product owners have to perform two contradictory tasks. On the one hand, product owners should trust the development teams to *self-organize* and not interfere with how the teams turn the backlog items into increments [181]. On the other hand, the whole organization should respect the product owner's decisions, which makes them non-negotiable authorities [181].

4.3.4 Artifacts

Similarities. There is agreement among all three roles that requirement artifacts need to evolve with the project. They share the view that artifacts must be continuously taken care of, and requirements have different stages. Furthermore, they agree that requirements demand different levels of detail depending on the project's progress. For example, requirements at the beginning are less specific since the initial focus is on the general scope of the project, while requirements in the implementation phase should already provide very concrete information.

Differences. Product owners operate mainly with natural language to present the requirements of their backlogs. In comparison, requirements engineers also use conceptual models to communicate the requirements among the stakeholders. In addition to natural language and conceptual models, building architects use prototypes to demonstrate the project's requirements graphically. According to IREB, requirements engineers use three different types of requirements: *functional, quality (non-functional),* and *constraints*. In contrast, building architects have eight dimensions: *urban, creative, functional, technical, economic, ecological, social, public [legal]*. *The Scrum Guide* does not specify whether and how requirements should be differentiated.

Table 11: Self-portrayal

ASPECT	REQUIREMENTS ENGINEERS [166]	BUILDING ARCHITECTS [187], [150]	AGILE PRODUCT OWNER [181]
Values and characte-ristics	• Analytical thinking • Empathy • Communication skills • Conflict resolution skills • Moderation skills • Self-confidence • Persuasiveness	• Trustee of the client • Holistic • Persuasive regarding innovative design and construction	• Commitment • Courage • Focus • Openness • Respect

Table 12: Core Activities

ASPECT	REQUIREMENTS ENGINEERS [166]	BUILDING ARCHITECTS [187], [150]	AGILE PRODUCT OWNER [181]
Elicitation	• Obtaining requirements from stakeholders and other sources • Refining requirements	• Clarifying tasks and dependencies • Explaining tasks and dependencies	• Refining the items in the product backlog • Collaborating with the development team for refinement
Documen-tation	• Describing elicited requirements adequately	• Creating artifacts based on type, size, and progress of the project to the required extent and level of detail • Considering all disciplines and individual specific requirements	• Clearly expressing product backlog items
Validation and negotia-tion	• Involving correct stakeholders • Separating identification and correction of errors • Repeating validation • Validating from different views • Creating development artifacts • Changing documentation type adequately	• Coordinating involved parties • Inspecting artifacts and the contribution of all involved parties • Consulting the client • Negotiating with involved parties • Planning the implementation	• Inviting relevant project participants to sprint review • Ensuring the development team understands product backlog items • Explaining which product backlog items are *done* and which are *not done* yet in sprint review
Manage-ment	• Assigning attributes • Defining views on requirements • Prioritizing requirements • Tracing requirements • Versioning requirements • Managing requirements changes • Measuring requirements	• Adapting artifacts based on progress and contributions • Provisioning current progress of the project to all involved parties • Updating artifacts based on progress and contributions • Supervising construction • Controlling costs	• Optimizing value of the development team's work • Ordering items in the product backlog according to goals and missions • Ensuring that product backlog is visible, transparent, clear to all, and shows next task • Updating the product backlog at any time • Helping the development team (who make the final estimate) to scope and select tradeoffs

Table 13: Artifacts

ASPECT	REQUIREMENTS ENGINEERS [166]	BUILDING ARCHITECTS [187], [150]	AGILE PRODUCT OWNER [181]
Types	• One requirement document	• Drawings • Cost controlling documents • Tendering documents • Time schedule • Construction diary	• Product backlog
Languages	• Natural language • Conceptual models	• Natural language • Conceptual models • 2D/ 3D Prototypes	• Natural language
View on the artifacts	• "Requirements change over the course of the entire development and life cycle of a system" [166].	• The required extent and level of detail depends on type, size and progress of the project under consideration of all disciplines specific requirements [187].	• "The product backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful" [181].
Requirement types	• Functional • Quality (non-functional) • Constraints	• Urban • Creative • Functional • Technical • Economic • Ecological • Social • Public [legal]	• Requirements

4.4 Discussion

Comparing the three roles of requirements engineer, building architect and product owner in theoretical terms enabled us to identify similarities and differences between them. We discuss our findings according to the topics *liability* in Subsection 4.4.1, *self-portrayal* in Subsection 4.3.2, *core activities* in Subsection 4.4.3, and *artifacts* in Subsection 4.4.4.

4.4.1 Liability

Requirements engineers often work in the same company as the development teams. Therefore, requirements engineers might be biased and act in favor of the company rather than doing what is best for the customer. However, by perceiving themselves as trustees of their contracting authorities, they might feel more accountable for the entire design and implementation process. Hence, they might provide better advice and services that fulfill the needs of their customers and users. By acting as the client's trustee and their companies' trustee, requirements engineers might gain more understanding of the different parties and thus propose compromises that cover as many different needs as possible, including sustainability-related aspects.

Hypothesis 1. Software products are more user-friendly and sustainable if requirements engineers feel responsible for satisfying customers and users.

Hypothesis 2. Software projects in which requirements engineers feel mutually responsible for the entire development cycle progress more effectively.

4.4.2 Self-Portrayal

People with different professional backgrounds may have to work together on large projects, e.g., business administrators, developers, and designers. These project participants might have conflicting but equally important priorities. Unfortunately, not all of them will be concerned about the effect of their action on the overall quality of a product or on other groups. This behavior does not necessarily have to be due to people's self-interest. It seems likely that tight schedules, increasing workloads, and keeping the process and money flow going might prevent people from paying sufficient attention and understanding their project partners. While being under pressure to perform well, each team may also be measured against statistics and objectives that are not necessarily aligned with fulfilling the customer's needs. By adopting a holistic view, requirements engineers could discover interdependencies easier and settle conflicting issues sooner. This may lead to fewer implementation problems, improved quality, and a higher probability that customers and users will be more satisfied. Furthermore, requirements engineers should continue to exercise their analytical skills, but they should also be encouraged to pursue their creative abilities to come up with novel ideas. After all, new approaches may be needed if systems must be maintained or redeveloped, especially if the customers do not know what they want to make their system future-proof.

Hypothesis 3. The stakeholder's acceptance increases if requirements engineers adopt a holistic approach.

Hypothesis 4. Vague ideas and requirements are turned into concrete activities sooner if requirements engineers are encouraged to pursue their creative drive.

4.4.3 Core Activities

With its self-organizing approach, the *Scrum* framework enables project participants to operate more independently and achieve results faster. However, self-organization should not lead to hierarchies that, e.g., reduce the importance of the product owners. With diminished authority, product owners not only lose the opportunity to make significant contributions but also lose customers' trust. This can result in customers hiring their own product owners to ensure that their interests are sufficiently addressed [56]. In order to ensure that requirements engineers are given an equal position to the development teams, an organization should entrust them with responsibilities, such as consulting tasks that enable them to contribute to the project following the customers' interests.

Hypothesis 5. If development teams acknowledge the contribution of requirements engineers as important as their own, customer satisfaction will increase, and the risk of developing unfit-for-use products will reduce.

4.4.4 Artifacts

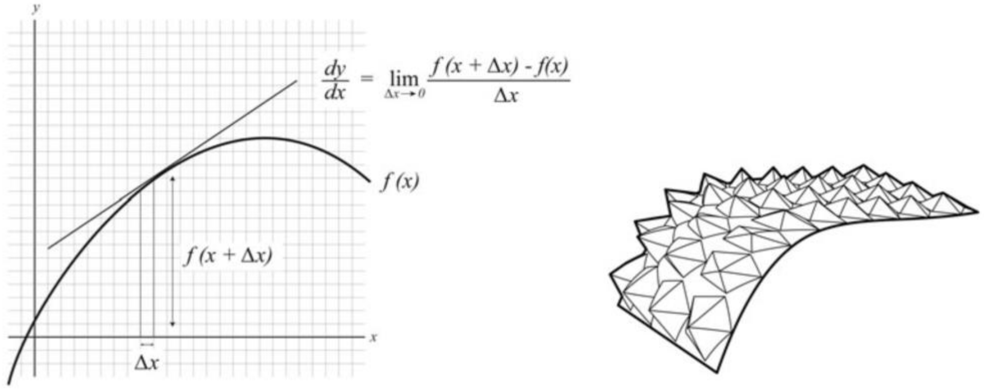
Requirements engineers have a wide range of options for expressing requirements using natural language and conceptual models. However, such abstract languages can reach limits regarding specific requirements, such as usability. Such requirements may be better communicated and evaluated in visual prototypes. In addition, prototypes contribute to improved comprehension, as international teams may have communication difficulties due to language barriers. Prototypes could also be more comprehensible to users and customers, who may not be as familiar with technical terms. Even though IREB has moved away from the binary representation of requirements, we assume that the three requirement types, *functional, quality*, and *constraint* requirements are insufficient. We suggest that with a more varied approach, like that of building architects requirement engineers might identify more requirements early on, which might be crucial for a holistic and sustainable product.

Hypothesis 6. If requirements engineers use prototypes to communicate requirements, it will save unnecessary implementation time.

Hypothesis 7. If requirements engineers apply a more diverse set of requirements dimensions, they will identify more requirements that could contribute to a holistic and sustainable product.

4.4.5 Limitation

We have conducted our analysis on a theoretical basis. To examine the role of building architects, we utilized documents from public institutions or well-known standard works. We focused on the German-speaking area since we could not identify any internationally recognized building architecture documents. Contrary to this, the documents we used for software engineering roles appear more international, but private institutions have provided them. We had to translate building architecture approaches into software engineering terms to transfer them. Therefore, it is possible that, due to the "translation," we used terms and concepts that German architects would not be able to reproduce in the same way. We assume building architects may need experience in software engineering to make similar observations as ours. Accordingly, we have to consider that different associations may exist behind the same term, as seen in Figure 6. The figure shows the possibility of different visualizations of differences [152]. Thus, it may be the case that, e.g., the atomistic representation of the requirements dimensions of German building architecture may not be reproducible by German building architects. Nevertheless, the abstraction enabled us to identify indications of possible synergistic effects.



2.4 A visualization of *differentiation* as understood by engineers.

Drawing by Christina Hoover.

2.5 A visualization of *differentiation* as understood by architects.

Drawing by Clare Olsen.

Figure 6: Visualizations of differentiations adopted from Olsen and Mac Namara [152]

4.5 Related Works

Software design and building architecture are analogous, according to Kapor [103]. To establish the role of a software designer, the author refers to the role of building architects. The author proposes that lecturers train software designers like building architects, not computer scientists. The new role should ensure that software products are user-friendly and well designed. Software programmers are too occupied with providing flawless and efficient codes. In a similar manner, Lauenroth [116] proposes a new role called *Digital Designer* that the author also based on building architects. Winograd and Tabor [215] agree with deriving beneficial insights from building architecture. However, the authors emphasize that we should not transfer every building architectural practice to software development due to essential differences. For example, many small and private software projects do not need an "over-arching-master-designer" like a building architect. Lecturers also need to be careful about training software designers like building architects. For a discipline as young as software engineering, a history-based teaching module is not necessary yet, as for a discipline as old as building architecture. Even so, they report that examining building architecture is worthwhile to identify "provocative questions" that may advance software engineering.

Berry [15] argues that building software products are similar to building houses. The author has written two reports about his experience as a building client with requirements engineering expertise. In the first report [15], the author identified similarities. For example, building architects and clients share a similar relationship with requirements engineers and their customers. As an example of the differences, the author noted that building contractors have stricter expectations of their clients than software companies. For instance, if the client changed the requirements, building contractors would increase costs and delay the completion date. In contrast, software companies would accept their clients' change requests and not change anything in the budget or time frame. In the second report [16], the author describes that informing clients about their tasks in advance might be beneficial. The author's expertise in requirements engineering prevented them from performing massive rework after the completion of the construction. In this regard, the author suggests that customers pursue requirements engineering activities to specify their objectives. Another observation Berry [17] made relates to liability and product quality. The author reports that the product quality depends on whether the manufacturers are responsible for them. In the author's description, software companies are not equally responsible for their products as manufacturers of analog products. This lack of liability would cause poor software quality, which we can partly share with our comparison results. However, it also works the other way around. Mauger and Berry [135] state that building architects rarely apply requirements engineering activities. Their work illustrates how a building architect would have designed a more user-friendly synagogue kitchen using use case scenarios.

The *International Workshop on Learning from Other Disciplines for Requirements Engineering (D4RE)* [92] exemplifies that researchers view integrating different disciplines with requirements engineering as highly beneficial. Using the insights from other disciplines, Heß et al. [92] aim to advance requirements engineering.

4.6 Summary

In this chapter, we compared the roles of requirements engineers, product owners, and building architects. We based our analysis on documents that are characteristic of these roles. We identified similarities as well as differences. For example, we identified the shared view of involving the clients and users as soon as possible. We identified a difference regarding the requirements dimensions, which is the base of Hypothesis 7. Based on this hypothesis, we prepared our studies in Chapter 6 and Chapter 7. We reported on related works, which acknowledged the similarities between building a house and programming a software product. Previous scholars indicate that integrating insights from the building industry could be beneficial for requirements engineering.

5 Perception and Needs of Practitioners Regarding Sustainable Software Design

Publication. This chapter is partly based on our paper *The Role of Linguistic Relativity on the Identification of Sustainability Requirements: An Empirical Study* [162]. It aims to investigate practitioners' perspectives on sustainable software design. My contributions to this study comprise the research design, conducting the interviews, the data preparation, and analysis, discussion of our findings, as well as leading the writing of the paper.

Contribution. Our interviewees' responses highlighted the complex reality that must be taken into account when designing sustainable software. According to our study, stakeholders need to address several challenges and concerns to apply sustainability-related requirements dimensions. Nevertheless, the participants recognized the benefits of incorporating sustainability-related dimensions into the daily work environment and provided suggestions on how to do so.

5.1 Motivation

According to the Karlskrona manifesto, requirements that take sustainability into account play a crucial role in sustainable software design. However, other factors, such as the support of a wide range of stakeholders, are just as essential [4]. Therefore, we conducted semi-guided interviews with sixteen practitioners. The interviews aimed to gain a first impression from practitioners regarding sustainable software design. We were particularly interested in learning what they thought about sustainability-related requirements dimensions. From the interviewee's responses, we aimed to draw insights we should consider for our requirements engineering extension to promote sustainable software design.

Based on the findings of the interviews, we defined sustainable software. Using our proposed dimensions set Dim_{SUST} as an example presented in Section 6.2, we identified potential benefits and concerns associated with sets incorporating sustainability-related dimensions. The set Dim_{SUST} consists of the nine dimensions: *ecological, social, individual, economic, technical, integrative, aesthetic, legal,* and *purpose*. The first five dimensions represent the sustainability-related dimensions. In addition, we identified challenges that may prevent sustainability-related dimensions from being applied in practice and gathered recommendations for how to facilitate their application. We organized the remainder of this chapter as follows. In Section

5.2, we describe the research design. Then, we present our results in Section 5.3 and discuss our findings in Section 5.4. Finally, we report on related works in Section 5.5 and summarize the chapter in Section 5.6.

5.2 Research Design

In this section, we present our research questions in Subsection 5.2.1 and our sample in Subsection 5.2.2. We outline our interview procedure in Subsection 5.2.3 and our analysis approach in Subsection 5.2.4.

5.2.1 Research Questions

We aimed to gain a deeper understanding of practitioners' perceptions regarding sustainable software design and sustainability-related requirements dimensions. To this end, we were interested in answering the following research questions:

RQ1. What do the participants associate with sustainable software?

RQ2. What are participants' concerns about requirements dimensions that include sustainability-related aspects?

RQ3. What challenges do participants anticipate when using requirements dimensions that include sustainability-related aspects?

RQ4. What benefits do participants expect from requirements dimensions that include sustainability-related aspects?

RQ5. What do participants recommend to promote sustainable software design?

5.2.2 Sample

A total of sixteen practitioners participated in our study. We listed our interviewees in Table 14 and provided additional information about them. We were looking for company managers, product owners, developers, and user-experience designers involved in software development. Since we focused on industry practice, we excluded students, university researchers, and practitioners with less than two years of work experience. Furthermore, the interviewees worked or have worked in projects that applied an agile development process or at least adapted an approach of an agile development process. As a means of finding participants, we used snowball sampling and convenience sampling [107]. Our first step was to send out an email to our immediate circle of acquaintances. We provided the selection criteria, a short description of our study, and stated that participation is voluntary. Following Marshall's recommendation [132], we continued to conduct interviews until we had enough insights and there were no more new findings.

Table 14: Overview interviewees

ID	OCCUPATION	EXP. YEARS	DOMAIN	COMPANY SIZE	ROLE	COMPANY TYPE
M1	Manager	14	IT-Consulting	8.000	Ext. contractor	Private
CU1	Product owner	7	Culture and language	3.800	Customer	Non-profit
CU2	UX designer	5	Culture and language	3.800	Customer	Non-profit
PO1	Product owner	13	Pharmacy	51.000	Int. contractor	Private
PO2	Product owner	6	Media	9.000	Ext. contractor	Private
PO3	Product owner	20	Enterprise software	100	Ext. contractor	Private
D1	Developer	19	Enterprise software	100.000	Ext. contractor	Private
D2	Developer	10	Advertisement	700	Ext. contractor	Private
D3	Developer	3	Enterprise software	100	Ext. contractor	Private
D4	Developer	4	Enterprise software	100	Ext. contractor	Private
D5	Developer	10	Green electricity	500	Int. contractor	Private
D6	Developer	15	Media	2	Ext. contractor	Private
UX1	UX designer	3	Pharmacy	51.000	Int. contractor	Private
UX2	UX designer	6	Green electricity	500	Int. contractor	Private
UX3	UX designer	10	Freelance	1	Ext. contractor	Private
UX4	UX designer	2	Freelance	1	Ext. contractor	Private

5.2.3 Interview Procedure

We conducted semi-structured interviews with our participants to gather detailed information about their views on the dimensions. As two interviewers, one guided the conversation while the other took notes and asked follow-up questions as necessary. Each interview took about 45 to 60 minutes. In order to analyze the answers, we audio-recorded the conversation via Zoom with the participants' permission and fully transcribed the interviews. Following Creswell and Creswell [45], we designed an interview guideline. We conducted the interviews in German and included the English translation of the interview guideline in Appendix A. We organized the interview into the following four parts:

1. **Introduction and Working Experiences.** First, we introduced ourselves and summarized the general conditions, such as duration. We again asked for permission to record. Then we started with general questions about their current occupation and work experiences.

2. **Experience with Requirements Engineering and Software Sustainability.** In the second part, we were interested in the participant's experience with requirements engineering and software sustainability.
3. **Sets of Requirements Dimensions.** The third part aimed to obtain as much genuine feedback as possible regarding Dim_{SUST}. To this end, we showed the participants three sets of requirements dimensions, as presented in Section 6.2. Besides our own set, we also presented the dimensions *functional* and *non-functional* requirements and those of *ISO Standard 25010:11 - System and Software Quality Models* [98]. We named the sets A ($Dim_{FR\text{-}NFR}$), B (Dim_{ISO}), and C (Dim_{SUST}) in the interviews. Additionally, we offered option D, allowing participants to say they do not need dimensions. We presented our set Dim_{SUST} with other sets of dimensions to reduce the so-called *demand effect* [153]. This effect represents the participants' attempt to guess the researcher's goals and adjust their behavior accordingly [153]. We asked clarifying and open-ended questions to guide the interview, e.g., "What are your first impressions of the dimensions?" and "Which one would you use to design sustainable software?"
4. **Demographics and Future Work.** During our last part, we asked the participants for a few demographic details, concluded with the remaining questions from the participants, and thanked them for their participation.

5.2.4 Analysis

We recorded each interview and transcribed it. We divided the transcripts into two random sets. As two coders, we performed a thematic analysis following Clarke and Braun's [38] approach, which included six recursive steps: "1. Familiarizing yourself with the data and identifying items of potential interest, 2. Generating initial codes, 3. Searching for themes, 4. Reviewing potential themes, 5. Defining and naming themes, 6. Producing the report."

1. **Familiarizing.** First, we worked independently on our respective sets. Throughout the process, we read and reread each transcript. According to the research question, we identified potential insights in the data.
2. **Generating Initial Codes.** We generated initial codes to capture the most exciting aspects of the data. We decided that each of us would compile a list of codes.
3. **Searching for Themes.** Based on our list, we searched for initial themes. After that, we met and discussed our list of codes and themes.
4. **Reviewing Potential Themes.** As part of our evaluation process, we examined our identified themes to determine whether they captured the most critical aspects of the data and whether they were relevant to the research question.
5. **Defining and Merging Themes.** During this step, we extracted the most appropriate themes, merged themes, and formulated new themes if necessary. Using these themes as a guide, we reread and analyzed our transcripts once again.

6. **Producing the Report.** By writing the report in Section 5.3, we refined the analysis and added quantitative figures. Our objective was to present our analysis comprehensively and insightfully.

5.3 Results

In the following section, we present the interviewees' responses according to our research questions. We start with their general associations with sustainable software in Subsection 5.3.1. We continue with their concerns in Subsection 5.3.2 and benefits in Subsection 5.3.3 that they assume by applying sustainability-related dimensions. We finish with their challenges in Subsection 5.3.4 and their recommendations to promote the incorporating of sustainability-related dimensions into daily working activities in Subsection 5.3.5.

5.3.1 Attributes of Sustainable Software

Our interviewees used the following three characteristics to describe sustainable software:

Long-Lasting. According to D6, this refers to maximizing a product's durability: "So for me, sustainability is about software development that does not become a throw-away product but creates something that lasts as long as possible and can ideally [...] be used in different projects." PO2 and UX1 also think usability is essential in extending the lifespan of a software product. Furthermore, prior research is crucial in designing future-proof software products that can adapt to changing user requirements, as UX1 reported.

Resource-Conserving. UX3 emphasizes that the production and maintenance of sustainable software should consider not only material and financial resources but also human resources. A modular programming approach and open-sourcing may be effective in achieving this goal, as suggested by D2: "Build standard components and share them, maybe even make them freely available as open source and give them to others." This statement aligns with PO3's opinion that sustainable software design should support reusing components to avoid redeveloping. The use of green energy to run servers and other devices was another resource-conserving aspect that the interviewees mentioned.

Promoting Sustainable Behavior. UX2 states that sustainable software can affect the environment and society. Additionally, UX2 suggests using interaction design to promote sustainable behavior through software. "[e.g.], I buy a new car, and it has an eco-mode. But whenever I start driving, it is in *normal* mode. Why is *eco* not the default?"

5.3.2 Concerns

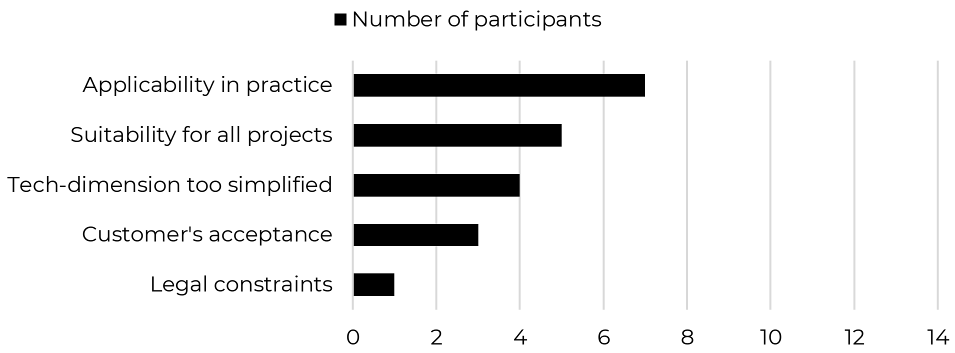


Figure 7: Interview results – concerns

The biggest concern of the interviewees was that dimensions sets like Dim_{SUST} might be **difficult to apply in practice**. D6 revolved his concerns around how to formulate sustainability-related requirements and implement them: “I was just thinking, ecological, social, or integrative, those are things I would have to keep thinking about to formulate specific points that result from them. I have not figured out what I would do with such requirements yet.” In addition, PO1 and PO2 were unsure how to measure the dimensions. They could not imagine yet how to assess whether a project sufficiently fulfills these dimensions.

Five participants evaluated the set Dim_{SUST} as **unsuitable for every kind of project**. PO1 and D4 reported that it would take “too much effort for most development projects,” and UX2 assumed the set was not fitting for small projects.

Furthermore, four interviewees stated that the **technical dimension is too simplified.** The interviewees perceived this dimension as too vague and that it needs more specific information to be applicable. Due to the familiarity with dimensions such as *security* or *maintainability,* PO2 would probably feel more comfortable if these dimensions were included in the set Dim_{SUST}. UX3 noted, “In set C, I am missing the topic of *compatibility* and *usability*. And *efficiency*, as well.” PO1 and PO2 preferred working on dimensions such as *maintainability* and *security*.

Other interviewees reported that the **customer’s acceptance might be difficult**. PO3 expressed their worries that customers would be more reluctant to accept such a set. Similarly, D4 voiced the concern that the benefits for the customer might not be obvious enough.

PO3 added that there may be **legal restrictions** that could affect the other dimensions. PO3 reported that the integration of open-source code could save various resources, but the project domain may prohibit the use of open-source code.

5.3.3 Benefits

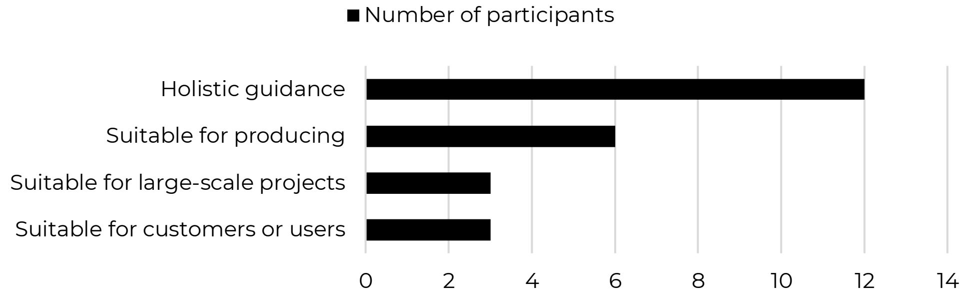


Figure 8: Interview results – benefits

Twelve interviewees stated that the set Dim_{SUST} **offers divers and holistic guidance** in a project that transcends technical considerations. This was described by D2: "as a developer, I am, of course, often more from the technical direction. But I have also noticed that the product is not good because it is technically perfect, but that there is a multitude of factors that flow into it, and only if they are all evenly distributed [...] [can] the product be good in the end." On a similar note, CU2 mentioned: "It simply includes the environment, [it] is not only focused on the product but also includes the entire ecosystem and thinks further than just: we are now developing a product, that must be efficient." Furthermore, the interviewees recognized that sustainability often gets discarded when using classical requirements sets, which they could prevent by integrating sustainability dimensions as explained by CU1: "[...] because, for example, ecological, individual, or social requirements, I think, are often requirements that ultimately fall by the wayside when you think in terms of the more classic requirement categories or work with these. I think that basically, requirements such as maintainability are safe, there are always enough people who think about it, [...], and ecological requirements tend to fall off the table or are simply not as present." Moreover, dimension sets like Dim_{SUST} also lead to perspectives they would not have thought about. Before we presented Dim_{SUST} to M1, M1 mainly thought of ecological aspects when talking about sustainability and then said when M1 saw the set Dim_{SUST} "I could have thought of that (social dimension), too, [...] that was not on my note at all."

The interviewee PO2 further pointed out that the model is **suitable for producing sustainable software** especially "at the beginning of a project, so when somehow the cornerstone and the foundation is built, that this (sustainability) is once also considered." D4 also confirmed that "the set C looks quite suitable for producing sustainable software."

According to UX2, the set is especially beneficial for **large-scale projects** stating: "If I would be a software service provider and had to build the Corona app together with the federal government, then, of course, you could say that we would take this set because it is simply a huge thing and [...] I think you have more opportunities to think bigger and use a requirements set like that because you have more resources available." D4 shared this view and told us: "So in the direction of Windows, Microsoft Windows that if you really somehow have something that is widespread, that should be used by everyone, that is really intended for society, for the whole of society in general."

The interviewees D5 and D6 found the set Dim_{SUST} not too technical and therefore **suitable for working with customers or users**. Since they perceived Dim_{SUST} as less technical, they assumed Dim_{SUST} might be easier to comprehend by customers and users. However, D5 added: "I think it depends on the relationship with the customer and whether the customer wants something or whether you explain to them what would be good. [...] It is more a matter of the relationship. Apart from that, I find set C to be well applicable."

5.3.4 Challenges

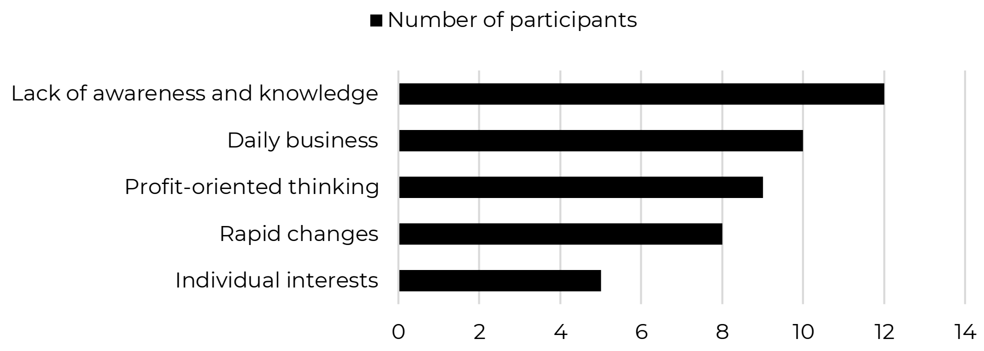


Figure 9: Interview results – challenges

The most reported challenge was people's **lack of awareness and knowledge** on software sustainability. D6 stated, "so my subjective feeling is that I say it (issue of software sustainability) is probably at least officially very low at the moment, but I think that [it] is like these non-functional requirements, that everyone would like to have that, but has never named it so concretely." UX4 expressed concern: "I have worked in marketing agencies [...], and sustainability plays absolutely no role there and that it tends to be laughed at." D2 confirmed, "And it is not the case that this is an issue for us. We also have development rounds or together with our superiors, where we regularly exchange ideas and pick each other up on new topics. This is not necessarily an explicit topic that is dealt with there, and it could certainly take place more." Furthermore, they are missing official definitions as D5 has reported: "this term (sustainable software) is not really defined. As I said, it does not mean that it has not already been used in some way or that it has not been promoted; perhaps it is just not known by that name, so to speak." This lack of knowledge and awareness could hinder the application of dimensions that go beyond the ecological dimension. M1 reported: "What I find difficult about the term sustainability is that I think it has already been coined, that sustainability already has a green imprint. Sustainability means recycling. Sustainability means carbon footprint. Sustainability means planting trees. I think that is already very set as a term. That is why it was very difficult for me to break out of the green framework when I thought of sustainable software."

The next challenge is to deal with their **daily business**. The interviewees report that they have to deal with time pressure in their daily work. They are concerned that if they address issues such as sustainability on their initiative, they may not receive any commissions. D1 told us: "I naturally risk not getting the job. So that is a bit of a balancing act," and D5 stated that they have the feeling that they have no time "to take a deep breath and think things through again in peace, but rather, especially as a service provider, every project is actually stressful. Some projects are even behind schedule before it has started. That is why you fall back into what you know. You then take that first and do not want to question everything." D3 perceive their power as so little that they are more occupied with getting their jobs done. Which is why they cannot imagine incorporating issues like sustainability: "Since we are at the bottom of the ladder [...] we have to keep up with what they come up with at the top. And then we are much more concerned with making sure that the software runs at all." In some cases, the contractors are also busy creating the groundwork for their projects which they would have expected from their customers, as UX2 has reported: "[...], requirements management has been one of the core problems in every job I have had."

Another big challenge was that the interviewees perceived their environment as **profit-oriented thinking**. D1 stated: "So that is the reality, so if a customer just wants to somehow maximize a profit or something, then the first question is not about any social aspects" and added, "The market determines to a large extent what you have to offer as a service provider." M1 stated "I do not know what kind of customer would actually appreciate such a system, such a model." M1 continues: "This is the question that customers ask. And what does it mean to me 'Does this give me more money? Does that give me any market advantage?' and if we say, 'No, it does not do either, but you will have a better conscience for it because you are doing something that does not hurt the world,' most customers will tell us 'That is very sweet, but we have a real business here, and we want to have made more money at the end than we had at the beginning of the year.'" D9 confirms this observation "So I think when they look at it (set Dim_{SUST}), they first see okay, nice, ecological, and so on. But where is money? Efficiency?"

In addition to their daily work stress, the interviewees also have to keep up with **rapid changing towards technology and business**. UX2 raises that sustainability aspects could only be applied if they do not stop technological progress. D5 experienced that an application "would have continued to work if it had stayed the same. But [...] the industry has also continued to develop." which is why the application was not able to fulfill the need of the client anymore. PO3 has observed that applications had to be redeveloped entirely because of new requirements and trends for the front end, even though the previous applications could still perform the desired tasks. On the other hand, PO3 reports that adjustments and modernizations of the backend are sometimes difficult to enforce because they are often costly, and in the end, they are not visible to the end user.

Furthermore, the interviewees have to deal with **individual interests** before even considering sustainability-related aspects. PO3 reported that "managers always want to ensure that they get something good for themselves." This can be very troublesome if they cannot see their own advantage within this set. Also, individual ambitions can hinder sustainable thinking, as UX2 has stated: "I go to a designer and say, 'Do not put in videos because that uses too much energy, then they would also shake their head and say 'Yeah, but I just want to make an awesome design.'" Perhaps developers may not take such dimensions seriously, as D4 commented: "That is a bit far away from the development process, I would say. For me as a developer, it does not matter whether a feature I am adding is economic or whether it is integrative." M1 stated that projects have failed because of the lack of people's support: "the middle management says 'No, so we are not up for what you are doing there at all. We do not really care what the CTO's vision is right now." Furthermore, we must anticipate that we will not only need to inform people but also convince them to change over time, as reported by M1 and UX1. UX1 experienced that customers and users did not want to change. They did not want to know why they should exchange their current system that works for them for a new one, which they would have to learn all over again.

5.3.5 Recommendations

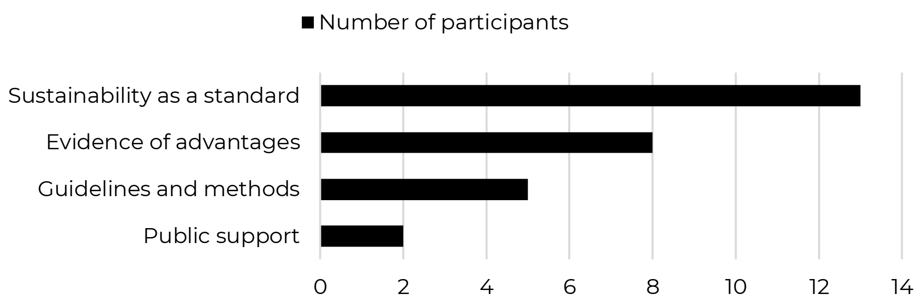


Figure 10: Interview results – recommendations

The interviewees reported that the market and the customers are decisive for being able to consider sustainability-related aspects and thus be able to work according to sets like *Dim*$_{SUST}$. To this end, the companies would have to establish **sustainability as a standard request**. D1 recommended integrating the concept of sustainable software into the company culture so it will not feel like an extra effort but rather "normal" to develop sustainable software. PO2 suggested that "it is important to actually include this as a fixed component in the area of requirements management and, above all, to anchor it in the goals of the companies. This could be through target agreement systems such as OKRs or something similar." CU1 reported discussing how to integrate sustainability issues into their tender documents to motivate contractors to suggest sustainable solutions. Since it is essential to them as clients to provide sustainable products, CU1 stated that they attract contractors who are also interested in sustainability. According to M1, we should consider the need of the industry: "The software [...] must be sustainable. [...] The business goals and values

[...] must be sustainable. All of this has to go hand in hand because sustainable software in isolation does not work, in my opinion." To this end, PO2 reported that they established the role of a *Chief Sustainability Officer* in their company. This new role is responsible for the sustainable orientation of their company. Alternatively, initiatives like *Sustainability Visionaries* push sustainability issues across departments by conducting workshops and events like in the company of CU1 and CU2.

Furthermore, they need **evidence of advantages** to see whether applying such dimensions sets will be profitable, e.g., gaining competitive advantages, improving their image, or saving money. Not just because the companies can attract more users but also employees like D1, who reported, "[...] and funnily enough, that is also one of the reasons why I no longer work there. Because it feels like you are always starting over again, I totally miss the sustainability. I do not feel like doing something and have the feeling that in two weeks, no one will care anymore. Not even the customer." UX1 reported that they had to conduct their own studies to convince customers about benefits, which is not always feasible in terms of time, and therefore new ideas may not be implemented.

Therefore, the interviewees have an interest in **guidelines and methods**. UX2 wished for a set of sustainable requirements. D6 said, "[...] to have such a scheme, on the basis of which you can think through the whole thing in order to ultimately [...] not to disregard certain things." PO1 and PO2 wished for methods or tools to measure sustainability, not necessarily as a specific money value but maybe a score. D5 wished sustainable software "becomes even more of a term that it is even more defined, what it actually means because it is still certainly a bit fuzzy." D5 also believed that Dim_{SUST} needs to be more concrete in terms of the technical dimensions to serve as a basis for a guideline.

Moreover, the last prerequisite they need to be able to implement sustainability-related requirements is **support from public institutions or the state**. D4 suggested that the state could motivate companies and customers to develop sustainable software with subsidies. It might also be helpful to have overarching standards, such as ISO standards, to follow, as a UX2 has suggested. UX2 continues that certification programs such as the *Blue Angel* [72] for software could help by being mandatory, e.g., for software applications with public interest.

5.4 Discussion

We discuss our implication in Subsection 5.4.1 and threats to validity in Subsection 5.4.2

5.4.1 Implications

Even though our practitioners recognize the importance of software sustainability, it seems to have little or no influence on everyday industry practices. This observation aligns with the findings of Karita et al. [104] that software development projects often perceive sustainability as a non-relevant topic. The responses indicate that the combination of factors, such as competitive pressure, profit-driven thinking, lack of awareness, and the interests of individuals might have caused this contradiction. Our interviewees recommended that we need to take various actions to establish sustainability as a standard in software development.

Our interview results indicate that sustainable software is still primarily associated with “green” or ecological concerns. During our interview with participant M1, this became particularly evident. According to M1, the concept of sustainability is already highly preoccupied, and other relevant aspects, such as social issues, are not included in its context. However, since the United Nations report *Our Common Future* [198], we must at least take into account *ecological, social,* and *economic* aspects to act in a sustainable manner.

In light of this, it would be advantageous to redefine sustainability to encompass a variety of implications. Putting all the aspects of sustainability into a succinct definition will require time, discussions, and the awareness that sustainability will always be a changing concept. A fundamental component of a new industry standard could be a set of requirements dimensions, such as Dim_{SUST}, as presented in Section 6.2. According to the linguistic relativity hypothesis, presented in Subsection 2.2.3, interviewees believed dimensions could assist in identifying requirements. Despite acknowledging Dim_{SUST}*’s* holistic approach, the interviewees pointed out its technical dimension was too vague. The interviewees suggested modifying Dim_{SUST} to include more technical-focused dimensions, such as those specified in the *ISO Standard 25010:2011 – Systems and Software Quality Models* [98]. This suggestion is similar to the proposal of Lago et al. [112] to frame sustainability as a software quality attribute. Sustainability-related dimensions may be more acceptable to stakeholders if we provide them with standard dimensions like those, we can find in the ISO standards. Stakeholders may take time to realize sustainability’s importance and benefit in software development. Thus, proposing Dim_{SUST} or similar sets of dimensions more approachable manner would be beneficial.

Another issue raised by interviewees was regarding long-lasting software applications. The interviewees considered maintaining the software for the longest possible period to make it sustainable. However, there may come a time when the operation and maintenance of old software systems consume too many resources to yield any lasting benefits. Sustainable software should never reach this stage. We should design it in a way that allows regular and efficient updates and modernization. Even if it is necessary to replace a software system, we should strive to reuse its components. We can save resources for future endeavors by designing software with exchangeable modules. The research community should provide sufficient and current insights about technology and design trends to support the industry in designing sustainable software.

5.4.2 Threats to Validity

The implications of our findings do not provide a comprehensive picture of industry practice and professional opinions. All interviewees were self-motivated, and we did not offer any compensation. This indicates an active interest in the topic of sustainability. As a result, our practitioners who agreed to participate in our interview are likely to have been more interested in sustainability than a representative sample might have been. We sent sample questions and an overview of our research goals to potential interviewees. To avoid misunderstandings, we provided our interviewees with the opportunity to ask clarifying questions before, during and after each interview. Nevertheless, we cannot exclude that there might have been misunderstandings that the interviewees might not have dared to ask. The interview questions remained the same during the whole duration of the study. While we conducted the interviews in German with German practitioners, we reported the results in English. The translation might include translation losses due to the individual nature of each language.

As our interviews are based on a relatively small, non-representative sample, the quantitative figures we report in our findings, the reader must treat those with care. From a statistical viewpoint, these quantities are not generalizable. While the quantitative data we report are factual for our sample, a study based on a representative sample might yield different results. Hence, with respect to generalizability, our quantitative findings should be considered as hypotheses rather than generally valid facts.

5.5 Related Works

In previous studies, researchers used various methods to examine practitioners' perspectives on software sustainability. The results of these research endeavors were similar to our findings, indicating that the industry is interested in approaches to promoting sustainable software design.

Chitchyan et al. [36] conducted an exploratory qualitative interview study with thirteen requirements practitioners. According to their results, the topic of sustainability has not yet gained popularity in the software industry. They report: "software practitioners tend to have a narrow understanding of the concept of sustainability; organizations show limited awareness of its potential opportunities and benefits, and the norms in the discipline are not conducive to sustainable outcomes."

Groher and Weinreich [80] confirm the observations of Chitchyan et al. [36]. Their exploratory interview study with ten team leads or lead developers revealed that practitioners associate sustainability primarily with technical aspects, such as "maintainability and extensibility." Furthermore, the authors miss environmental consideration.

Bambazek et al. [10] surveyed 45 employees from the IT sector or a department related to IT. Their study sought to identify how practitioners can use agile processes, specifically *Scrum*, to develop sustainable software systems. They found that the participants did not have a common understanding of sustainable systems. Nevertheless, their interviewees believe that incorporating sustainability into agile development processes has great potential. To this end, they suggest, e.g., establishing sustainability experts.

In their study, Manotas et al. [129] surveyed 464 practitioners and conducted 18 in-depth interviews. Their interviewees were practitioners who "appeared to have experience with green software engineering." The authors observed that green software engineering practitioners care and think about energy when they build applications. However, if practitioners had access to the necessary information and support infrastructure, they might achieve greater success. For example, "Requirements elicitation strategies would be more useful if they helped practitioners easily understand and express how much energy usage is reasonable for a given task."

Salam and Khan [174] reviewed 54 research papers and surveyed 108 practitioners. The authors aimed to identify the challenges and critical risk factors multisourcing vendors encounter when developing green and sustainable software. These critical risk factors are, e.g., "lack of green RE practices," "high power consumption," "poor software design (architectural, logical, physical, and user interface)," or "lack of green software development knowledge."

Calero et al. [32] conducted a study using only documents. The authors examined whether software companies align their strategies with environmental concerns in their study. To this end, they have reviewed the corporate social responsibility policies of ten software companies. They investigated these documents according to the UN's sustainable development goals. Their analysis results indicate that most companies take sustainability into account. However, the companies mainly focus on hardware-based approaches to reduce the carbon footprint of their hardware resources.

5.6 Summary

The study involved 16 semi-guided interviews with practitioners working as developers, UX/UI designers, product owners, managers, or on the customer side. As software development usually involves more than just software developers, we sought an interdisciplinary sample. We were interested in understanding what practitioners think about sustainable software development and how sustainable dimensions may contribute to this. We interviewed all practitioners in pairs and spoke with each individually. We described what sustainable software means to practitioners based on a thematic analysis. Additionally, we sought to discover what concerns, benefits, and challenges the interviewees associated with sustainable software design and the use of corresponding requirements dimensions. As a result of the interview, we also received recommendations as to what we should do in practice to ensure that we can use and promote sustainable software design. The interviewees expressed an interest in supporting the promotion of sustainable software design. In addition, they found it helpful to apply sustainability-related dimensions, such as Dim_{SUST}. Due to daily performance pressures and a lack of evidence regarding the benefits of sustainable software design, they believe it may be challenging to do so. Related studies have reported similar results. However, during the interviews, we observed that practitioners were very interested in sustainable software design. Nevertheless, our interviewees missed supporting approaches and evidence. We have addressed these challenges in the following chapters.

PART II: SOLUTION

6 Exploring Sustainability Dimensions for Requirements Engineering

Publication. This chapter is partly based on our paper *ShapeRE: Towards a Multi-Dimensional Representation for Requirements of Sustainable Software* [163] and extends Hillen's research [94]. It focuses on exploring the relationship of language with the identification of requirements. My contributions to this study comprise the research design, providing the objects of research, the literature study, the data preparation, and analysis, discussion of our findings, as well as leading the writing of the paper.

Contribution. The study served primarily as a pilot study. We gained initial insights into the relationship between requirements dimensions and requirements identification. We tested different sets of requirements dimensions and realized, e.g., that we need to strive for comparability when selecting the sets. Additionally, the study provided us with more insights on what improvements we need to make, such as redefining descriptions to increase comprehensibility.

6.1 Motivation

We discussed in Chapter 2 that sustainability is an omnipresent issue in our society. However, as we pointed out in Chapter 5, it has not yet played a significant role in the software industry. Therefore, sustainability is still not a given aspect of software design, as we observed in Chapter 3. We examined how we can apply our natural language to consider sustainability-related requirements. According to the linguistic relativity hypothesis, as described in Subsection 2.2.3, requirements dimensions could support identifying requirements in general and considering sustainability-related requirements. We define a sustainability-related requirement as a requirement that we can match to at least one of the sustainability dimensions of the Karlskrona manifesto [13]. In order to explore whether the linguistic relativity hypothesis is relevant to identifying requirements, we pursued two objectives with this exploratory study.

Our first objective was to gain a preliminary understanding of whether applying requirements dimensions or people's individual traits influence requirements identification. To this end, we conducted two online experiments. We conducted the experiments one after the other to implement the learnings from the first experiment in the second. As part of our experiment, we asked the participants to identify requirements for a grocery shopping app using a set of dimensions, as presented in Section 6.2. Second, we aimed to evaluate our study design as a preparation for our main study, as reported in Chapter 7. Our exploratory experiments allowed us to determine whether the research results answer our research questions. We evaluated whether the participants understood the assignments correctly based on our results.

The remainder of this chapter is structured as follows. In Section 6.2, we present the sets of requirements dimensions. We describe our research design in Section 6.3. We report the results of our Exploratory Experiment I in 6.4 and our Exploratory Experiment II in Section 6.5. We then discuss our findings and threats to validity in Section 6.7. Finally, we present related works in Section 6.7 and summarize this chapter in Section 6.8.

6.2 Sets of Requirements Dimensions

We present our examined sets of requirements dimensions $Dim_{FR\text{-}NFR}$ in Subsection 6.2.1, Dim_{ISO} in Subsection 6.2.2, and Dim_{SUST} in Subsection 6.2.3.

Exploratory Experiment I. In our first exploratory experiment, we included the set $Dim_{FR\text{-}NFR}$ and Dim_{SUST}. The set $Dim_{FR\text{-}NFR}$ consists of two dimensions: *functional* and *non-functional*. We chose $Dim_{FR\text{-}NFR}$ to investigate the most common approach to differentiate requirements [27],[53]. We chose Dim_{SUST} to examine the extent to which the linguistic relativity hypothesis holds true in identifying sustainability-related requirements by explicitly applying the sustainability dimensions of the Karlskrona manifesto [13]. The set Dim_{SUST} is an interdisciplinary set of nine dimensions, which we prepared to promote sustainable software design [162].

Exploratory Experiment II. However, we noticed after the first exploratory experiment that we might have skewed the comparison between $Dim_{FR\text{-}NFR}$ and Dim_{SUST} in favor of Dim_{SUST} due to the different number of dimensions. Therefore, we replaced $Dim_{FR\text{-}NFR}$ with Dim_{ISO} in Exploratory Experiment II. The set Dim_{ISO} includes the eight dimensions of the *ISO/IEC 25010:2011* standard for software quality [98]. We chose the set Dim_{ISO} because it has a similar number of dimensions to Dim_{SUST}. Additionally, we aimed to examine whether an established set of dimensions, such as the *ISO/IEC 25010:2011* standard, might already be sufficient to consider sustainability-related aspects. Since we follow the influencing and not the deterministic interpretation of the linguistic relativity hypothesis, we aimed to investigate whether it is necessary to apply sustainability dimensions explicitly to consider sustainability-related aspects. Since software quality also implies long-term usage, which stakeholders consider crucial for sustainable software [162], its dimensions might inspire participants to identify sustainability-related requirements.

6.2.1 $Dim_{FR\text{-}NFR}$: Functional and Non-Functional Requirements

The distinction of software requirements into *functional* requirements and *non-functional* requirements is one of the most widespread approaches [27],[53]. Nevertheless, the binary distinction has been evaluated as misleading and insufficient [74]. This evaluation is reflected especially by the observation that software developers treat non-functional requirements as leftovers [128], which they cannot assign to functional requirements. Even though researchers reported how necessary non-functional requirements are for software development, they are often ignored in practice [46]. This shortfall is critical since researchers consider sustainability a non-functional requirement [157]. As a result, this binary requirement classification approach might hinder stakeholders' ability to identify relevant requirements. However, requirements engineering is aware of the potential influence of language on requirements, as we can see from the numerous proposals on how to differentiate requirements [74]. We present the definition of functional requirements, adopted from Glinz [74], and the author's summary of non-functional requirements by Jacobsen et al. [99] in Table 15.

6.2.2 Dim_{ISO}: ISO/IEC 25010:2011 – System and Software Quality Models

As described in the previous subsection, several studies consider the binary distinction of requirements as insufficient to encompass the diversity of software requirements [27], [53], [74]. Therefore, researchers have already suggested numerous alternatives to specify the software requirements. Depending on the intentions, the alternatives differ in, e.g., their emphasis. One of these emphases is the aspect of software quality. Since software quality is critical for its longevity, we examined the impact of the eight dimensions of the *ISO Standard 25010:11 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE – System and software quality models* [98]. We describe the dimensions according to Pham et al. [162] in Table 16.

6.2.3 Dim_{SUST}: ShapeRE Dimensions

The Karlskrona manifesto [13] dimensions and the German *Fee Regulations for Services by Architects and Engineers – HOAI* [187] overlap significantly, which led to the combination of these nine dimensions [163]. Accordingly, we have adopted the dimensions *social, individual, ecological, economic,* and *technical* of the Karlskrona manifesto and its definition from Becker et al. [12] as well as *integrative, legal, design-aesthetics,* and *purpose* from HOAI [187] and *Guidelines for Sustainable Building* [61]. We defined the nine dimensions according to Pham et al. [162] in Table 17.

Table 15: Dimensions set – $Dim_{FR\text{-}NFR}$

DIMENSION	DESCRIPTION
Functional	Refers to an action that a system must be able to perform [74].
Non-functional	Refers to system properties, such as environmental and implementation constraints, performance, platform dependencies, maintainability, extensibility, and reliability [99], [74].

Table 16: Dimensions set – Dim_{ISO}

DIMENSION	DESCRIPTION
Functional suitability	Refers to the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions [98], [162].
Performance efficiency	Refers to the performance relative to the number of resources used under stated conditions[98], [162]
Compatibility	Describes the degree to which a product, system, or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment [98], [162].
Usability	Describes the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [98], [162].
Reliability	Describes the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time [98], [162].
Security	Describes the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization [98], [162]
Maintainability	Describes the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers [98], [162].
Portability	Describes the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another [98], [162].

Table 17: Dimensions set – Dim_{SUST}

DIMENSION	DESCRIPTION
Ecological	Covers the protection of the global and local ecosystem and saving natural resources, including, e.g., immediate production waste and energy consumption [12].
Economic	Is reflected by the degree to which life cycle costs are minimized, economic efficiency is improved, capital and product value are protected [12].
Social	Includes all factors that influence interhuman relationships between individuals or groups, and which promote "structures of mutual trust and communication in a social system and the balance between conflicting interests."[12].
Individual	"Covers individual freedom and agency (the ability to act in an environment), human dignity, and fulfillment; it includes individuals' ability to thrive, to exercise their rights, and to develop freely" [12], [162].
Technical	"It refers to maintenance, evolution, resilience, and the ease of transitions" [12] of artificial systems, such as soft- and hardware [12].
Legal	Refers to legal requirements and standards which are dependent on the usage, official regulations, company/domain specific etc. [187], [61], [162].
Integrative	Refers to the quality of its integration into existing systems, such as workflows, organizational structures, or other applications [187], [61], [162].
Design-aesthetic	Refers to the challenge to ensure a pleasing and beneficial use for the entire life of a product [187], [61], [162].
Purpose	Refers to fulfilling the product's purposes [187], [61], [162].

6.3 Research Design

In this section, we present our research questions in Subsection 6.3.2. and variables and hypotheses in Subsection 6.3.3. We describe our data preparation procedure in Subsection 6.3.4. and our analysis approach in Subsection 6.3.5. We outline the experiment settings and samples of our Exploratory Experiment I in Subsection 0 and Exploratory Experiment II in Subsection 0, respectively.

6.3.1 Research Questions

The following study presents our Exploratory Experiment I and II. We conducted these exploratory experiments to gain insight into the linguistic relativity hypothesis and our experiment design. To this end, we explore three aspects.

First, we examined whether our requirements dimensions have an impact on the number of identified requirements in general as well as on the number of sustainability-related requirements. Second, we were interested in whether the participants' individual traits could impact the results. To this end, we used the general number of identified requirements to examine whether there is a correlation. Third, the experiments should obtain insights for our main study, as reported in Chapter 7. We aimed for insights to improve our study design. To examine these aspects, we based our study on the following research questions:

RQ1. What impact do sets of requirements dimensions have on the number of identified requirements in general and related to sustainability?
RQ2. What impact do sets of requirements dimensions and individual traits have on the number of identified requirements in general?
RQ3. How do the participants perceive the experiments?

6.3.2 Variables and Hypotheses

Independent Variables – Set I. In our first exploratory experiment, our factor A_1 represented the requirements dimensions set $Dim_{FR\text{-}NFR}$ and Dim_{SUST}, as presented in Section 6.2. We included a control group which we labeled with Dim_{NO}. To examine whether individual traits impact the dependent variables, we asked the participants to provide details about them according to the factors B_1 – B_4 and B_6 – B_9 listed in Table 18. Furthermore, we evaluated the study design by asking for the participants' assessment according to the factors C_1 – C_4. We list our variables in Table 18.

Independent Variables – Set II. As a result of the first experiment, we assumed that $Dim_{FR\text{-}NFR}$ and Dim_{SUST} are not comparable due to the large difference in the number of dimensions. Therefore, we replaced $Dim_{FR\text{-}NFR}$ with Dim_{ISO} in our second exploratory experiment. Since we were interested in whether other factors might have influenced the results, we added the factors B_9 – B_{15}. The questions regarding the study design remained the same.

Dependent Variables. To investigate whether the linguistic relativity hypothesis applies to the requirements identification, we determined the number of identified requirements in general as our dependent variable Υ_{All}. According to the five dimensions of the Karlskrona manifesto [13], we defined Υ_{ECOL}, Υ_{ECON}, Υ_{SOC}, Υ_{IND}, and Υ_{TEC}, as our dependent variables to examine the number of sustainability-related requirements.

Hypotheses. We defined the following hypotheses to examine whether the set of requirements dimensions or individual factors influence the dependent variable. We rejected the null hypotheses if the p-value was higher than 0.05.

Hypotheses effect factor A_m:
$H_{0,A_m}: \mu_1 = \mu_2 = \cdots = \mu_i \quad (i = 1, \ldots, k_\alpha)$
H_{1,A_m}: at least one μ_i is different from the rest

Hypotheses effect factor B_q, that varies according to Table 18 :
$H_{0,B_q}: \mu_1 = \mu_2 = \cdots = \mu_j \quad (j = 1, \ldots, k_\beta)$
H_{1,B_q}: at least one μ_j is different from the rest

Hypotheses interaction effect:
$H_{0,A_m x B_q}: (\alpha\beta)_{ij} = 0$ for all pairs (ij) $\quad (i = 1, \ldots, k_\alpha; j = 1, \ldots, k_\beta)$
$H_{1,A_m x B_q}: (\alpha\beta)_{ij} \neq 0$ for at least one (ij) $\quad (i = 1, \ldots, k_\alpha; j = 1, \ldots, k_\beta)$

μ: expected mean of number of identified requirements
$(\alpha\beta)_{ij}$: effect due to any interaction between i-th level of A_m and j-th level of B_q
k_α: number of levels of factor A_m
k_β: number of levels of factor B_q
A_m: factor varies according to Table 18; m: number of factor group A
B_q: factor varies according to Table 18; q: number of factor group B

Table 18: Independent (IV) and dependent variables (DV)

TYPE	VARIABLES	EXPERIMENT	VALUES
IV	Factor A_1: Set of requirements dimensions	I	Dim_{NO}, $Dim_{FR\text{-}NFR}$, Dim_{SUST}
	Factor A_2: Set of requirements dimensions	II	Dim_{NO}, Dim_{ISO}, Dim_{SUST}
	Factor B_1: Age	I + II	Cardinal 1-3
	Factor B_2: Gender	I + II	Female, male, other
	Factor B_3: Occupation	I + II	Student, professional, other
	Factor B_4: IT background professional	I + II	No, yes
	Factor B_5: IT background education	II	No, yes
	Factor B_6: IT working experience	I + II	No, yes
	Factor B_7: RE education	I + II	No, yes
	Factor B_8: RE profession	I + II	No, yes
	Factor B_9: Familiarity with FR and NFR	I + II	No, yes
	Factor B_{10}: Familiarity with ISO Norms	II	No, yes
	Factor B_{11}: Familiarity with SUST dimensions	II	No, yes
	Factor B_{12}: Experience in requirements identification	II	No, yes
	Factor B_{13}: Programming experience	II	No, yes
	Factor B_{14}: UX/UI design experience	II	No, yes
	Factor B_{15}: Usage of grocery shopping app	II	No, yes
	Factor C_1: Comprehensibility of the assignment	I + II	Ordinal 1-5
	Factor C_2: Motivation	I + II	Ordinal 1-5
	Factor C_3: Interest	I + II	Ordinal 1-5
	Factor C_4: Completion time	I + II	Ordinal 1-5
	Factor C_5: Comprehensibility of the dimensions' definitions	I + II	Ordinal 1-5
DV	Υ_{ALL}: Number of all requirements	I + II	Integer
	Υ_{ECOL}: Number of ecological requirements	I + II	Integer
	Υ_{ECON}: Number of economic requirements	I + II	Integer
	Υ_{SOC}: Number of social requirements	I + II	Integer
	Υ_{IND}: Number of individual requirements	I + II	Integer
	Υ_{TEC}: Number of technical requirements	I + II	Integer

6.3.3 Data Preparation

In both experiments, we first removed the incomprehensible requirements, such as broken sentences whose content we could no longer understand. We determined how many requirements in general each group identified. Furthermore, we calculated the average number of requirements per participant for each group. We applied a deductive coding approach to counting the number of sustainability-related requirements. We prepared a coding guideline based on the definitions of the dimensions: *ecological, economic, social, individual,* and *technical*, as described in Subsection 6.2.3. We followed a single-blind approach to code the identified requirements of the participants. To this end, we involved three coders, including the author of this thesis. One coder shuffled all the given requirements and concealed from which group or dimension the requirement might originate. That way, the two other coders could determine the sustainability-related requirements according to the guideline without knowing the origin of each requirement. The two coders coded the requirements independently of each other. They coded, e.g., the requirement "The app should support environmentally friendly packaging" as ecological since it considers the environment or "The app should have low system requirements" as technical. We coded the requirement "Sharing favorite products" as social, which refers to the interaction between people. With this procedure, we could determine how many requirements of group A and group B align with the five sustainability dimensions. In case of a mismatch, meaning that the researchers coded the same requirement differently, we discussed these mismatches and agreed upon a final code. Initially, we provided for the factors B_4 – B_{15} a Likert scale of five levels as response options. We had to re-scale the five level-scale to a binary scale to gain an informative and not too fragmented statistical result. For example, the participants got for factor B_9 the statement: "I already knew about classifying software requirements in the dimensions 'functional' and 'non-functional' before the survey." They could choose one of the following options: 0 = Disagree, 1 = Disagree somewhat, 2 = Neither agree nor disagree, 3 = disagree somewhat, 4 = disagree. We mapped the answer 0 – 2 to group *No* and the others to group *Yes*. Furthermore, the participants had the option to choose from seventeen sectors, including IT, regarding their occupation and education background. This distinction also produced a result that was far too fragmented. To perform statistical analyses, we formed the two groups, *IT-related* and *not IT-related.* To answer RQ 3, no preparation of the participants' answers was necessary.

6.3.4 Analysis

We used both descriptive and inferential methods to evaluate the results. Regarding the descriptive statistics, we calculated the number of valid requirements for each group. By preparing the data, we count the number of requirements in general and related to one of the five sustainability dimensions. We calculated the means of the given answers to assess the experiment design.

We performed an *Analysis of Variance* (ANOVA) with SPSS statistics to test our hypotheses. Our goal was to examine whether applying a set of requirements dimensions influences the identification of requirements in general and sustainability-related requirements. Since the number of observations was fairly small, it was likely that the data was not normally distributed. Stevens [190] suggests that the data should be normally distributed, but the author also reports that the ANOVA is still robust enough to handle non-normally distributed data. To answer RQ1, we performed a one-way ANOVA, which follows Model 1. To investigate whether individual traits influence requirements identification, we conducted a two-way ANOVA. Our two-way ANOVA follows Model 2.

Model 1:

$y_{il} = \mu + \alpha_i + \varepsilon_{il}$ $(i = 1, ..., k_\alpha; l = 1, ..., n_i)$

y_{il} : value of dependent variable Υ_t of l-th subject under i-th level of factor A_m

μ: overall mean response

α_i: effect due to i-th level of factor A_m

ε_{il}: residual error of l-th subject under i-th level of factor A_m

k_α: number of levels of factor A_m

n_i: number of participants of i-th level of factor A_m

Υ_t: dependent variable varies according to Table 18, t: number of dependent variable

A_m: factor varies according to Table 18, m: number of factor group A

Model 2:

$y_{ijl} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijl}$ $(i = 1, ..., k_\alpha; j = 1, ... k_\beta; l = 1, ..., n_{ij})$

y_{ijl} : value of dependent variable Υ_t of l-th subject under i-th level of factor A_m and j-th level of factor B_q

μ: overall mean response

α_i: effect due i-th level of factor A_m

β_j: effect due j-th level of factor B_q

$(\alpha\beta)_{ij}$: effect due to any interaction between A_m and j-th level of B_q

ε_{il}: residual error of l -th subject under i-th level of factor A_1 and j-th level of factor B_q

k_α : number of levels of factor A_m

k_β: number of levels of factor B_q

n_{ij}: number of participants of i-th level of factor A_m and j-th level of factor B_q

Υ_t: dependent variable, varies according to Table 18

A_m: factor varies according to Table 18; m: number of factor group A

B_q: factor varies according to Table 18; q: number of factor group B

6.3.5 Exploratory Experiment I

Experiment Settings

We conducted a between-subjects design and used the tool LimeSurvey to create the online experiment. We designed the experiment according to the guidelines of Wohlin et al. [217]. The participants could perform the experiment in English or German. The completion time required about 25 minutes. As shown in Table 19, we structured the experiment in the following three parts:

1. **Introduction.** We explained the study purpose, the participation conditions, the estimated completion time, and the data protection guidelines. To avoid influencing the participants, we ensured that the descriptions and instructions never mentioned the topic of sustainability throughout the experiment. The participants had to give explicit consent to start the experiment.
2. **Assignment.** Once they had provided their consent, we assigned the participants randomly to one of the three experimental groups. To prepare for the task ahead, we asked the participants to imagine being part of a development team that is supposed to identify requirements for a grocery shopping app. Depending on the experimental group, the participants received an explanation of their respective dimensions and corresponding examples, as depicted in Figure 11. We informed the participants that they would get eighteen minutes to complete the assignment. For group B ($Dim_{Fr\text{-}NFR}$) and group C (Dim_{SUST}), the time was equally divided according to the number of dimensions.
3. **Questionnaire and Conclusion.** The last part consisted of questions regarding the factors B_1 – B_4 and B_6 – B_9, as presented in Table 18. Before submitting their answers, we provided the participants with the opportunity to leave comments. After completing the study, we clarified to the participants that they were participating in an experiment, provided a summary of our research project on sustainable software design, and thanked them for their participation.

Table 19: Exploratory Experiment I – procedure

PART	DESCRIPTION	TIME LIMIT
1. Introduction	Explanation of participation conditions and assignment setting	-
2. Assignment $Dim_{FR\text{-}NFR}$, Dim_{SUST}, or Dim_{NO}	Explanation of dimensions and assignments	18 min.
3. Questionnaire and conclusion	Demographics, feedback, and summary	-

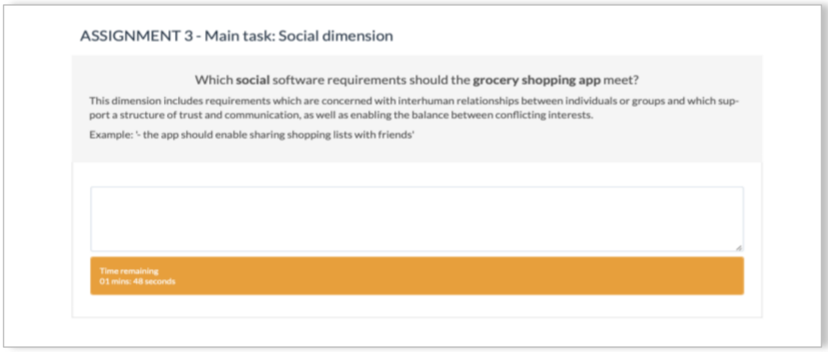


Figure 11: Example assignment

Sample

To acquire participants, we applied a mix of convenience and snowball sampling [217], [107]. To this end, we contacted acquaintances. We did not give any incentives to the participants. Overall, a total of 14 participants completed the study. One other participant submitted an uncompleted study. Hence, we had to exclude this submission. Of the remaining 14 participants, one person was diverse, seven male and six female. One participant belonged to the age group 65 years or above, one belonged to the group 35 to 64, and the other 12 belonged to the group 18 to 34 years. We had four professionals, eight students, and two had other occupations. Ten participants had an IT-related background, seven had IT working experience, ten learned about requirements engineering within their education, eight had experience with requirements engineering in their professional life, and eight were familiar with the dimensions functional and non-functional. We provide an overview of the randomized distribution of our sample into the experimental groups in Table 20.

Table 20: Exploratory Experiment I – groups

GROUP	NO. PARTICIPANTS	DIMENSIONS SET
A	4	Dim_{NO}
B	5	$Dim_{FR\text{-}NFR}$
C	5	Dim_{SUST}

6.3.6 Exploratory Experiment II

Experiment Settings

We conducted another between-subjects design and used the tool LimeSurvey to create the online experiment. We kept the guidelines of Wohlin et al. [217] and adjusted the study design additionally according to Reips and Krantz [170], and Lietz [119]. According to the additional adjustments, we divided our questionnaire into two parts. We used the first part of the questionnaire as a warm-up and the second part as a cool-down session. Furthermore, we included a trial task to provide the participants with an opportunity to get familiar with the assignment type before engaging with the primary assignment. The participants could perform the experiment in English or German. The completion time required about 30 minutes. As shown in Table 21, we structured the second experiment in the following five parts:

1. **Introduction.** We explained the purpose of the study, the participation conditions, the estimated completion time, the study procedure, and the data protection guidelines. Furthermore, we asked the participants not to consult additional help and to avoid any distractions. Additionally, we included information about an opportunity to participate in a raffle after completing the study. To prevent influencing the participants, we made sure that we did not mention the topic of sustainability or software quality throughout the experiment. The participants had to give explicit consent to be forwarded to the experiment.
2. **Questionnaire I.** After their approval, the participants started with warm-up questions regarding their software development experiences according to the variables B_4 – B_8, as presented in Table 18.
3. **Assignment.** Without their knowledge, we assigned the participants randomly to one of three groups A, B, or C, as shown in Table 22. According to their assigned experiment group, we introduced the participants to the trial and main task. First, the participants received the information that the tasks would be time-boxed: two minutes for the trial task and eighteen minutes for the primary task. Furthermore, group B was introduced to the set of requirements dimensions Dim_{ISO} and group C to Dim_{SUST} according to the description in Section 6.2. After the participants confirmed reading the instructions, we forwarded the participants to the trial task. Using a trial task, the participants learned how we set the primary task. We asked the participants to identify as many requirements as possible for a fitness app in keywords. Depending on their group, the participants were given one random dimension from their assigned set or none if they belonged to the control group. Once the participants completed the trial task, we proceeded them to the main task. In group B and group C, we sequentially showed the participants their assigned dimensions in random order. For each sequence, the participants of group B got two minutes and fifteen seconds, and group C received two minutes for each sequence. The interface remained the same, as shown in Figure 11.
4. **Questionnaire II.** To control the independent variables B_1 – B_3 and B_9 – B_{15}, we asked the participants for their demographic details and experience with requirements dimensions. Before submitting their answers, we provided the participants with an opportunity to leave comments.
5. **Conclusion.** After completing the study, we clarified to the participants that they took part in an experiment, provided a summary of our research endeavor in sustainable software design, gave them the information to participate in the raffle, and thanked them for their participation.

Table 21: Exploratory Experiment II – procedure

PART	CONTENT	TIME LIMIT
1. Introduction	Explanation of participation conditions	-
2. Questionnaire I	Experience with software development	-
3. Assignment Dim_{ISO}, Dim_{SUST}, or Dim_{NO}	Explanation of dimensions	-
	Trial task (one random dimension of the assigned set)	2 min.
	Main task (all dimensions randomized of the assigned set)	18 min.
4. Questionnaire II	Demographics	-
5. Conclusion	Summary	-

Sample

For Exploratory Experiment II, we applied a mix of convenience and snowball sampling as well [217], [107]. This time, we gave the participants a chance to participate in a raffle for two vouchers. Since we intended to offer a raffle for the next study, we wanted to test the process in this trial run. Overall, we included a total of 15 people in our analysis. Four were female, and eleven were male. All participants belonged to the age group 18 to 34 years. Regarding the participants' occupations, one was unemployed, four were employees, and ten were students. Five participants studied an IT-related program, and six had an IT-related occupation. Five participants had IT working experience, ten programming experience, seven UX/UI-design experience, seven learned about requirements engineering during their education, and nine had professional experience in requirements engineering. Six participants were experienced with requirements identification, seven knew the dimensions functional and non-functional, four the ISO dimensions, and two were familiar with our sustainability dimensions. Two participants used a grocery shopping app. We provide an overview of the randomized distribution of our participants into the experimental groups in Table 22.

Table 22: Exploratory Experiment II – groups

GROUP	NO. PARTICIPANTS	DIMENSIONS SET
A	5	Dim_{NO}
B	5	Dim_{ISO}
C	5	Dim_{SUST}

6.4 Results Exploratory Experiment I

In the following section, we present the descriptive and inferential statistics of our Exploratory Experiment I. We report the results regarding the impact of applying the requirements dimensions in Subsection 6.4.1 and then the results considering individual traits in Subsection 6.4.2. We conclude with the feedback on our experiment design 6.4.3.

6.4.1 Impact of Requirements Dimensions

Descriptive Statistics

In total, Exploratory Experiment I resulted in 178 requirements across all groups. As we present in Figure 12, group A (Dim_{NO}) identified 43 requirements, group B (Dim_{FR-NFR}) 42 requirements, and group C (Dim_{SUST}) came up with 93 requirements. However, we must consider that the control group consists only of four participants. In Table 23, we list the means and standard deviations of the identified requirements according to our dependent variables, as presented in Table 18. We observed that applying any set of dimensions does not mean that the number of identified requirements in general will increase. The participants of group B (Dim_{FR-NFR}) identified 8.60 requirements on average, and group C identified 18.60 requirements. Our control group A (Dim_{NO}) participants identified 10.50 requirements on average.

Group B (Dim_{FR-NFR}) and group C (Dim_{SUST}) participants identified the same number of technical requirements with an average of 2.60 requirements per participant compared to the control group A (Dim_{NO}). However, participants of group C (Dim_{SUST}) identified more requirements on average, with 1.80 ecological and economic requirements, 1.60 social and 2.0 individual requirements compared to participants of the other groups. Except for the technical requirements, the control group A (Dim_{NO}) identified more requirements per person than group B (Dim_{FR-NFR}) regarding the other dimensions.

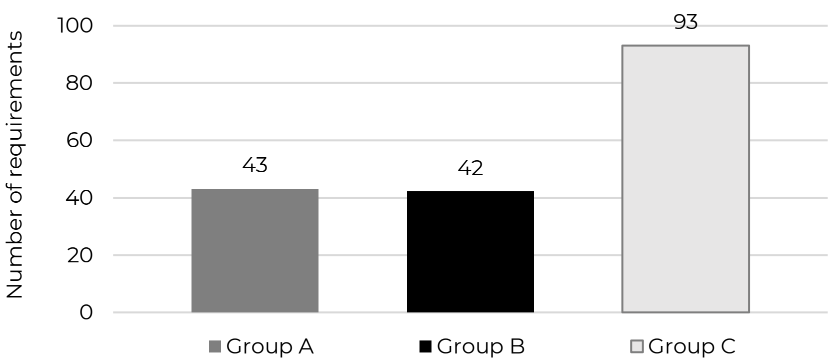


Figure 12: Exploratory Experiment I – total number of requirements of each group

Table 23: Exploratory Experiment I – one factor descriptive statistic

	IV								
	DIM_NO			DIM_FR-NFR			DIM_SUST		
DV	N	μ	σ	N	μ	σ	N	μ	σ
Y_{ALL}	4	10.50	3.317	5	8.60	4.336	5	18.60	8.081
Y_{ECOL}	4	0.75	0.957	5	0.40	0.548	5	1.60	1.140
Y_{ECON}	4	1.50	1.000	5	0.00	0.00	5	1.80	1.304
Y_{SOC}	4	0.25	0.500	5	0.20	0.447	5	1.60	0.548
Y_{IND}	4	1.75	1.500	5	1.00	0.707	5	2.00	0.707
Y_{TEC}	4	0.50	0.577	5	2.60	2.074	5	2.60	0.548

Inferential Statistics

We can observe that applying requirements dimensions significantly impacts the number of identified requirements in general. This observation also applies to the economic and social requirements. Therefore, we can reject the null hypothesis H_{0,A_1}, as presented in Subsection 6.3.2, for these cases. Regarding the ecological, individual, and technical requirements, we could not observe a significant impact of applying requirements dimensions. Hence, we accept the null hypothesis H_{0,A_1} for these dependent variables. We list the results of our inferential statistics in Table 24.

Table 24: Exploratory Experiment I – one factor inferential statistics

DV	TYPE III SS	DF	MEAN SQ.	F	P-VALUE
Y_{ALL}	277.457	2	138.729	4.131	**0.046**
Y_{ECOL}	3.779	2	1.889	2.271	0.149
Y_{ECON}	9.129	2	4.564	5.123	**0.027**
Y_{SOC}	6.107	2	3.054	12.214	**0.002**
Y_{IND}	2.679	2	1.339	1.37	0.294
Y_{TEC}	12.6	2	6.3	3.572	0.064

6.4.2 Impact of Individual Traits

Descriptive Statistics

In Table 25, we list the means and standard deviations of the identified requirements in general according to the factors B_1 – B_4 and B_6 – B_9, as presented in Table 18. Regarding the age, we could not make any unusual observations. However, male participants of group B (*Dim_{FR-NFR}*) identified with 7.67 requirements on average fewer requirements than the male participants of the control group A (*Dim_NO*) with an average of 11.5 requirements and group C (*Dim_SUST*) with 17.50 requirements. Regarding the occupations, professionals also identified fewer requirements with set *Dim_FR-NFR* than the other participants, as illustrated in Figure 13. Participants of the control group A without an IT-related sector identified more than their peers in group B. In group C (*Dim_SUST*), all the participants had an IT background. They achieved the best results with 18.60 requirements on average. Regarding the IT working experience, all participants in group B identified fewer requirements than the others. Group B (*$Dim_{FR\text{-}NFR}$*) participants who did not perform requirements engineering in their professional life identified fewer requirements than the others, as shown in Figure 14. Similarly, for the participants who learned about requirements engineering within their education. The participants of group B (*$Dim_{FR\text{-}NFR}$*) who were unfamiliar with the dimensions *functional* and *non-functional* also identified fewer requirements than their peers, as shown in Figure 15.

Table 25: Exploratory Experiment I – two factors descriptive statistics of identified requirements in general

FACTOR B_q		FACTOR A_1								
		DIM_{NO}			$DIM_{FR\text{-}NFR}$			DIM_{SUST}		
		N	μ	σ	N	μ	σ	N	μ	σ
B_1: Age	18-34	2	8.00	0.00	5	8.60	4.37	5	18.60	8.08
	35-64	1	11.00	-	-	-	-	-	-	-
	65 +	1	15.00	-	-	-	-	-	-	-
B_2: Gender	Female	2	9.50	2.12	2	10.00	2.83	2	23.50	10.61
	Male	2	11.50	4.95	3	7.67	5.51	2	17.50	6.36
	Diverse	-	-	-	-	-	-	1	11.00	-
B_3: Occupation	Student	2	8.00	0.00	3	9.00	4.58	3	21.33	10.02
	Pro.	2	13.00	2.83	1	4.00	-	1	16.00	-
	Other	-	-	-	1	12.00	-	1	13.00	-
B_4: Sector	Not-IT rel.	2	13.00	2.83	2	8.50	4.95	5	18.60	8.08
	IT rel.	2	8.00	0.00	3	8.67	5.033	-	-	-
B_6: IT working experience	No	3	11.33	3.51	2	10.00	2.83	2	21.00	14.14
	Yes	1	8.00	-	3	7.67	5.51	3	17.00	4.58
B_7: RE education	No	2	13.00	2.83	1	12.00	-	1	13.00	-
	Yes	2	8.00	0.00	4	7.75	4.50	4	20.00	8.602
B_8: RE profession	No	3	11.33	3.51	2	8.00	5.66	1	31.00	-
	Yes	1	8.00	-	3	9.00	4.58	4	15.50	4.80
B_9: Familiarity FR & NFR	No	1	11.00	-	3	7.00	4.36	2	26.50	6.36
	Yes	3	10.33	4.04	2	11.00	4.24	3	13.33	2.52

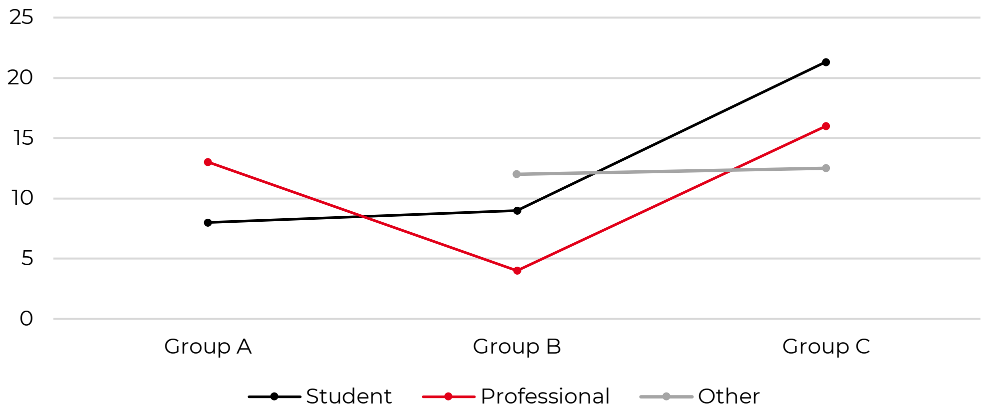


Figure 13: Exploratory Experiment I – average number of requirements in general and occupation

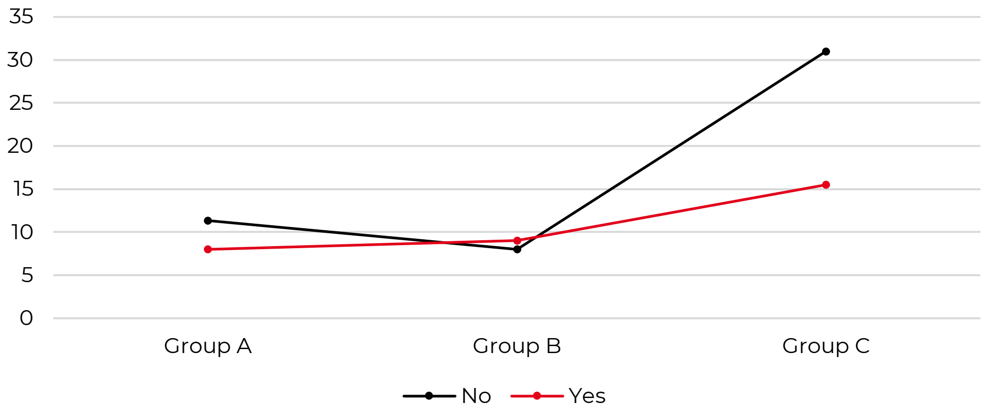


Figure 14: Exploratory Experiment I – average number of requirements in general and requirements engineering in profession

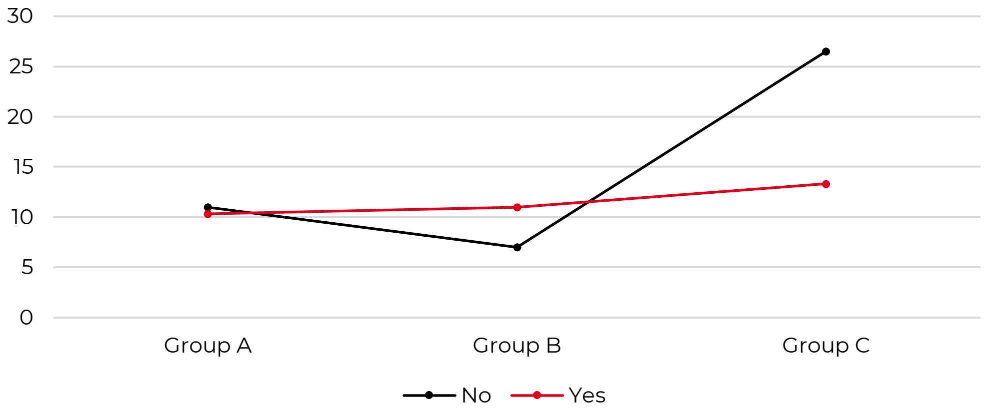


Figure 15: Exploratory Experiment I – average number of requirements in general and familiarity with the dimensions functional and non-functional requirements

Inferential Statistics

We observed that applying sets of requirements dimensions significantly influences the number of identified requirements in general, as presented in Table 24. However, considering the individual factors, the requirements dimensions have a significant influence in combination with the factors B_8 (RE profession), and B_9 (familiarity with the dimensions functional and non-functional requirements). Figure 14 illustrates that participants without prior requirements engineering experience in their working life are likely to identify more requirements with set *Dim*$_{SUST}$. However, only one person in group C matched these traits. Similarly, participants unfamiliar with the dimensions *functional* and *non-functional* identified more requirements with Dim_{SUST} than with $Dim_{FR\text{-}NFR}$, as shown in Figure 15. Moreover, we observed a significant interaction effect for this factor.

Hence, we can reject the null hypothesis H_{0,A_1} regarding the factors B_8 and B_9. Additionally, we observed an interaction effect between our factor B_9 and the dimensions set. Therefore, we can reject the null hypothesis $H_{0,A_1 x B_9}$. We can accept the null hypotheses regarding the other factors.

Table 26: Exploratory Experiment I – two factors inferential statistics regarding requirements in general

FACTORS		T III SS	DF	MEAN SQ.	F	P-VALUE
Age	A_1: Dim. set	302.267	2	151.133	4.043	0.056
	B_1: Age	33.000	2	16.500	0.441	0.656
	Dim. set * Age	0.000	0	-	-	-
Gender	A_1: Dim. set	333.333	2	166.667	4.654	0.052
	B_2: Gender	86.357	2	43.178	1.206	0.355
	Dim. set * Gender	32.083	2	16.042	0.448	0.656
Occupation	A_1: Dim. set	147.940	2	73.970	1.771	0.249
	B_3: Occupation	8.265	2	4.132	0.099	0.907
	Dim. set * Occup.	112.683	3	37.561	0.899	0.494
Sector	A_1: Dim. set	277.484	2	138.742	3.626	0.070
	B_5: Sector	12.742	1	12.742	0.333	0.578
	Dim. set * Sector	14.561	1	14.561	0.381	0.553
IT working experience	A_1: Dim. set	289.352	2	144.676	3.452	0.083
	B_6: IT working exp.	31.148	1	31.148	0.743	0.414
	Dim. set * IT w. exp.	1.683	2	0.841	0.020	0.980
RE education	A_1: Dim. set	88.004	2	44.002	1.211	0.347
	B_7: RE education	1.446	1	1.446	0.040	0.847
	Dim. set * RE edu.	75.635	2	37.817	1.041	0.397
RE profession	A_1: Dim. set	466.602	2	233.301	11.132	**0.005**
	B_8: RE profession	93.081	1	93.081	4.441	0.068
	Dim. set * RE pro.	133.482	2	66.741	3.184	0.096
Familiarity FR & NFR	A_1: Dim. set	318.878	2	159.439	8.993	**0.009**
	B_9: Familiar FR&NFR	32.231	1	32.231	1.818	0.214
	Dim. set * FR&NFR	185.583	2	92.791	5.234	**0.035**

6.4.3 Feedback Experiment Design

To examine our experiment design, we gave our participants five statements according to the variables C_1 – C_4, as presented in Table 18. They could agree or disagree with the statements using a five-point Likert scale. In the following, a high value also corresponds to a high agreement. The participants assessed the study as rather interesting, with an average of 3.73. They were quite motivated since they rated the statement “I was motivated while participating in the survey” with an average of 4.06. Regarding the comprehensibility of the study, the participants reported that they slightly agree, with an average score of 3.53. Eight participants perceived the given time for the assignments as “just right,” four as “too short,” and the remaining three participants evaluated the given time as “too long.” Two of these three participants belonged to the control group.

6.5 Results Exploratory Experiment II

In the following section, we present the descriptive and inferential statistics of our Exploratory Experiment II. We report the results regarding the impact of applying the requirements dimensions in Subsection 6.4.1 and then the results considering individual traits in Subsection 6.5.2. We conclude with the feedback on our second experiment design 6.5.3.

6.5.1 Impact of Requirements Dimensions

Descriptive Statistics

In total, Exploratory Experiment II resulted in 251 requirements across all groups. As presented in Figure 16, group A (Dim_{NO}) identified 50 requirements, group B (Dim_{ISO}) identified 94 requirements, and group C (Dim_{SUST}) came up with 107 requirements. Table 27 lists the means and standard deviations of the identified requirements according to our dependent variables, as we present in Table 18. In this experiment, we observed that applying a set of dimensions increased the number of identified requirements in general. Participants of group B (Dim_{ISO}) identified 18.80 requirements on average, and group C identified 21.40 requirements. The participants of our control group A (Dim_{NO}) identified 10.00 requirements on average.

Participants of group B (Dim_{ISO}) identified more technical requirements with an average of 7.20. Group C (Dim_{SUST}) participants identified only 3.00 technical requirements on average, and group A (Dim_{NO}) participants 1.00. However, group C (Dim_{SUST}) participants identified more requirements, with 1.80 social and 1.80 individual requirements, 2.20 economic and 2.60 ecological requirements compared to the participants of the other groups. Group B (Dim_{ISO}) and group A (Dim_{NO}) participants identified with 1.00 the same number of ecological requirements on average. On the one hand, the group A (Dim_{NO}) participants identified more economic requirements with 1.20 requirements on average and 0.80 individual requirements. On the other hand, the group B (Dim_{ISO}) participants identified more social requirements with 0.60 requirements than the control group A (Dim_{NO}) participants.

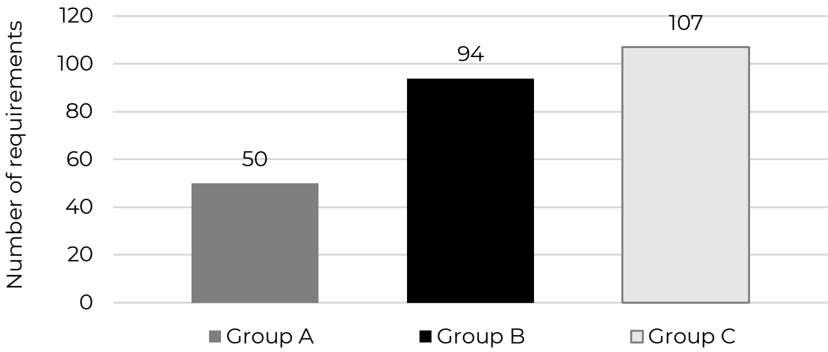


Figure 16: Exploratory Experiment II – total number of requirements of each group

Table 27: Exploratory Experiment II – one factor descriptive statistics regarding sustainable requirements and in general

	IV								
	DIM_{NO}			$DIM_{FR\text{-}NFR}$			DIM_{SUST}		
DV	N	μ	σ	N	μ	σ	N	μ	σ
Y_{ALL}	5	10.00	4.796	5	18.80	6.058	5	21.40	1.949
Y_{ECOL}	5	1.00	1.732	5	1.00	0.707	5	2.60	1.14
Y_{ECON}	5	1.20	1.095	5	0.40	0.894	5	2.20	0.837
Y_{SOC}	5	0.00	0.000	5	0.60	1.342	5	1.80	0.837
Y_{IND}	5	0.80	1.304	5	0.20	0.447	5	1.80	0.837
Y_{TEC}	5	1.00	1.225	5	7.20	2.588	5	3.00	1.581

Inferential Statistics

We can observe that applying requirements dimensions significantly impacts the number of identified requirements in general. This observation also applies to the economic, social, and technical requirements. Therefore, we can reject the null hypothesis H_{0,A_2} for these cases, as presented in Subsection 6.3.2. Regarding the ecological and individual requirements, we could not observe a significant impact of applying requirements dimensions. Hence, we accept the null hypothesis H_{0,A_2} for these dependent variables. We list the results of our inferential statistics in Table 28.

Table 28: Exploratory Experiment II – one factor inferential statistics

DV	T III SS	DF	MEAN SQ.	F	P-VALUE
Y_{ALL}	356.933	2	178.467	8.431	**0.005**
Y_{ECOL}	8.533	2	4.267	2.667	0.110
Y_{ECON}	8.133	2	4.067	4.519	**0.034**
Y_{SOC}	8.400	2	4.200	5.04	**0.026**
Y_{IND}	6.533	2	3.267	3.769	0.054
Y_{TEC}	100.133	2	50.067	14.037	**0.001**

6.5.2 Impact of Individual Traits

Descriptive Statistics

Table 29 lists the means and standard deviations of the identified requirements in general according to the individual factor B_1 – B_{15}, as presented in Table 18. Since we had only one age group, we could not observe any variances. We observed that experienced participants in requirements identification of group B (Dim_{ISO}) identified with 20.50 requirements on average, slightly more than their peers in group C (Dim_{SUST}) with 20.00 and in control group A (Dim_{NO}) with 9.00, as illustrated in Figure 17. However, there was only one experienced person in group A and group C. We observed that the participants identified more requirements with an increasing number of dimensions regarding the other factors. We illustrated this observation using the factor B_8 (RE profession) in Figure 18 and the factor B_9 (familiarity with the dimensions functional and non-functional requirements) in Figure 19.

Inferential Statistics

We observed that applying sets of requirements dimensions significantly influences the number of identified requirements in general, as presented in Table 28. Compared to the result of Exploratory Experiment I, as reported in Section 6.4, the application of requirements dimensions remains significant, and most of the individual traits had no significant influence, except for B_8 (RE profession). Figure 18 demonstrates that participants with prior requirements engineering experience in their working life are likely to identify more requirements exposed to all the level treatments. This individual trait had a significant influence as well. Therefore, we can reject the null hypotheses H_{0,A_2} and H_{0,B_8} for this case. Regarding the other individual traits, we can accept the null hypotheses $H_{0,B_1-B_7}$and $H_{0,B_9-B_{15}}$, while we can reject the null hypothesis H_{0,A_2}.

6.5.3 Feedback Experiment Design

To examine our second experiment design, we gave our participants the same five statements according to the variables C_1-C_5, as presented in Table 18. They could agree or disagree with the statements using a five-point Likert scale. In the following, a high value also corresponds to a high agreement. The participants assessed the study as rather interesting, with an average of 3.27. They were quite motivated since they rated the statement “I was motivated while participating in the survey” with an average of 3.93. Regarding the study’s comprehensibility, the participants reported higher comprehensibility with an average score of 4.00 compared to the first experiment. Eight participants perceived the given time for the assignments as “just right” and two as “too short.” Three participants evaluated the given time as “too long,” and the remaining two provided no specification.

Table 29: Exploratory Experiment II – two factors descriptive statistics of identified requirements in general

FACTOR B_q		FACTOR A_2								
		DIM_{NO}			$DIM_{FR\text{-}NFR}$			DIM_{SUST}		
		N	μ	σ	N	μ	σ	N	μ	σ
B_1: Age	18-34	5	10	4.796	5	18.8	6.058	5	21.4	1.949
	35-64	-	-	-	-	-	-	-	-	-
	65 +	-	-	-	-	-	-	-	-	-
B_2: Gender	Female	2	10.00	7.071	1	20.00	-	1	20.00	-
	Male	3	10.00	4.583	4	18.50	6.952	4	21.75	2.062
	Diverse	-	-	-	-	-	-	-	-	-
B_3: Occupation	Student	2	10.00	7.071	4	20.50	5.447	4	21.75	2.062
	Pro.	2	7.50	2.121	1	12.00	-	1	20.00	-
	Other	1	15.00	-	-	-	-	-	-	-
B_4: Sector occupation	Not-IT rel.	5	10.00	4.796	1	19.00	-	3	21.00	1.732
	IT related	-	-	-	4	18.75	6.994	2	22.00	2.828
B_5: Sector education	Not-IT rel.	5	10.00	4.796	2	17.00	2.828	3	20.00	0.000
	IT related	-	-	-	3	20.00	8.000	2	23.50	0.707
B_6: IT working experience	No	4	10.25	5.500	2	17.50	3.536	4	20.75	1.500
	Yes	1	9.00	-	3	19.67	8.021	1	24.00	-
B_7: RE education	No	5	10.00	4.796	0	-	-	3	21.33	2.309
	Yes	0	-	-	5	18.80	6.058	2	21.50	2.121
B_8: RE profession	No	2	5.50	0.707	1	12.00	-	3	21.00	1.732
	Yes	3	13.00	3.464	4	20.50	5.447	2	22.00	2.828
B_9: Familiarity FR & NFR	No	3	8.67	5.508	2	21.50	9.192	3	22.33	2.082
	Yes	2	12.00	4.243	3	17.00	4.359	2	20.00	-
B_{10}: Familiarity ISO 25010:11	No	5	10.00	4.796	4	18.50	6.952	2	21.50	2.121
	Yes	0	-	-	1	20.00	-	3	21.33	2.309
B_{11}: Familiarity Sustain. Set	No	4	8.75	4.500	4	18.50	6.952	5	21.40	1.949
	Yes	1	15.00	-	1	20.00		-	-	-
B_{12}: Programm. experience	No	3	8.67	5.508	-	-	-	2	20.00	0.000
	Yes	2	12.00	4.243	5	18.80	6.058	3	22.33	2.082
B_{13}: UI/ UX experience	No	4	10.25	5.500	1	12.00	-	3	21.33	2.309
	Yes	1	9.00	-	4	20.50	5.447	2	21.50	2.121
B_{14}: Identificat. experience	No	4	10.25	5.500	1	12.00	-	4	21.75	2.062
	Yes	1	9.00	-	4	20.50	5.447	1	20.00	-
B_{15}: Use grocery shop. app	No	4	11.00	4.899	4	20.50	5.447	5	21.40	1.949
	Yes	1	6.00	-	1	12.00	-	0	-	-

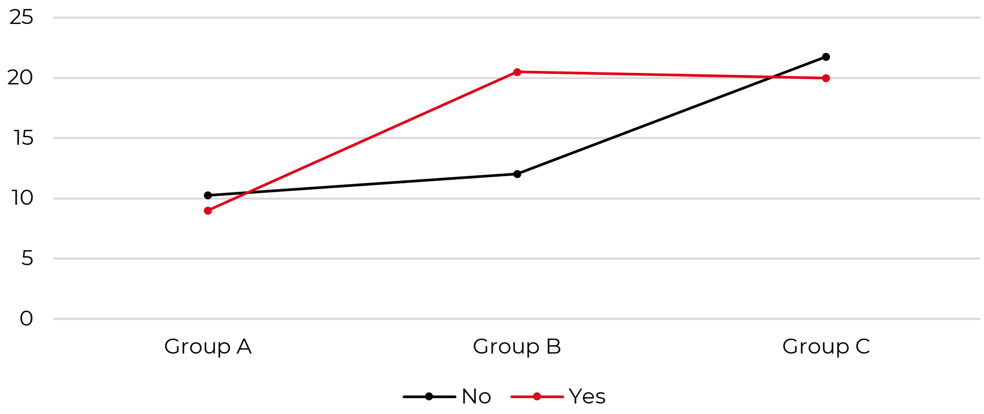


Figure 17: Exploratory Experiment II – average number of requirements in general and experience in requirements identification

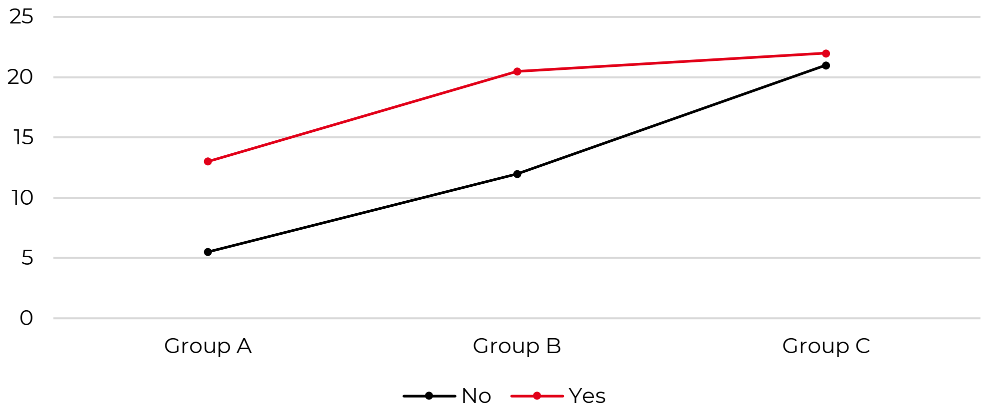


Figure 18: Exploratory Experiment II – average number of requirements and requirements engineering in profession

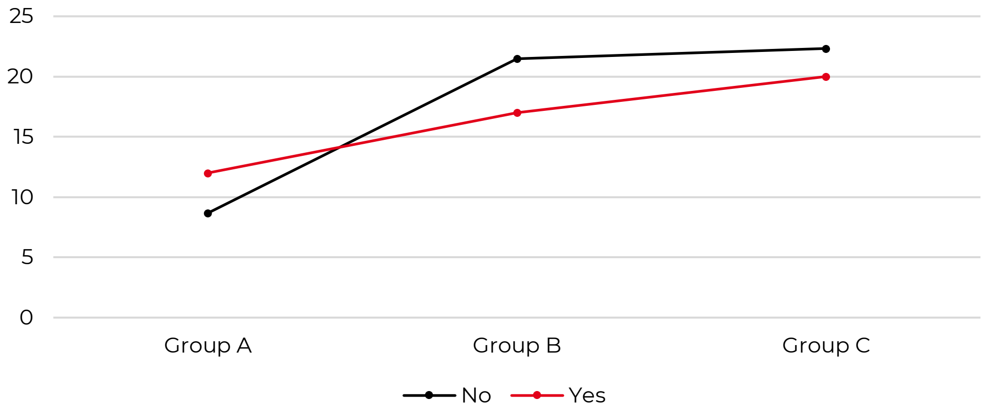


Figure 19: Exploratory Experiment II – average number of requirements in general and familiarity with the dimensions functional and non-functional requirements

Table 30: Exploratory Experiment II – two factors inferential statistics regarding requirements in general

FACTORS		T III SS	DF	MEAN SQ.	F	P-VALUE
Age	A_2: Dim. set	356.933	2	178.467	8.431	**0.005**
	B_1: Age	0	-	-	-	-
	Dim. set * Age	0	-	-	-	-
Gender	A_2: Dim. set	281.950	2	140.975	5.080	**0.033**
	B_2: Gender	0.019	1	0.019	0.001	0.980
	Dim. set * Gender	4.236	2	2.118	0.076	0.927
Occupation	A_2: Dim. set	271.235	2	135.617	6.944	**0.018**
	B_3: Occupation	77.696	2	38.848	1.989	0.199
	Dim. set * Occupation	22.465	2	11.233	0.575	0.584
Sector occupation	A_2: Dim. set	258.033	2	129.017	5.105	**0.030**
	B_4: Sector Occ.	0.270	1	0.270	0.011	0.920
	Dim. set * Sector Occ.	0.750	1	0.750	0.030	0.867
Sector education	A_2: Dim. set	218.044	2	109.022	4.771	**0.035**
	B_5: Sector education	25.350	1	25.350	1.109	0.317
	Dim. set * Sector edu.	0.150	1	0.150	0.007	0.937
IT working experience	A_2: Dim. set	278.405	2	139.202	5.249	**0.031**
	B_6: IT working exp.	5.208	1	5.208	0.196	0.668
	Dim. set * IT exp	9.033	2	4.517	0.170	0.846
Programm. experience	A_2: Dim. set	282.415	2	141.208	6.031	**0.019**
	B_7: Programm. exp.	19.267	1	19.267	0.823	0.386
	Dim. set * IT Pro. Exp	0.600	1	0.600	0.026	0.876
UI/ UX experience	A_2: Dim. set	267.426	2	133.713	6.174	**0.021**
	B_8: UI/UX exp.	16.502	1	16.502	0.762	0.405
	Dim. set * UI/UX exp.	46.226	2	23.113	1.067	0.384
RE education	A_2: Dim. set	251.248	2	125.624	5.441	**0.023**
	B_9: RE education	0.033	1	0.033	0.001	0.970
	Dim. set * RE edu.	0.000	0	-	-	-
RE profession	A_2: Dim. set	361.988	2	180.994	12.776	**0.002**
	B_{10}: RE profession	99.086	1	99.086	6.994	**0.027**
	Dim. set * RE pro.	36.188	2	18.094	1.277	0.325
Identificat. experience	A_2: Dim. set	204.633	2	102.317	4.784	**0.038**
	B_{11}: Identifat. exp.	8.067	1	8.067	0.377	0.554
	Dim. set * Ident. exp.	53.433	2	26.717	1.249	0.332
Familiarity FR & NFR	A_2: Dim. set	320.867	2	160.433	6.881	**0.015**
	B_{12}: Familiar. FR&NFR	4.900	1	4.900	0.210	0.658
	Dim. set * FR&NFR	39.267	2	19.633	0.842	0.462
Familiarity ISO 25010:11	A_2: Dim. set	258.665	2	129.332	5.129	**0.029**
	B_{13}: Familiar. ISO	0.853	1	0.853	0.034	0.858
	Dim. set * ISO	1.333	1	1.333	0.053	0.823
Familiarity SUST Set	A_2: Dim. set	273.027	2	136.513	6.178	**0.018**
	B_{14}: Familiar. SUST	24.025	1	24.025	1.087	0.322
	Dim. set * SUST	9.025	1	9.025	0.408	0.537
Use grocery shop. app	A_2: Dim. set	194.323	2	97.162	5.514	**0.024**
	B_{15}: Usage groc. app	72.900	1	72.900	4.137	0.069
	Dim. set * groc. app	4.900	1	4.900	0.278	0.609

6.6 Discussion

In the following section, we discuss the implications of our results in Subsection 6.6.1 and threats to validity in Subsection 6.6.2.

6.6.1 Implications

Guiding Requirements Identification. The results of Exploratory Experiment I indicate that applying requirements dimensions can positively and negatively affect identifying requirements. We found that participants who received Dim_{SUST} identified more requirements with an average result of 18.6 per person than their comparison groups. Participants who received $Dim_{FR\text{-}NFR}$ identified fewer requirements with an average result of 8.6 than the control group participants, which identified 10.50 requirements per person. Despite this, according to our inferential statistics, applying requirements dimensions significantly influences the identification of requirements in general. Regarding sustainability requirements, we could only observe a significant influence relating to the economic and social dimensions. We only observed a significant influence of applying requirements dimensions for the individual traits in two cases. Participants without prior experience in requirements engineering or were familiar with the dimensions *functional* and *non-functional* performed better with Dim_{SUST} than Dim_{ISO}. The experienced participants could identify more requirements with an increasing number of dimensions. Hence, we can assume inexperienced stakeholders, such as users, will likely identify more requirements with Dim_{SUST} than with $Dim_{FR\text{-}NFR}$ or without dimensions.

The results of Exploratory Experiment II indicate that the provision of requirements dimensions positively impacts the number of requirements identified in general. The participants identified 21.4 requirements per person with Dim_{SUST}, 18.8 requirements with Dim_{ISO}, and the control group identified 10.0 requirements. In terms of the sustainability-related requirements, we observed significant results only in relation to the economic, social, and technical dimensions. Taking into account the individual traits, we observed that factor A_2 always had a significant impact on the number of requirements identified in general. However, the results of Exploratory Experiment II showed that regardless of the treatment level, participants who are professionally experienced in requirements engineering performed better than the inexperienced participants.

Overall, we can conclude that providing the requirements dimensions Dim_{SUST} and Dim_{ISO} will probably increase the number of requirements identified in general. In contrast, without dimensions or with $Dim_{FR\text{-}NFR}$, fewer requirements might be identified. According to this observation, the set we select can significantly impact the number of identified requirements. Furthermore, it appears that applying our chosen set of requirements dimensions could inspire our participants to consider social- and economic-related requirements. Moreover, our results suggest that applying these dimensions might reduce the risk of incomplete requirements.

Regarding our selected individual traits, we found that they do not play a significant role in many cases. However, we cannot determine whether prior professional experience in requirements engineering significantly impacts the requirements identification. According to Exploratory Experiment II, experienced participants performed better regardless of the treatment level of factor A_2, while in Exploratory Experiment I, they only performed slightly better with $Dim_{FR\text{-}NFR}$. In contrast, both exploratory experiments indicate that inexperienced users could identify more requirements with Dim_{SUST}. We suggest being cautious when using technical-focused dimensions, such as the ones of $Dim_{FR\text{-}NFR}$ and Dim_{ISO}, with inexperienced users when identifying requirements. Moreover, our results suggest that applying requirements dimensions might reduce the risk of incomplete requirements.

Study Design. We were able to improve the comprehensibility of our experiment as a result of the feedback we received. Despite our efforts, we were not able to improve the other factors. Nevertheless, since we were able to obtain reasonable results, we consider the experiment design to be acceptable. Accordingly, we used the second experiment for the following study, as reported in Chapter 7. Performing our analysis approach, we noticed that the ANOVA is appropriate when examining two factors simultaneously. In order to ensure that the results will not change when we consider more than two factors, we changed the statistical analysis procedure for the following study. In addition, we learned from Exploratory Experiment II that we controlled too many individual traits, which made the analysis unnecessarily complicated and with little added value. We also needed to adjust the response options to a larger scale as they were far too small-scaled and caused fragmented results. Furthermore, we realized that there are an infinite number of individual traits that we cannot possibly take into account. Therefore, we sought to include a within-subjects design experiment to observe a sample whose individual traits are irrelevant to the study.

6.6.2 Threats to Validity

According to the guideline of Wohlin et al. [217], we discuss the threats to the internal, conclusion, external, and construct validity.

Internal Validity. For both exploratory experiments, each participant received a randomized dimension order during their tasks to reduce the *practice effect* [136]. We did not offer compensation for Exploratory Experiment I. By using convenient sampling and involving acquaintances, perhaps the so-called *demand effect* [153] could have occurred. However, since we assure the participants that their answers will be anonymized, we can assume that the participants answered truthfully. For Exploratory Experiment II, we offered participants the chance to enter a raffle for vouchers as an incentive.

Conclusion Validity. Comprehensibility was fairly neutral in Exploratory Experiment I. We could improve it for Exploratory Experiment II. Since the participants' answers were reasonable, we can assume that the descriptions and questions were sufficient. Furthermore, the participants were not distributed equally according to their individual traits. For example, in Exploratory Experiment I, regarding the factor B_9 (familiarity with the dimensions functional and non-functional requirements) we had in our control group A, one person who was unfamiliar with the dimensions *functional* and *non-functional*, while three were. Due to the small number of participants, we expected this distribution and interpreted the results as indications rather than facts. Using a two-way ANOVA, we examined our individual factors in more detail. However, we were only able to examine two factors at a time. Thus, we cannot exclude the possibility that other interaction effects would have occurred if we had considered more than two factors simultaneously. Furthermore, we must consider that a normal distribution is recommended, yet ANOVA is robust even if a normal distribution is not present [190].

External Validity. We intended to use the exploratory experiments to prepare for the following study in Chapter 7. Accordingly, the number of participants is relatively small, and the results cannot be generalized. In order to increase the external validity, we allowed a rather heterogeneous sample and did not make any restrictions regarding the professional or educational background. Furthermore, all our participants conducted our exploratory experiments in German. A reproduction of the results is probably only possible in German.

Construct Validity. Since we conducted the experiments online, we could not ensure equal external conditions. In Exploratory Experiment I, we did not yet address these circumstances. Only in Exploratory Experiment II, we asked the participants to reduce external influences and not to use any assistance. Only if the participants agreed to these conditions, we allowed them to start the experiment. Therefore, we can assume that external influences did not affect the results of Exploratory Experiment II. We also found that the evaluation scales were too detailed, so we had to aggregate the responses to identify trends. This aggregation would have been made differently by other researchers. Furthermore, we included only a limited number of individual factors for both studies. In Exploratory Experiment I, we examined eight individual factors. For Exploratory Experiment II, we increased the number to 15. However, our analysis found that in many cases, the individual factors tended to have no or negligible influence on the results. Accordingly, we decided to keep the number of individual factors to a minimum so that the duration of the following study would not be too long, and we could reduce the risk of too many study dropouts.

6.7 Related Works

Numerous studies have examined how we can extract and classify existing requirements automatically using natural language attributes [133], [188], [123].

However, there is less research concerning how natural language impacts our ability to identify and perceive requirements. Nevertheless, this observation does not mean that researchers have not pointed out the gap. According to Venkataraman and Durg [201], future studies between academia and industry should address how language affects software engineers' emotions, thoughts, and performance. The authors also refer to the linguistic relativity hypothesis.

Based on Sutton [201], our language causes us to exist in our own worlds, which overlap to some extent. The author states that these linguistic worlds greatly influence our ability to communicate and understand requirements.

In their experiments, Lauenroth et al. [115] observed that words influence how we evaluate requirements. They report that their subjects recognized the same statements as a requirement or not, depending on the taxonomy they provided them.

Giachetti et al. [201] have presented a similar theory concerning the influence of external stimuli. They propose studies on whether model languages influence engineers' perceptions and thoughts about a system and its requirements.

In our research, we could not identify any published research investigating the influence of requirements dimensions on identifying requirements. We could identify the closest similar concepts in the field of creativity research. For example, Burnay et al. [30] did a study on creativity triggers. They examined whether external stimuli could encourage stakeholders to think more diversely. The authors reported that an external trigger "sets a structure and focus for the elicitation; it applies to any development approaches (e.g., it can integrate in agile or waterfall methodologies.); it can be integrated into existing elicitation methodologies to improve their effectiveness."

6.8 Summary

We conducted two online experiments to explore the impact of applying requirements dimensions on requirements identification. To this end, we presented the examined sets of requirements dimensions $Dim_{FR\text{-}NFR}$, Dim_{ISO}, and Dim_{SUST}. We described our adaptations from the first experiment to the second experiment regarding our research design. We defined variables and hypotheses and described our data preparation and analysis approach. Our findings partially confirm the linguistic relativity hypothesis. Using the requirements dimensions Dim_{SUST} and Dim_{ISO} led to more identified requirements, but not with $Dim_{FR\text{-}NFR}$. In both exploratory experiments, applying sustainability dimensions significantly influenced identifying economic- and social-related requirements. Additionally, we observed a significant result regarding the technical dimension in the second exploratory experiment. Most of our controlled individual traits had no significant effect. Despite this, we cannot exclude that the trait of having professional experience in requirements engineering might have a significant impact on the ability to identify requirements. In contrast, we observed that inexperienced participants in requirements identification identified more requirements in general when they applied Dim_{SUST}. However, we need to consider that these experiments are only exploratory, and the number of participants was fairly small. Furthermore, we reported our learnings about the experiment design. Our related works also suggest that external stimuli can broaden the perception and thoughts of people.

7 Guiding Requirements Engineering Through Sustainability Dimensions

Publication. This chapter is partly based on our paper *The Role of Linguistic Relativity on the Identification of Sustainability Requirements: An Empirical Study* [162]. It focuses on examining the impact of different sets of requirements dimensions on the identification process. My contributions to this study comprise the research design, providing the objects of research, conducting the experiments, the data preparation, and analysis, discussion of our findings, as well as leading the writing of the paper.

Contribution. The study revealed that our examined requirements dimensions could influence requirements identification. We observed that our participants identified more requirements with the dimensions than without dimensions. Accordingly, this observation suggests that using requirements dimensions could mitigate the problem of incomplete requirements. This observation also applies to sustainability-related requirements. However, we could observe only in some cases that using dimension sets that explicitly include sustainability dimensions leads to more sustainable dimensions than sets that did not explicitly include sustainability dimensions.

7.1 Motivation

In Chapter 6, we explored how dimensions can influence requirements identification. Moreover, we learned a lot about our experiment design and adapted it as follows:

We conducted two online experiments based on the design of Exploratory Experiment II, as described in Subsection 0. However, we decided to conduct Experiment I as a between-subjects experiment and Experiment II as a within-subjects experiment. According to Kitchenham [96], we designed two different experimental settings to prevent confirmatory results. Our objective was to counteract the weaknesses of each design with their strengths to enhance the validity of our results both internally and externally.

Each experiment involved the participants identifying requirements for a grocery shopping app according to a set of requirements dimensions or without dimensions. We described the dimensions in Section 6.2.

We reduced the query of individual traits to eight factors. In the exploratory study, we sought to explore which possible individual traits might have influence the dependent variables, whereas, in this study, we were primarily interested in studying the influence of our dimensions. Our study controlled the individual traits to determine whether we could find further explanations for the results. Consequently, we also changed our statistical analysis approach. We used a multiple regression analysis to simultaneously consider more individual traits, as Creswell and Creswell recommended [45]. In conducting this analysis, we investigated whether applying requirements dimensions result in more requirements in general and more sustainability-related requirements. We define a sustainability-related requirement as a requirement that we can match to at least one of the sustainability dimensions of the Karlskrona manifesto [13]. Furthermore, we aimed to examine whether including sustainability dimensions led to more identified sustainability-related requirements. To this end, we assessed an inter-coder agreement based on Cohens' Kappa for each investigated dimension [138].

In this chapter, we describe our research design in Section 7.2 and report our results in Section 7.3. We discuss our findings in Section 7.4 and related work in Section 7.5. Finally, we summarize the chapter in Section 7.6.

7.2 Research Design

In this section, we present our research questions in Subsection 7.2.1. and variables and hypotheses in Subsection 7.2.2. We describe our data preparation procedure in Subsection 7.2.3. and our analysis approach in Subsection 7.2.4. We outline the experiment settings and samples of our Experiment I in Subsection 7.2.5 and Experiment II in Subsection 7.2.6, respectively.

7.2.1 Research Questions

First, we investigated whether the dimensions influence the identification of requirements in general. Second, we examined whether they influence the identification of sustainability-related requirements. For the examination of the influence of requirements dimensions, we based our study on the following research questions:

RQ1. What impact do sets of requirements dimensions have on the number of identified requirements in general?

RQ2. What impact do sets of requirements dimensions have on the number of identified sustainability-related requirements?

7.2.2 Variables and Hypotheses

We considered nine independent variables $x_1 - x_9$, as listed in Table 31. We used the independent variable x_1 (set of requirements dimensions) in both experiments. For Experiment I, we considered the individual factors $x_2 - x_9$. We controlled these independent variables with a questionnaire. Since the participants of Experiment II went through all treatment levels (Dim_{NO}, Dim_{ISO}, Dim_{SUST}) we did not need to consider the individual factors. To answer our RQ1, we calculated the number of identified requirements in general as our dependent variable Υ_{All} (number of all requirements). For RQ2, we calculated the number of identified requirements according to the five dimensions of the Karlskrona Manifesto [13]. To this end, we defined Υ_{ECOL}, Υ_{ECON}, Υ_{SOC}, Υ_{IND}, and Υ_{TEC}, as our dependent variables. Since we focused on whether sets of requirements dimensions have an influence on our dependent variables, we defined the following hypotheses. We rejected the null hypotheses if the p-value was higher than 0.05.

Hypotheses

$H_0: \beta_1 = 0$

$H_1: \beta_1 \neq 0$

β_1: coefficient of independent variable x_1 (set of requirements dimensions)

Table 31: Independent (IV) and dependent variables (DV).

TYPE	VARIABLES	VALUES
IV	x_1: Set of requirements dimensions	Dim_{NO}, Dim_{ISO}, Dim_{SUST}
	x_2: Age	Integer
	x_3: Gender	Female, male, diverse
	x_4: Occupation	Student, professional, other
	x_5: IT background	Yes, no
	x_6: IT working experience	Ordinal 1-5
	x_7: Personal importance of sustainability	Ordinal 1-5
	x_8: Familiarity with ISO 25010 (software quality)	Ordinal 1-5
	x_9: Usage of a grocery shopping app	Yes, no, no information
DV	Υ_{All}: Number of all requirements	Integer
	Υ_{ECOL}: Number of ecological requirements	Integer
	Υ_{ECON}: Number of economic requirements	Integer
	Υ_{SOC}: Number of social requirements	Integer
	Υ_{IND}: Number of individual requirements	Integer
	Υ_{TEC}: Number of technical requirements	Integer

7.2.3 Data Preparation

We conducted the same procedure outlined in our exploratory study in Chapter 6. We described the data preparation in detail in Subsection 6.3.3. We first removed the incomprehensible requirements. We determined how many requirements in general each group identified. We coded the requirements according to our coding guideline to determine the number of sustainability-related requirements. We followed a single-blind approach to code the identified requirements of the participants. The two coders did not know the origin of each requirement. The two coders coded the requirements independently of each other.

7.2.4 Analysis

We used both descriptive and inferential methods to evaluate the results. Regarding the descriptive statistics, we calculated the number of valid requirements for each treatment. By preparing the data, we calculated the number of requirements related to one of the five sustainability dimensions [13].

Following Creswell and Creswell [45], we applied a multiple regression analysis to consider the multitude of our independent variables, which we list in Table 31. The regression model of each dependent variable for Experiment I is summarized in Model 3. In our within-subjects experiment, the participants were exposed to all levels of the treatment to avoid individual differences distorting the results. Participants served as their baselines. Therefore, it was not necessary to consider the independent variables $x_2 - x_9$ in our analysis. Hence, we based our Experiment II on a simple regression model, as summarized in Model 4. We assessed the inter-coder agreement based on Cohens' Kappa for each of the investigated dimensions [138].

Model 3:

$$Y_l = \beta_0 + \beta_1 * x_{1l} + \ldots + \beta_9 * x_{9l} + \varepsilon_l \qquad (l = 1, \ldots, n)$$

Model 4:

$$Y_l = \beta_0 + \beta_1 * x_{1l} + \varepsilon_l \qquad (l = 1, \ldots, n)$$

Y_l: value of the dependent variable Υ_t of the l-th subject
$x_{1l}, \ldots, x_{9l}$: independent variables, where "x_1: Set of requirements dimensions"
β_0: intercept
$\beta_1, \ldots, \beta_9$: coefficient of the respective independent variable
ε_l: residual error l-th subject
n: number of participants
Υ_t: dependent variable varies according to Table 31, t: number of dependent variable

7.2.5 Experiment I

Experiment Settings

Our online experiment design is based on the guidelines of Wohlin et al. [217] and Reips and Krantz [170]. The participants could perform the experiment in English or German. We used the tool LimeSurvey to create the experiment. The completion time required about 30 minutes. We included the English version of the questionnaire in Appendix B. We organized the experiment in the following five parts, as shown in Table 32.

1. **Introduction.** We explained the study purpose, the participation conditions, the estimated completion time, the study procedure, and the data protection guidelines. Furthermore, we asked the participants not to consult additional help and to prevent any distractions. Additionally, we informed the participants about an opportunity to take part in a raffle after completing the study. To avoid influencing the participants, we made sure that we did not mention the topic of sustainability and software quality throughout the experiment. The participants had to give explicit consent to be forwarded to the study.
2. **Questionnaire I.** After their approval, the participants started with warm-up questions regarding their software development experiences. We intended to control the independent variables $x_4 - x_6$ with these questions.
3. **Assignment.** Without their knowledge, we randomly assigned the participants to one of the three groups, A, B, or C, as shown in Table 33. We introduced the participants to the trial and primary task according to their experimental groups. We informed the participants that the tasks would be time-boxed: two minutes for the trial task and eighteen minutes for the primary task. Additionally, we presented to group B the set of requirements dimensions Dim_{ISO}, as shown in Table 16, and to group C the dimensions of Dim_{SUST} according to Table 17. After the participants confirmed that they had read the instructions, we forwarded them to the trial task. Using a trial task, the participants learned how the primary task is set. To this end, we asked the participants to identify as many requirements as possible for a fitness app in keywords. Depending on the experimental group, the participants were given one random dimension from their assigned set or none if they belonged to the control group. Once the participants completed the trial task, they continued with the primary task. We sequentially showed the participants of group B and group C their assigned dimensions in random order. For each sequence, the participants of group B got two minutes and fifteen seconds, and group C received two minutes for each sequence. Figure 11 shows an example of an assignment.
4. **Questionnaire II.** In order to measure the independent variables $x_2, x_3, x_7 - x_9$, i.e., the participants' demographic information, we asked them to fill in another questionnaire. Before submitting their answers, we provided the participants with an opportunity to leave comments.

5. **Conclusion.** After completing the study, we clarified to the participants that they were part of an experiment, provided a summary of our research endeavor, and thanked them for their participation.

Table 32: Experiment I – between-subjects design procedure

PART	CONTENT	TIME LIMIT
1. Introduction	Information about process and participation conditions	-
2. Questionnaire I	Experience with software development	-
3. Assignment Dim_{NO}, Dim_{ISO}, or Dim_{SUST}	Explanation of dimensions	-
	Trial task (one random dimension of the assigned set)	2 min.
	Main task (all dimensions randomized of the assigned set)	18 min.
4. Questionnaire II	Demographics	-
5. Conclusion	Summary	-

Sample

In order to acquire participants, we approached 18 industry partners and used various mailing lists of our four local universities. Overall, a total of 70 people completed the study. With the completion of the study, the participants also had the opportunity to take part in a raffle for a commercial voucher as an incentive. There were 42 female, 28 male, and no diverse participants. Participants' average age was 30 years, ranging from 18 to 46 years. 30 participants were students, 40 professionals, and no other occupation was declared. 42 participants had no IT background, and 28 had. 17 participants stated that they had at least one year of working experience in software development. Regarding the importance of sustainability in their daily life, the participants reported an average score of four out of five points. 24 participants were familiar with the software quality *ISO Standard 25010:2011*, while 46 had no or little knowledge. Only 25 participants used a grocery shopping app, whereas 35 did not.

Table 33: Experiment I – groups

GROUP	NO. PARTICIPANTS	DIMENSIONS SET
A	28	Dim_{NO}
B	21	Dim_{ISO}
C	21	Dim_{SUST}

7.2.6 Experiment II

Experiment Settings

Experiment II was structured in the same way as Experiment I. We only adjusted the assignment part, as presented in Table 34, since every participant had to go through all three treatment levels.

3. **Assignment.** The tasks were the same as in Experiment I, described in Subsection 7.2.5. Each assignment (1-3) included an explanation, a trial task, and a primary task. The trial task again asked the participants to identify requirements for a fitness app, and the primary task asked the participants to identify requirements for a grocery shopping app. All participants applied both sets of requirements dimensions and no dimensions to identify requirements. As shown in Table 35, we formed six different groups with permuted orders of the treatments to reduce the *practice effect* [136]. We randomly assigned the participants to one of the groups. The completion time required about 80 minutes.

Table 34: Experiment II – within-subjects design procedure

PART	CONTENT	TIME LIMIT
1. Introduction	Same as experiment I	-
2. Questionnaire I	Same as experiment I	-
3. Assignments: Assignment 1, Assignment 2, Assignment 3 (based on Table 35)	Explanation of dimensions	-
	Trial task (one random dimension of the assigned set)	2 min.
	Main task (all dimensions randomized of the assigned set)	3 x 18 min.
4. Questionnaire II	Same as experiment I	-
5. Conclusion	Same as experiment I	-

Sample

We recruited 31 students who studied human computer interaction at our affiliated university. For their participation, the students received credits needed to complete their study program. The average age of the student was 23 years. Eight male, 23 female, and no diverse students participated in Experiment II. We randomly assigned each student to one of the groups, as presented in Table 35. The students reported an average score of four out of five points regarding the importance of sustainability in their daily life. 22 students were familiar with the software quality *ISO Standard 25010:2011*, while nine were not. Seven students used a grocery shopping app, compared to 24 who stated that they did not.

Table 35: Experiment II – groups and their respective set of requirements dimensions

GROUP	N	ASSIGNMENT 1	ASSIGNMENT 2	ASSIGNMENT 3
A	**5**	Dim_{NO}	Dim_{ISO}	Dim_{SUST}
B	**5**	Dim_{NO}	Dim_{SUST}	Dim_{ISO}
C	**5**	Dim_{ISO}	Dim_{NO}	Dim_{SUST}
D	**5**	Dim_{ISO}	Dim_{SUST}	Dim_{NO}
E	**6**	Dim_{SUST}	Dim_{NO}	Dim_{ISO}
F	**5**	Dim_{SUST}	Dim_{ISO}	Dim_{NO}

7.3 Results

In the following section, we present the descriptive and inferential statistics of Experiment I in Subsection 7.3.1 and of Experiment II in Subsection 7.3.2.

7.3.1 Experiment I

Descriptive Statistics

In total, Experiment I resulted in 1,076 requirements across all groups. Initially, we started with 1,125 requirements, but we had to sort out 49 requirements due to their incomprehensibility. The two coders had 116 mismatches. The intercoder agreement amounted to 0.83 for the ecological dimension, 0.75 for the social and economic each, and 0.43 for the individual and 0.53 for the technical, respectively.

As presented in Figure 20, the control group A (Dim_{NO}) identified 301 requirements, group B (Dim_{ISO}) 356 requirements, and group C (Dim_{SUST}) came up with 419 requirements. Regarding the average number of requirements in general group A (Dim_{NO}) identified 10.75 requirements, compared to 16.95 requirements in group B (Dim_{ISO}) and 19.95 in group C (Dim_{SUST}), as presented in Table 36. We observed that a participant without a set of requirements identified fewer requirements on average than a participant with a set. Furthermore, group B (Dim_{ISO}) identified more technical requirements with an average number of 4.67 per participant than the other groups. However, group C participants (Dim_{SUST}) identified more requirements regarding the dependent variables Υ_{ECOL} with an average number of 2.43 requirements, Υ_{ECON} with 1.67, Υ_{SOC} with 1.76, and Υ_{IND} with 2.19 than the participants of the other groups. Regarding Υ_{ECON} requirements, group A (Dim_{NO}) participants identified a higher average number of 1.29 requirements per person than group B (Dim_{ISO}) with 0.43.

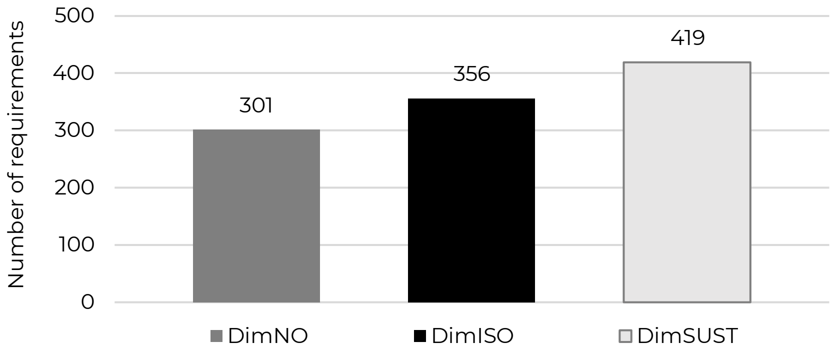


Figure 20: Experiment I – total number of requirements in general of each group

Table 36: Experiment I – descriptive statistics

	IV														
	DIM_{NO}					DIM_{ISO}					DIM_{SUST}				
DV	min	max	μ	$\tilde{x}$	σ	min	max	μ	$\tilde{x}$	σ	min	max	μ	$\tilde{x}$	σ
Υ_{ALL}	4	26	10.75	10	4.8	3	31	16.95	17	8.51	7	41	19.95	21	9.17
Υ_{ECOL}	0	7	0.32	0	0.61	0	3	0.71	1	0.84	0	5	2.43	2	1.39
Υ_{ECON}	0	3	1.29	1	1.65	0	2	0.43	0	0.67	0	4	1.67	2	0.91
Υ_{SOC}	0	5	0.86	1	0.89	0	4	0.81	0	1.16	0	5	1.76	1	1.26
Υ_{IND}	0	5	1.21	1	1.28	0	4	1.14	1	1.23	0	4	2.19	3	1.36
Υ_{TEC}	0	5	0.86	0	1.32	0	13	4.67	5	3.39	0	5	1.90	2	1.51

Inferential Statistics

We summarized the inferential statistics in Table 37. Regarding our dependent variable Υ_{All} , we can reject the null hypothesis H_0. The participants we provided with a set of requirements dimensions identified significantly more requirements per person than those without a set. However, there was no significant difference between group B (*Dim_{ISO}*) and group C (*Dim_{SUST}*).

Furthermore, group B (*Dim_{ISO}*) participants identified significantly more technical requirements compared to group C (*Dim_{SUST}*) and group A (*Dim_{NO}*). Thus, we can reject the null hypothesis H_0 relating to the dependent variable Υ_{TEC}. However, for the other dependent variables, the group B (*Dim_{ISO}*) participants could not significantly identify more requirements than their peers in the other groups.

Group C (*Dim_{SUST}*) participants identified significantly more requirements in terms of the dependent variables Υ_{ECOL}, Υ_{SOC}, and Υ_{IND} compared to the group A (*Dim_{NO}*) participants. Hence, we can reject the null hypothesis H_0 regarding these dependent variables. However, group C (*Dim_{SUST}*) participants identified significantly more requirements only regarding the dependent variables Υ_{ECOL} and Υ_{ECON}, compared to the group B (*Dim_{ISO}*) participants. Group A (*Dim_{NO}*) participants achieved no significant results. Neither in comparison with group B (*Dim_{ISO}*) or group C (*Dim_{SUST}*).

Considering the individual traits, we observed that regarding the dependent variable Υ_{All} the independent variable x_7 (personal importance of sustainability) yielded a significant result as reported in Table 38. According to the adjusted R^2 the set of the chosen predictors can explain the results only with a value of 0.302. Regarding the dependent variable Υ_{ECOL} no other predictor had a significant influence, as shown in Table 39. According to the adjusted R^2 the set of the chosen predictors can explain the results only with a value of 0.448. This also applies to Υ_{ECON}, as we presented in Table 40, Υ_{SOC}, as shown in Table 41, Υ_{IND}, as summarized in Table 42. According to the adjusted R^2 the set of the chosen predictors can explain the results only with a value of 0.091 for Υ_{ECON} and Υ_{SOC}, and 0.198 for Υ_{IND}. Regarding the dependent variable Υ_{TEC} the predictors x_7 (personal importance of sustainability) and x_8 (familiarity with ISO 25010 (software quality)) had a significant influence, as shown in Table 43. However, the significance of x_8 had a negative influence. According to the adjusted R^2 the set of the chosen predictors can explain the results only with a value of 0.489.

Table 37: Experiment I – inferential statistics

DV	IV												
	DIM_{NO}				DIM_{NO}				DIM_{ISO}				
	Predictor	**B**	**p**		**Predictor**	**B**	**p**		**Predictor**	**B**	**p**		**Adj R^2**
Υ_{All}	DIM_{SUST}	**8.113**	**0.001**	▲	DIM_{ISO}	**7.613**	**0.001**	▲	DIM_{SUST}	0.500	0.830	□	0.310
Υ_{ECOL}	DIM_{SUST}	**2.120**	**0.001**	▲	DIM_{ISO}	0.418	0.174	□	DIM_{SUST}	**1.702**	**0.001**	▲	0.448
Υ_{ECON}	DIM_{SUST}	0.198	0.609	□	DIM_{ISO}	-0.754	0.054	□	DIM_{SUST}	**0.952**	**0.033**	▲	0.091
Υ_{SOC}	DIM_{SUST}	**0.758**	**0.031**	▲	DIM_{ISO}	0.100	0.771	□	DIM_{SUST}	0.658	0.096	□	0.091
Υ_{IND}	DIM_{SUST}	**0.939**	**0.016**	▲	DIM_{ISO}	0.240	0.526	□	DIM_{SUST}	0.699	0.108	□	0.198
Υ_{TEC}	DIM_{SUST}	0.483	0.423	□	DIM_{ISO}	**4.187**	**0.001**	▲	DIM_{SUST}	-3.703	0.001	▽	0.489

▲ positive significance ▽ negative significance □ no significance

Table 38: Experiment I – regression Υ_{All}

PREDICTORS	B	P
(Constant)	0.719	0.893
Age	-0.021	0.894
Gender	0.491	0.807
Occupation	2.313	0.317
IT background	2.481	0.313
IT working experience	1.593	0.137
Sustainability important	**3.068**	**0.004**
Knowing ISO quality	-1.188	0.094
Usage grocery shop. app	-2.292	0.181
Dim_{ISO}	**7.613**	**0.001**
Dim_{SUST}	**8.113**	**0.001**
Adjusted R^2		0.302

Table 39: Experiment I – regression Υ_{ECOL}

PREDICTORS	B	P
(Constant)	-0.412	0.585
Age	-0.004	0.870
Gender	-0.400	0.160
Occupation	0.157	0.628
IT background	0.008	0.982
IT working experience	0.160	0.286
Sustainability important	0.231	0.111
Knowing ISO quality	0.078	0.431
Usage grocery shop. app	-0.001	0.995
Dim_{ISO}	0.418	0.174
Dim_{SUST}	**2.120**	**0.001**
Adjusted R^2		0.448

Table 40: Experiment I – regression Υ_{ECON}

PREDICTORS	B	P
(Constant)	0.762	0.424
Age	0.003	0.920
Gender	0.417	0.245
Occupation	0.393	0.336
IT background	-0.476	0.274
IT working experience	0.162	0.390
Sustainability important	0.002	0.990
Knowing ISO quality	0.111	0.370
Usage grocery shop. app	0.014	0.964
Dim_{ISO}	**-0.754**	**0.054**
Dim_{SUST}	0.198	0.609
Adjusted R^2		0.091

Table 41: Experiment I – regression Υ_{SOC}

PREDICTORS	B	P
(Constant)	0.621	0.465
Age	-0.023	0.345
Gender	0.130	0.683
Occupation	0.010	0.977
IT background	-0.247	0.523
IT working experience	0.168	0.318
Sustainability important	0.325	0.048
Knowing ISO quality	-0.092	0.408
Usage grocery shop. app	-0.012	0.964
Dim_{ISO}	0.100	0.771
Dim_{SUST}	**0.758**	**0.031**
Adjusted R^2		0.091

Table 42: Experiment I – regression Υ_{IND}

PREDICTORS	B	P
(Constant)	0.093	0.921
Age	0.032	0.241
Gender	0.326	0.353
Occupation	0.404	0.314
IT background	-0.117	0.783
IT working experience	-0.009	0.960
Sustainability important	0.066	0.709
Knowing ISO quality	-0.233	0.060
Usage grocery shop. app	-0.392	0.189
Dim_{ISO}	0.240	0.526
Dim_{SUST}	**0.939**	**0.016**
Adjusted R^2		0.198

Table 43: Experiment I – regression Υ_{TEC}

PREDICTORS	B	P
(Constant)	-1.028	0.488
Age	-0.047	0.278
Gender	0.871	0.120
Occupation	0.299	0.638
IT background	0.502	0.458
IT working experience	0.555	0.061
Sustainability important	**0.978**	**0.001**
Knowing ISO quality	**-0.390**	**0.047**
Usage grocery shop. app	-0.621	0.189
Dim_{ISO}	**4.187**	**0.001**
Dim_{SUST}	0.483	0.423
Adjusted R^2		0.489

7.3.2 Experiment II

Descriptive Statistics

In total, Experiment II resulted in 1,811 requirements across all groups. Initially, we started with 1,925 requirements, but we had to sort out 114 requirements due to their incomprehensibility. The two coders had 111 mismatches. The intercoder agreement amounted to 0.74 for the ecological dimension, 0.68 for the social and economic dimensions, and 0.5 for the individual and technical dimensions, respectively.

The participants identified 521 requirements without a set of dimensions, 667 with Dim_{ISO}, and 623 requirements with Dim_{SUST}, as shown in Figure 21. Regarding the average number of requirements in general Υ_{All} the participants identified 21.45 requirements with set Dim_{ISO}, compared to 20.10 requirements with Dim_{SUST}, and 16.81 without a set, as presented in Table 44. We observed that the participants identified more requirements on average with a set of requirements dimensions than without one. Furthermore, the participants identified more technical requirements with an average of 4.77 requirements per person when they applied set Dim_{ISO} compared to the other treatments. However, the participants identified with Dim_{SUST} more requirements on average regarding the dependent variables Υ_{ECOL} with 2.13 requirements, Υ_{ECON} with 1.95, Υ_{SOC} with 1.97, and Υ_{IND} with 2.23 requirements than with the other treatments. Without any dimensions, the participants could identify more requirements on average regarding Υ_{ECOL} with 1.16 requirements per person, Υ_{ECON} with 1.58 and Υ_{IND} with 1.23.

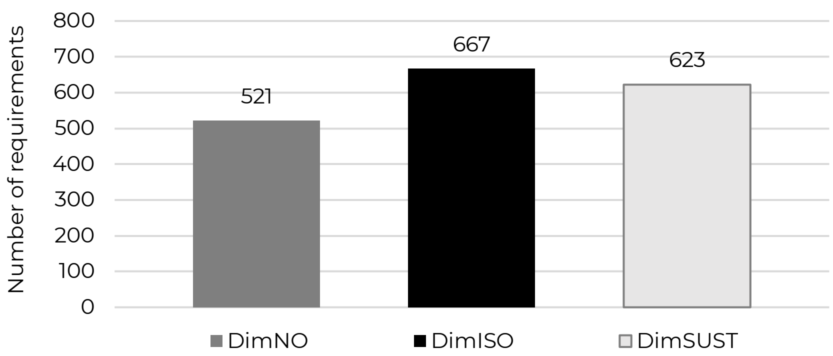


Figure 21: Experiment II – total number of requirements of each group

Table 44: Experiment II – descriptive statistics

DV	IV														
	DIM_{NO}					DIM_{ISO}					DIM_{SUST}				
	min	max	μ	$\tilde{x}$	σ	min	max	μ	$\tilde{x}$	σ	min	max	μ	$\tilde{x}$	σ
Υ_{All}	1	39	16.81	16	8.97	8	60	21.52	19	10.24	8	48	20.10	19	9.27
Υ_{ECOL}	0	6	1.16	1	1.36	0	4	0.97	1	1.11	0	6	2.13	2	1.38
Υ_{ECON}	0	5	1.58	1	1.5	0	4	0.74	1	0.89	0	4	1.94	2	1.15
Υ_{SOC}	0	3	1.03	1	0.87	0	5	1.10	1	1.3	0	6	1.97	2	1.19
Υ_{IND}	0	3	1.23	1	1.08	0	4	1.03	1	1.11	0	6	2.23	2	1.43
Υ_{TEC}	0	9	1.84	1	2.19	0	14	4.77	4	3.5	0	4	1.94	2	1.23

Inferential Statistics

We summarized the inferential statistics in Table 45. Regarding Υ_{All} we can again reject the null hypothesis H_0 . The participants identified significantly more requirements per person with a set of requirements dimensions than without one. However, there was no significant difference between the application of the set Dim_{ISO} and set Dim_{SUST}.

Once again, the participants identified significantly more technical requirements with the Dim_{ISO} than with the other treatments. Thus, we can reject the null hypothesis H_0 relating to the dependent variable Υ_{TEC} . However, for the other dependent variables, the participants could not significantly identify more requirements with Dim_{ISO}.

When using Dim_{SUST}, the participants identified significantly more requirements in terms of the dependent variables Υ_{ECOL}, Υ_{SOC}, and Υ_{IND} compared to the control group A (Dim_{NO}) participants. Hence, we can reject the null hypothesis H_0 regarding these dependent variables. However, when applying Dim_{SUST} the participants identified significantly more requirements regarding the dependent variables Υ_{ECOL}, Υ_{ECON}, Υ_{SOC}, and Υ_{IND} compared to when applying Dim_{ISO}.

Without any dimensions, the participants identified significantly more requirements regarding Υ_{ECON} compared to using *Dim_{ISO}*. Since the participants were exposed to all treatments, we did not need to control individual traits.

Table 45: Experiment II – inferential statistics

DV	IV												
		DIM$_{NO}$				DIM$_{NO}$				DIM$_{ISO}$			
	Predictor	B	p		Predictor	B	p		Predictor	B	p		Adj R^2
Υ_{All}	DIM$_{SUST}$	**3.290**	**0.006**	▲	DIM$_{ISO}$	**4.710**	**0.001**	▲	DIM$_{SUST}$	-1.419	0.217	□	0.783
Υ_{ECOL}	DIM$_{SUST}$	**0.968**	**0.001**	▲	DIM$_{ISO}$	-0.194	0.479	□	DIM$_{SUST}$	**1.194**	**0.001**	▲	0.397
Υ_{ECON}	DIM$_{SUST}$	0.355	0.197	□	DIM$_{ISO}$	-0.839	0.003	▽	DIM$_{SUST}$	**1.194**	**0.001**	▲	0.319
Υ_{SOC}	DIM$_{SUST}$	**0.935**	**0.001**	▲	DIM$_{ISO}$	0.065	0.803	□	DIM$_{SUST}$	**0.871**	**0.001**	▲	0.291
Υ_{IND}	DIM$_{SUST}$	**1.000**	**0.001**	▲	DIM$_{ISO}$	-0.194	0.483	□	DIM$_{SUST}$	**1.194**	**0.001**	▲	0.327
Υ_{TEC}	DIM$_{SUST}$	0.097	0.843	□	DIM$_{ISO}$	**2.935**	**0.001**	▲	DIM$_{SUST}$	-2.839	0.001	▽	0.537

▲ positive significance ▽ negative significance □ no significance

7.4 Discussion

In the following section, we discuss the implications of our results in Subsection 7.4.1 and threats to validity in Subsection 7.4.2.

7.4.1 Implications

Our results suggest that requirements dimensions can guide a requirements identification process. The experiments demonstrated that our two sets of dimensions led to more requirements than no given set. Accordingly, practitioners may be able to address their recurring problem of missing requirements [140] by applying requirements dimensions. During the project, participants might be able to identify specific requirements they would otherwise not have identified until much later.

Our results indicate that we can significantly identify more technical requirements with the set *Dim$_{ISO}$*, than with *Dim$_{SUST}$* or without any requirements dimensions. The technical focus of the set can likely explain this observation, e.g., the mention of modularity to measure maintainability or time behavior to measure performance efficiency [98]. Regarding identifying ecological-related requirements, the participants performed significantly better with the set *Dim$_{SUST}$* than with the other two treatment levels.

The results on social- and individual-related requirements are less clear than the aforementioned dimensions. In both experiments, participants achieved significantly more social- and individual-related requirements with *Dim$_{SUST}$* than without dimensions. However, if we compare *Dim$_{ISO}$* and *Dim$_{SUST}$*, we could only observe a significant difference in Experiment II. This observation is perhaps due to the fact that the differentiation of those requirements is less precise, which could also explain the relatively moderate coders' agreement regarding these dimensions.

The control group's results imply that it makes no difference whether participants receive requirements dimensions or the set Dim_{SUST} to identify economic-related requirements. Only in comparison with set Dim_{ISO} participants identified significantly fewer economic-related requirements. The topic of economics seems to be ubiquitous, as no further assistance is needed to identify economic-related requirements. Based on the comparison with set Dim_{ISO}, economic aspects may be pushed into the background when participants focus on other aspects.

In summary, we can say that providing Dim_{ISO} or Dim_{SUST}, can increase the chance that stakeholders identify more requirements and consider sustainability-related requirements. However, the examined sets might guide requirements identification with a specific emphasis. Using Dim_{ISO} will likely guide the stakeholders to consider a broad range of technical-related requirements. While applying Dim_{SUST}, we might raise the chance of considering ecological- and economic-related requirements. Regarding the dimensions social and individual, both sets might be appropriate.

Future work needs to examine what meaning these preliminary results may have for the individuals involved in the design of software products. The results provide us with at least two possible interpretations. One possibility would be to explore how external stimuli can be optimized further so that people can identify requirements more efficiently. As another possibility, we might interpret it as a signal that we should do something to enable individuals to formulate their own requirements or ideas without or with as little external stimulus as possible. However, the second option may make people more independent of our produced tools and systems. It is uncertain whether one discipline can pursue the second option alone since it may require this discipline to dispense itself partially.

7.4.2 Threats to Validity

According to the guideline of Wohlin et al. [217], we discuss the threats to the internal, conclusion, external, and construct validity.

Internal Validity. We randomized the order of the requirements dimensions to reduce the effects [136] related to the sequence of dimensions for both experiments. In addition, we permuted the order of the treatments in Experiment II. We compensated the participants of Experiment I with an opportunity to win a commercial voucher in a raffle after completing the study. Since the participants of Experiment II were students, they received credit points for a completed study. Participants also had to confirm prior to the study that they had not yet participated in the study. Due to the length of the completion time and no additional credit points or higher chances of winning through repeated participation, we can assume that each participant has only taken part in the study once.

Conclusion Validity. As described in Chapter 6, we ran an exploratory study before conducting the experiments. We used the gathered feedback to examine whether the participants correctly comprehended the assignments and questions. This evaluation enabled us to improve the design of this study. In addition, we could test our coding guideline by analyzing the data from the exploratory study. We applied researcher triangulation. Even though we carried out the qualitative assessment in pairs, we cannot exclude that other researchers would have arrived at a different coding for the requirements identified with Dim_{ISO} or no dimensions. The intermediate results of the kappa test showed that an interpretation is not so clear. To enhance conclusion validity, we asked the participants to complete a trial task before they were allowed to do the primary task. Through this, we could keep participants from spending time on the primary task to learn how it works.

External Validity. Only students of the same study program participated in Experiment II. It might mitigate its external validity since we assume the participants have a similar background. However, it is common to use students as subjects in research since it is more difficult to recruit professionals for such studies, which require a particular completion time. We tried to decrease this potential threat by recruiting participants with different occupations and fields of expertise for Experiment I. Furthermore, our experiments may not be replicable because only two participants conducted the study in English and the other 68 participants in German. It might lead to different results if others conduct this study in a language other than German or English. Additionally, our samples are relatively small. Therefore, our quantitative reports should serve to derive further hypotheses than proven theories.

Construct Validity. We conducted the experiments online. Before the participants could start the experiment, they had to confirm that they would avoid distractions and not ask for help. Therefore, we can assume that the participants carried out the experiments correctly, but we cannot completely exclude any external influences. The sample sizes of 70 participants for Experiment I and 31 for Experiment II are relatively small. Thus, there is a risk that there is no normal distribution, which ideally should be given for regression to be applied. However, according to Wooldrigde [219], a regression analysis is robust enough under conditions that are not ideal. Furthermore, we included only a limited number of individual factors for Experiment I, as listed in Table 29. Thus, it is possible that other factors could have influenced the dependent variables.

7.5 Related Works

In the previous chapter, we indicated that we could not find any studies that examine whether the linguistic relativity hypothesis plays a role in identifying requirements. However, we identified studies examining how explicitly addressing sustainability stimulates new perspectives among stakeholders.

To initiate and guide discussions regarding the impact of software systems on sustainability, Duboc et al. [50] propose a question-based framework.

Lago et al. [112] present a framework for identifying sustainable software qualities and revealing potential conflicts between them to discuss trade-offs.

Seyff et al. [183] follow a similar approach. To enable negotiations focusing on the impact of requirements on the sustainability of a software system, they propose the *WinWin* negotiation model.

Alternatively, Cabot et al. [31] propose a negotiation method that visualizes the interrelationships between business-relevant goals and sustainable requirements.

Brito et al. [26] propose a way to model sustainability concepts and manage conflict situations using concern-oriented requirements. A situation such as this may arise from interactions between sustainability dimensions or interactions with other aspects of a system.

We see that addressing sustainability can play a special role in ensuring that stakeholders take sustainability into account. It has gained such importance that the separate discipline *sustainability communication* [76] has emerged. According to Godemann and Michelsen [76]: "This discipline has set itself the goal not only of providing a clear and persuasive understanding of sustainable development and of campaigning for its acceptance but above all of involving people in the process of sustainable development and motivating them to take part in it actively." Similarly, we discuss the importance of communication in promoting sustainable software design in the subsequent chapter.

7.6 Summary

We conducted two online experiments. We prepared Experiment I as a between-subjects design and Experiment II as a within-subjects design. In total, 70 participants performed Experiment I, and 31 conducted Experiment II. We defined the variables and hypotheses and described our data preparation and analysis approach. According to the linguistic relativity hypothesis, we observed that applying requirements dimensions could influence the identification process. If stakeholders intend to pursue a technical identification focus, it is beneficial to use Dim_{ISO}. If they also aim to identify ecological-related requirements, we suggest applying Dim_{SUST}. Regarding the social- and individual-related requirements, both sets could be beneficial compared to applying no dimensions. In contrast, considering economic-related requirements does not need much additional support unless stakeholders decide to apply Dim_{ISO}, which might push back economic-related requirements. Our results provide indications that stakeholders can guide the project direction early on by selecting appropriate requirements dimensions. However, we need to consider that these experiments are based on a relatively small number of participants. In summary, we can say that external stimuli like applying requirements dimensions can broaden the mind of people. In the next chapter, we will exemplify how we can use these external stimuli to create a shared view on requirements. To this end, we prepared a card-based communication approach on the dimensions of Dim_{SUST} to promote sustainable software design, as presented in Chapter 8.

8 Sustainability Poker: Impact Estimation

This chapter presents our card-based estimation and communication approach, *Sustainability Poker*. As stated in Chapter 5, among the most common issues concerning promoting sustainable software design is the lack of awareness and shared understanding. This issue could lead to communication difficulties. However, as described in Section 2.2, we need effective communication to promote sustainable software design. By enabling effective communication, we can reduce the risk of project failures and raise the chance of considering sustainability-related aspects. To this end, we extended Faltin's research [59]. Using our approach, we aimed to promote interdisciplinary teams to estimate and communicate a requirement's impact on a software product's sustainability. Our estimation approach is based on our dimensions set Dim_{SUST}, as presented in Subsection 6.2.3.

8.1 Motivation

According to our study in Chapter 5, practitioners lack awareness and knowledge about sustainable software products. Furthermore, we learned in Chapter 2 that implementing sustainable software products might require novel ideas. However, it can be of particular importance to understand and communicate the potential or consequences of novel ideas or novel requirements since novel requirements have to deal with the challenge of not being able to fall back on many experiences. Hence, stakeholders might need as many people as possible to promote sustainable software design. To this end, we prepared a card-based approach to assist interdisciplinary teams in promoting sustainable software design. We focused on interdisciplinary teams since implementing sustainable software products requires contributions from different disciplines [13]. The estimation approach is based on our dimensions set Dim_{SUST}, as described in Subsection 6.2.3. As a result of applying our approach, an interdisciplinary team can find a position concerning a particular product or a requirement regarding its impact on a software product's sustainability. To accomplish this, we prepared dimension cards to explore the requirement from various perspectives. In this process, we aim to help the team better understand each other's potential perspectives. The dimension cards should serve as a basis for the team to understand better where a possible perspective may originate.

We report the preparation of our card-based approach in the following sections. In Section 8.2, we describe the estimation procedure, and in Section 8.3, how we evaluated our procedure. We present our results in Section 8.4 and discuss our findings in Section 8.5. We conclude with related works in Section 8.6 and a chapter summary in Section 8.7.

8.2 Estimation Procedure

In the following, we present our estimation and communication approach based on the *Planning Poker* [79] method, as described in Subsection 2.2.4. Our estimation procedure consists of an individual estimation and a group estimation, which we present in Subsection 8.2.1. To this end, we have prepared artifacts that consist of our *Sustainability Cards*, a short description of a requirement and a rating scale, as described in Subsection 8.2.2.

8.2.1 Process

After conducting the estimation, the team should better understand why they wish to implement or not implement a requirement. Furthermore, the team members should think about how they would implement a requirement to achieve a sustainable solution. We suggest performing the following steps:

1. **Individual Estimation.** Each team member has the opportunity to formulate their own opinions about a given product or requirement. As part of this process, we prepared *Sustainability Cards* that contain questions. Team members can rate the impact of the requirements on a five-point scale based on the answers to these questions. Two more points were later added to the scale, as described in Subsection 8.3.1. The team members must provide a rating for each card, and we calculated an overall rating at the end. As shown in Figure 26, we summarized the results of all the team members' estimations in an overview.
2. **Group Estimation.** This overview allows team members to see where their estimates differ. Figure 26 illustrates the artifact the members can use to explain why they arrived at their estimates. As a result, team members might change their estimates and could encourage members to reconsider adjusting the requirements to achieve a sustainable software design.

8.2.2 Artifacts

We created cards based on our set Dim_{SUST}, as shown in Figure 22. For the five dimensions, *ecological, economic, social, individual*, and *technical*, we adopted the subdimensions and questions from Duboc et al. [50]. For the other four dimensions, *integrative, legal, design-aesthetic*, and *purpose*, we adopted Faltin's [59] subdimensions and questions. In Table 46, we present the nine dimensions and their subdimensions. As seen in Figure 23, we first created a preliminary version based on our interview results, as reported in Chapter 5. Our first version integrated the dimensions from the software quality *ISO standard 25010:11* and positive examples for each dimension. However, based on our pre-study, we observed that our preliminary version was far too overwhelming. For the remainder of the research, we reduced it to only our dimensions.

Figure 24 illustrates the first version of our artifact. As part of our artifact, we have provided a brief description of a selected requirement, along with a *Sustainability Card* and a rating scale. We assigned each point on the rating scale a value. For example, the statement "positive influence" has a value of +2, and the statement "negative influence" has a value of -2. As shown in Figure 27, we illustrate an example of the average value for each participant at the end of the procedure. Further iterations revealed that one estimate was insufficient to cover an entire card. Thus, in Iteration III, we offered the participants the opportunity to provide an estimate based on the subdimension, as shown in Figure 25. Participants wanted to indicate whether they had no idea or believed the requirement had both a positive and negative impact. Consequently, we expanded the rating scale. Furthermore, in our third iteration, we added the aspect of "considering third-party providers" to the technical dimension. In the last iteration, we divided the nine cards into 41 separate cards, as shown in Figure 26.

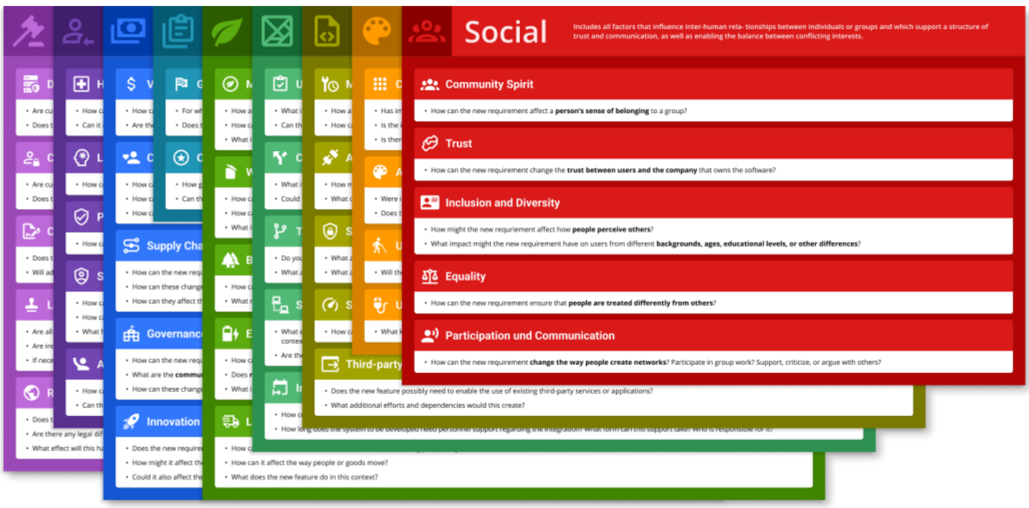


Figure 22: Overview – dimension cards and questions adopted from Faltin [59] and questions from Duboc et al. [50]

Table 46: Subdimensions and questions adopted from Duboc et al. [50] and Faltin [59]

DIMENSION	SUBDIMENSION

Ecological [50]	• Material and Resources
	  ○ How and what materials are consumed to produce and operate the software or service?
	  ○ How can it change the way its users consume materials?
	  ○ What impact does the new requirement have on the consumption of materials and resources?
	• Waste and Pollution
	  ○ How can the production or use of the software generate waste or emissions?
	  ○ How can the use promote (or affect) recycling?
	  ○ What is the impact of the new requirement on generating waste or emissions?
	• Biodiversity
	  ○ How can the software or service affect the soil, plants, or animals?
	  ○ What role does the new requirement play in this context?
	• Energy
	  ○ How can the software or service influence the need for energy generation?
	  ○ How much energy is required?
	  ○ Does renewable energy power the hardware? Is there a way to incentivize this?
	  ○ What impact does the new requirement have in this context?
	• Logistics
	  ○ How can the software or service affect the need for moving people or goods?
	  ○ How can it affect the way people or goods move?
	  ○ What does the new feature do in this context?

Economic [50]	• Value
	  ○ How can the new requirement create or destroy monetary value? For whom?
	  ○ Are there other related types of business value? For whom?
	• Customer Relationship
	  ○ How can the new requirement affect relationships between the company and its customers?
	  ○ How can it enable the co-creation or co-destruction of value?
	  ○ How can it affect the financial situation of customers and others?
	• Supply Chain
	  ○ How can the new requirement affect the company's supply chain?
	  ○ How can these changes in the supply chain affect the financial situation?
	  ○ How can they affect the financial situation of customers and others?
	• Governance
	  ○ How can the new requirement influence how decisions are made and by whom?
	  ○ What are the communication channels through which the relationships take place?
	  ○ How can these changes affect the company's financial situation, customers, and partners?
	• Innovation
	  ○ Does the new requirement promote innovation? Is it innovative in itself?
	  ○ How might it affect the financial situation?
	  ○ Could it also affect the financial situation of customers and partners?

Social [50]	• Community Spirit
	  ○ How can the new requirement affect a person's sense of belonging to a group?
	• Trust
	  ○ How can the new requirement change the trust between users and the company that owns the software?
	• Inclusion and Diversity
	  ○ How might the new requirement affect how people perceive others?
	  ○ What impact might the new requirement have on users from different backgrounds, ages, educational levels, or other differences?
	• Equity
	  ○ How can the new requirement ensure that people are treated differently from others?
	• Participation and Communication
	  ○ How can the new requirement change the way people create networks? Participate in group work? Support, criticize, or argue with others?

Individual [50]	• Health
	  o How can the new requirement improve or worsen a person's physical, mental, and/ or emotional health?
	  o Can it make a person feel good or bad – e.g., (un)appreciated, (dis)respected, (in)dependent, or coerced?
	• Lifelong Learning
	  o How can the new requirement affect people's skills?
	• Privacy
	  o How can the new requirement reveal (or help conceal) a person's identity, whereabouts, or relationships?
	• Safety
	  o How can the new feature prevent (or protect) a person from physical harm?
	  o How can it make a person feel more (or less) exposed to danger?
	  o What happens if it is used in an unintended way?
	• Agency
	  o How can the new feature enable (or prevent) a person from taking action/decision when necessary?
	  o Can those affected by the product or service understand the impact, raise concerns, or be represented by someone?

Technical [50]	• Maintenance
	  o How are the operating system and runtime environment likely to change, and what does that require of the maintainers of the software?
	  o How can the correctness of the software be affected by or affect the correctness of other systems?
	• Adaptability
	  o How could someone use the code in a different context? What can make this easier/difficult?
	  o What can make it easier/challenging for the system to adapt to new usage scenarios?
	• Security
	  o What assets created by the new requirements and controlled by the software be desirable to an attacker? E.g., financial information, whereabouts, or preferences of people, etc.
	  o What are the risks associated with these assets? What are other likely vulnerabilities of the software?
	• Scalability
	  o How can the new requirement affect the workload? What can make this easier/difficult?
	• Third-party provider integration
	  o Does the new feature possibly need to enable the use of existing third-party services or applications?
	  o What additional efforts and dependencies would this create?

Legal [59]	• Data protection
	  o Are current data protection laws known and has the software been developed accordingly?
	  o Does the new requirement disproportionately increase the complexity of the product in comparison to the existing laws?
	• Copyright
	  o Are current copyright laws known, and have they been considered?
	  o Does the new feature disproportionately increase the complexity of the product in relation to the laws?
	• Contracts
	  o Does the new feature require contract adjustments with customers?
	  o Will additional contracts have to be concluded? Will the complexity of the contracts be increased disproportionately?
	• Legal Barriers
	  o Are all licenses for third-party software or infrastructure used in the software available, up-to-date, and documented accordingly?
	  o Are industry-specific laws and regulations known that impose particular requirements on the software?
	  o If necessary, are permits, licenses, or approvals required for the business field?
	• Regional Differences
	  o Does the new requirement make the software available in other countries?
	  o Are there any legal differences in the new countries compared to previously known laws?
	  o What effect will this have on the product?

<table>
<tr><td>Integrative [59]</td><td>

Usage Context

What is the context of use in which the software is integrated? What are the needs, wishes, and goals of the users?
Can the new requirement be easily integrated into the existing context?

Change Process

What is the importance of integrating the new requirement in its context? Will the integration digitize processes? What effects could this have on the people involved?
Could the new requirement cause disruptions in processes? How can these be prevented or subsequently resolved?

Test complexity

Do you need to test the new requirement separately? What kind of testing is necessary?
What and which people are needed for this? What impact can be expected?

System integration

What existing systems are there that may affect the software being developed? Should the software integrate with other systems? Which persons have which responsibilities in the context?
Are there any special requirements for the software that should be specified separately in the contract? (e.g., particularly fast response times, downtimes. Etc.)

Introduction Process

How complex is your software? Can you assume that users will understand the software immediately? Will training be needed?
How long does the system to be developed need personnel support regarding the integration? What form can this support take? Who is responsible for it?

</td></tr>
<tr><td>Design-aesthetic [59]</td><td>

Clarity

Has important content been placed in the foreground? Can users find the content or areas they are looking for? Has content that belongs together been visualized in groups?
Is the interaction of text, icons, and images coherent?
Is there a recognition value between individual areas of the software? Do visual elements support the information architecture?

Aesthetic

Were images, fonts, shapes, and colors used according to appropriate design rules?
Does the design look contemporary? What are the current trends in interface design? Does it make sense to adapt the design accordingly?

User-Centered

Will the new requirement optimize the presentation and wording of texts for different users?

Usability

What knowledge or physical attributes are required to use the software after the new requirement is introduced, and how might this affect different types of users?

</td></tr>
<tr><td>Purpose [59]</td><td>

Goal-oriented

For what purpose will the users apply the software or service? What is the goal that the users want to achieve with the software or service?
Does the new requirement help or hinder the users from achieving their goals?

Outcome Quality

How good is the quality of the results of the software or service? Are the results effective?
Can the new requirement improve the quality of results?

</td></tr>
</table>

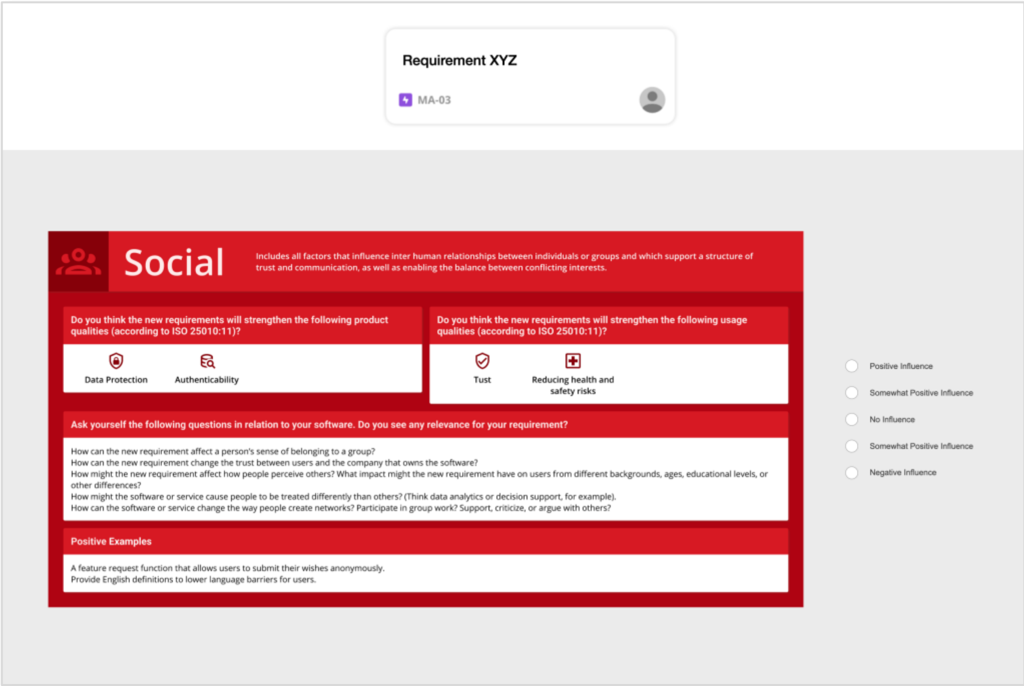


Figure 23: Pre-Study – modified representation of the dimension social based on Duboc et al. [50] and Faltin [59]

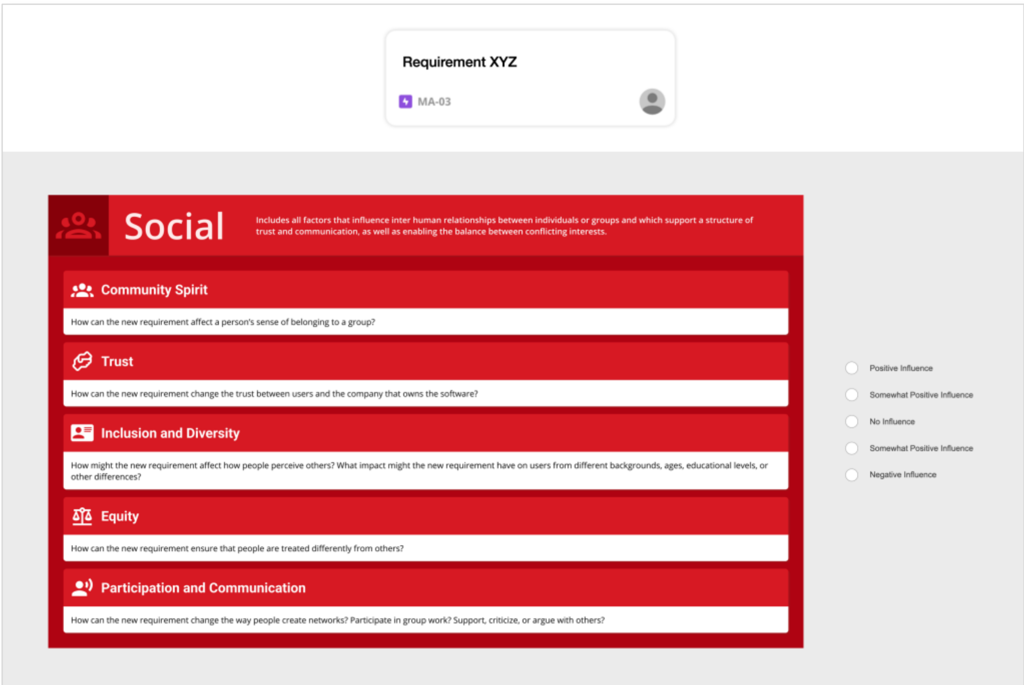


Figure 24: Iteration I & II – modified representation of the dimension social based on Duboc et al. [50] and Faltin [59]

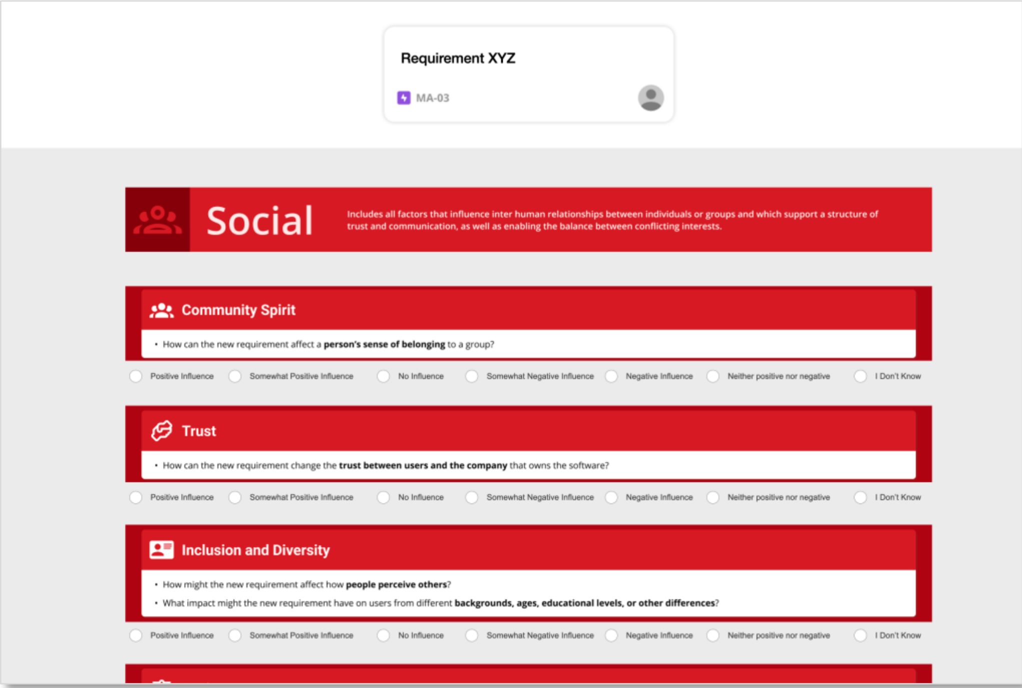


Figure 25: Iteration III – modified representation of the dimension social based on Duboc et al. [50] and Faltin [59]

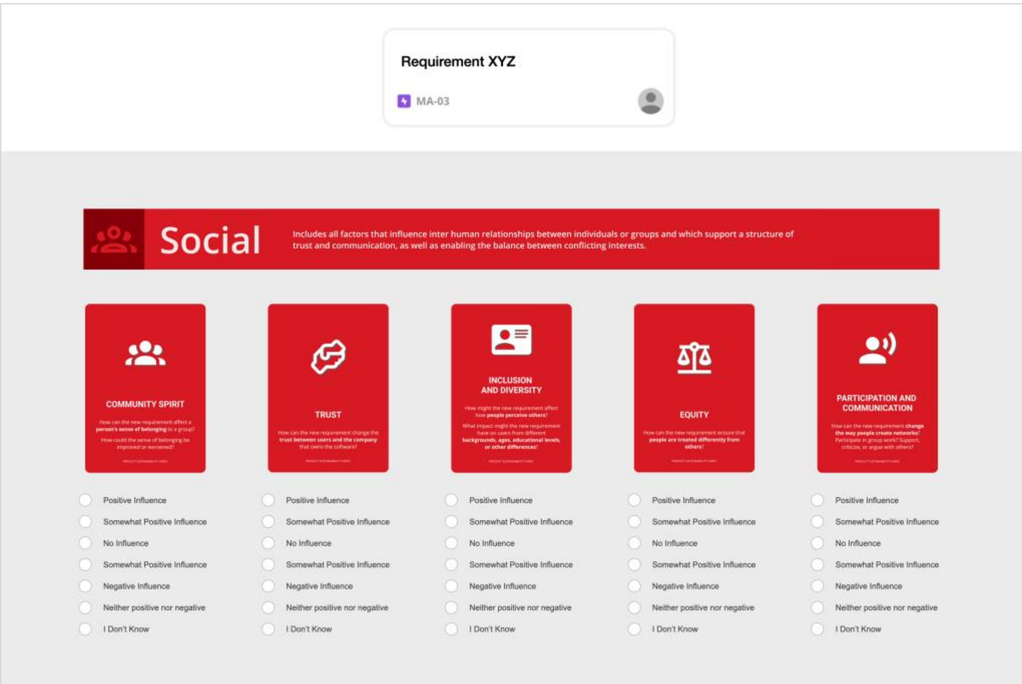


Figure 26: Iteration IV – modified representation of the dimension social based on Duboc et al. [50] and Faltin [59]

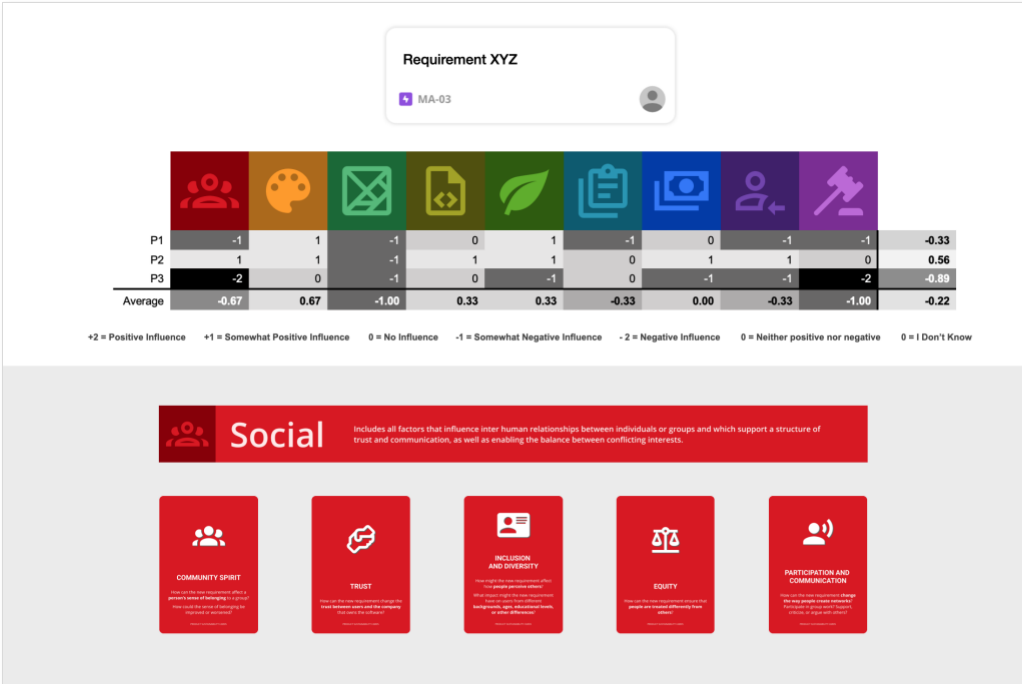


Figure 27: Example results and modified representation of the dimension social based on Duboc et al. [50] and Faltin [59]

8.3 Research Design

Using the following research design, we simulated and evaluated our estimation approach to gain suggestions for improvements. To simulate the individual estimation, we embedded it in a survey, as described in Subsection 8.3.1. We simulated the group estimation with focus group discussions in Subsection 8.3.2. We described our analysis approach in Subsection 8.3.3. We present our sample in Subsection 8.3.4 and conclude with a brief description of the pre-study in Subsection 8.3.5.

8.3.1 Individual Estimation – Survey

Following the guidelines of Leavy [118], we prepared an online survey for the *individual estimation*. We provided the survey only in German and created it with the tool LimeSurvey. The completion time was about 30-45 minutes. Five parts make up our survey. In the respective iteration, we adjusted the estimation based on the previous iteration's learnings and our research objectives. We summarized the structure of the survey in Table 47 and conducted the individual estimation as follows:

Table 47: Survey procedure

PART	CONTENT
1. Introduction	Information about process and participation conditions
2. Preparation	Presentation of the cards and the estimation procedure
3. Estimation according to iteration I – IV	Providing epic example, the cards and estimation scale in randomized order
4. Questionnaire	Demographics
5. Conclusion	Summary

1. **Introduction.** We explained the study purpose, the participation conditions, the estimated completion time, the study procedure, and the data protection guidelines. We informed the participants that we would donate ten euros to an organization of their choice after completing the survey. The participants had to give explicit consent to be forwarded to the survey.
2. **Preparation.** We introduced the participants to our *Sustainability Cards* and the upcoming estimation assignment. We explained to the participants that we would give them a requirement and asked them to estimate its impact on a software product's sustainability. To this end, we provided them with our cards and a rating scale. We gave the participants the opportunity to explain their estimation in a comment box for each estimation.

3-I. Estimation – Iteration I. We asked the participants to imagine themselves as members of a development team for a ticketing platform. We gave them the requirement to improve the platform by using blockchain. We showed the participants in a randomized order our artifacts, as depicted in Figure 24. Participants had no time limit to read through the cards and make an estimation.

3-II. Estimation – Iteration II. For the second iteration, we were interested in whether participants would evaluate our estimation approach differently if we changed the scenario and requirement. We asked the participants to imagine themselves as members of a development team for a social media platform. We gave them the requirement to allow the usage of the platform only by an invitation of an already registered person. We showed the participants in a randomized order our artifacts, as depicted in Figure 24. Participants had no time limit to read through the cards and make an estimation.

3-III. Estimation – Iteration III. For the third iteration, we used the scenario of the development of the social media platform and its restricted access. We showed the participants in a randomized order our artifacts, as depicted in Figure 25. Participants had no time limit to read through the cards and make an estimation. In the first two iterations, we primarily used the subdimensions to describe each dimension more concretely and to ask questions that would help participants better estimate the impact of the requirement. Since we could observe in the previous iterations that participants could not consider all questions equally, we enabled the participants to estimate the requirement based on the subdimensions. Furthermore, the participants suggest more response options. Therefore, we extended the rating scale by two other options: "I don't know" and "Neither positive nor negative influence." We showed the participants in a randomized order the dimensions. Participants had no time limit to read through the cards and make an estimation.

3-IV. Estimation – Iteration IV. Since we got positive feedback regarding the option to rate the subdimensions separately, we prepared another set of cards, as demonstrated in Figure 26. We presented each subdimension in a single card. For the fourth iteration, we provided the scenario of the ticket-selling platform and its blockchain integration. We showed the participants the cards in a randomized order. Participants had no time limit to read through the cards and make an estimation.

4. Questionnaire. To collect participants' details, we asked for demographic information and their working experiences. Before submitting their answers, we provided the participants with an opportunity to leave comments.

5. Conclusion. After completing the study, we asked the participants to decide which initiative we should send our donation to. We thanked them for their participation and invited them to the following focus group discussion.

8.3.2 Group Estimation – Focus Group Discussions

We conducted four semi-guided focus group discussions for each iteration. The discussants were the participants who performed the individual estimation of the respective iteration and one facilitator. We prepared each discussion with the online tool Miro to present the individual estimation results to the participants. We allowed the participants to leave notes. Each discussion lasted approximately 80 minutes. We audio-recorded the conversation with the participants' permission and fully transcribed it for our analysis. Following Morgan and Hoffman [145], we prepared a discussion guideline in German. We included an English translation of the guideline in Appendix C. We had three parts: an introduction, a discussion, and a conclusion. Due to the different focus of the iterations, we organized the discussion as follows:

1. Introduction. All participants in the discussion introduced themselves. We informed all participants about the general conditions, such as duration, and asked permission to record the discussion.

2-I-II. Discussion Iteration I and II. As a reminder, we presented the respective requirement, the individual estimates, and the reasons for the participants' assessment. To start the discussion, we focused on the estimations, which differed from each other. As in *Planning Poker*, we asked the group to discuss their initial estimations and whether they could agree on a shared estimation.

Furthermore, we asked the participants to report what they liked or disliked about the dimension cards.

2-III. Discussion Iteration III. As a reminder, we presented the respective requirement, the individual estimates, and the reasons for the participants' assessment. Since we adjusted the rating scale, we started with the subdimensions, which the participants rated with "I don't know," and continued with the estimations which differed from each other. We asked the group to discuss their initial estimations and whether they could agree on a shared estimation. Furthermore, we asked the participants to suggest when they would apply our estimation approach in a Scrum Process and what they like or dislike about the dimension cards.

2-IV. Discussion Iteration IV. As a reminder, we presented the respective requirement, the individual estimates, and the reasons for the participants' assessment. We started with the subdimensions, which the participants rated with "I don't know," and continued with the estimations which differed from each other. We asked the group to discuss their initial estimations and whether they could agree on a shared estimation. Additionally, we asked the participants who should be involved in an estimation method like this one. Furthermore, we asked the participants for their assessment regarding the description and the selected icons.

3. Conclusion. During our last part, we concluded with the remaining questions from the participants and thanked them for their participation.

8.3.3 Analysis

Our analysis is based on the focus group discussions, and the participants' notes provided on the Miro board. We asked the participants during this discussion about the individual estimation during the survey. We transcribed the group discussion and performed a thematic analysis. We conducted a thematic analysis according to Clarke and Braun's [38] approach, which included six recursive steps: "1. Familiarizing yourself with the data and identifying items of potential interest, 2. Generating initial codes, 3. Searching for themes, 4. Reviewing potential themes, 5. Defining and naming themes, 6. Producing the report."

1. **Familiarizing.** First, we worked independently on our respective sets. Throughout the process, we read and reread each transcript. According to the research question, we identified potential insights in the data.
2. **Generating Initial Codes.** We generated initial codes to capture the most exciting aspects of the data. We decided that each of us would compile a list of codes.
3. **Searching for Themes.** Based on our list, we searched for initial themes. After that, we met and discussed our list of codes and themes.
4. **Reviewing Potential Themes.** As part of our evaluation process, we examined our identified themes to determine whether they captured the most critical aspects of the data and whether they were relevant to the research question.

5. **Defining and Merging Themes.** During this step, we extracted the most appropriate themes, merged themes, and formulated new themes if necessary. Using these themes as a guide, we reread and analyzed our transcripts once again.
6. **Producing the Report.** By reporting the results in Section 8.4. we refined the analysis. Our objective was to present our analysis comprehensively and insightfully.

8.3.4 Sample

In order to acquire participants, we used convenience and snowball sampling [217], [107]. We evaluated our estimation approach, *Sustainability Poker*, with twelve participants. We consider the procedure an integrative approach that can fit into various frameworks, e.g., *Scrum*. To this end, we formed four interdisciplinary groups, each consisting of one product owner, developer, and UX/UI designer, as summarized in Table 48. Our objective was to recreate a group that could also occur in practice. Furthermore, we focused on interdisciplinary teams since implementing sustainable software products requires contributions from different disciplines [13]. Each group has one female member and two male members. Only one participant was a student, while the other eleven were employees at a company. Our participants were 29 years on average.

Table 48: Overview participants and groups

ID	OCCUPATION	EXP. YEARS	RE EDUCATION	PRIVATE EPNGAGEMENT	ITERATION
PO1	Product owner	6-10	No	Yes	I
D1	Developer	6-10	Yes	Yes	I
UD1	UX/UI designer	1-5	Yes	No	I
PO2	Product owner	1-5	No	No	II
D2	Developer	1-5	No	No	II
UD2	UX/UI designer	1-5	Yes	Yes	II
PO3	Product owner	1-5	No	Yes	III
D3	Developer	6-10	Yes	No	III
UD3	UX/UI designer	1-5	No	No	III
PO4	Product owner	6-10	No	No	IV
D4	Developer	1-5	No	No	IV
UD4	UX/UI designer	1-5	Yes	Yes	IV

8.3.5 Pre-Study

We conducted a pre-study with two participants to examine our research design and a preliminary version of our cards. The first participant was a developer, and the second participant was a UX designer. We asked them to perform the individual estimation using the *think-aloud* method [100]. According to the participants, the questions lack structure. Therefore, we have added headings to the questions. Our adjustments resulted in subdimensions, which we visualized with icons to assist future participants in grasping the subdimensions more readily. Furthermore, in our preliminary dimension cards, we included the software quality dimensions of the *ISO standard 25010:11* and positive examples. We included these ISO dimensions since our interviewees of Chapter 5 suggested it, and Condori-Fernandez and Lago [40] perceived sustainability as a matter of software quality. However, the participants stated that this amount of information would not contribute to the estimation. Hence, we reduced the cards to only our dimensions. The pre-study enabled us to identify spelling errors and find more optimal wording for the questions to make them easier to understand.

8.4 Results

In the following, we present the results of Iteration I in Subsection 8.4.1, Iteration II in Subsection 8.4.2, Iteration III in Subsection 8.4.3, and Iteration IV in Subsection 8.4.4. We report the iterations according to the difficulties that participants faced in applying our approach, followed by the benefits that participants saw in the procedure. We conclude with the participant's recommendations on how to improve our approach.

8.4.1 Iteration I

Difficulties

Lack of Knowledge and Experience. The participants reported difficulties estimating the influence due to their lack of knowledge regarding the dimensions. For example, UD1 stated that estimating influence on the legal dimension was challenging. PO1 told us that "I really have no idea how to evaluate blockchain legally. It may be about security, but security is not legal. No idea." D1 could not imagine the influence on the individual dimension and questioned whether all dimensions are appropriate for all cases. Additionally, UD1 had not enough knowledge about the requirement: "I just don't know enough about blockchain."

Project-Phase Dependent. Whether an estimation is possible depends on the current state of the project phase. UD1 stated, "So, for example, this point about aesthetics, you can only estimate that at the very end, when something is actually already there, somehow."

Missing Requirement Details. The participants, who were familiar with blockchain, reported missing additional information about the requirement. For example, D1 said, "To really evaluate it like that, it would just have to be described in much more detail."

Card Design. PO1 pointed out that the cards contain too much information. Therefore, PO1 had difficulties in estimating the potential influence.

Insufficient Response Options. The participants reported that the rating scale did not cover enough response options. UD1 pointed out that the provided requirement could have positive and negative influences in some cases, "the mean value that you could give was 'no influence.' I found that inappropriate. It can just swing in both directions." UD1 continued, "So no influence can, of course, also be. Mostly it has some, [...] so that positive and negative just actually balance each other out." PO1 agreed with UD1 and stated: "[...] both advantages and disadvantages. 'No influence' is not right."

Benefits

Diversified Thinking. All participants believed that the estimation process gave them a versatile way of thinking about sustainability. UD1 emphasized: "And that's already good, simply to show which aspects should play into it at all." D1 found it particularly interesting that the breakdown into the dimensions allows project participants to look at possible influences separately: "And I think that if you really look at it that way, then I think it's interesting, especially now, to really have it broken down like that and to really have the opportunity to look at just the social part first." In addition, the cards offer aspects the participants would not have thought of, as UD1 and PO1 reflected: "Well, I would not have thought of all the fields of action at first."

Visibility of Participants' Views. PO1 appreciated that the estimation procedure revealed the diverse opinions of participants. This has broadened the horizon of PO1 and provided PO1 with additional aspects. D1 agreed with PO2. Additionally, D1 welcomed the overall rating results "I always find that very good when something like that just becomes quantitative somehow quickly, that you just really have a comparability." However, D1 expressed that the overall rating representation does not need to be a numerical value: "I think the ranking doesn't have to be numerical."

Save Implementation Time. UD1 stated that stakeholders should apply the estimation procedure at the beginning of a project. The procedure might encourage stakeholders to think about certain aspects more intensively before their implementation: "Yes, so I think it would definitely make sense at the very beginning before this system is even developed."

Card Design. In general, participants found the cards comprehensible. D1 even said, "I didn't have any point where I wasn't sure what was meant." UD1 found it "[...], that's actually all understandable too, so these subitems and questions." D1 found wordings and colorful design helpful in remembering the content: "So I could recall it again through the color and the headlines."

Recommendations

Estimating Subdimensions. The participants expressed that the estimation procedure seemed very extensive. UD1 reported that: "Well, I think that's just really a point where it gets stuck afterward, that it's not used the way it should be used," and

UD1 worried that “then (people) also just pick out the points that maybe they’ve already thought about anyway.” Nevertheless, UD1 also acknowledged that it takes time to address the issue of sustainability. PO1 pointed out to reduce this extensive impression so “that you’re not always deterred by this big chunk” in order to build sustainable software. To counteract this impression, PO1 suggested estimating the questions separately: “[...] I would start discussing these subcategories individually, and I think I would have gotten through much faster if I had simply said, ‘yes, no, positive, negative’ [...] for each subitem, and then it’s an average afterward. Then cognitively, I would have gotten through that mass much easier.” Furthermore, D1 also reported that the perception of the procedure also depends on its purpose. D1 said, “It depends if it’s a reference work, which you somehow just have on the table, which you read through once or so, then it’s exactly super.” However, D1 stated, “if it’s supposed to be somehow accompanying the workshop, which you’re supposed to hang on walls live somehow, then it’s a bit too extensive after all.”

Applying at the Beginning. Even though the effort seems extensive, the participants did not perceive it as impossible to integrate the estimation procedure into their professional lives. UD1 imagined using the procedure before an ideation phase. PO1 suggested splitting the dimensions and applying specific dimensions at different phases in the project: “[...] certain aspects right at the beginning [...] for example, ‘social, legal, individual, economic’ and so on. [...] part of such a, with us, it’s called Product Brief.”

Consider Third-Party Providers. D1 missed the consideration of third-party providers.

8.4.2 Iteration II

Difficulties

Lack of Knowledge and Experience. In the second iteration, participants also reported that they found the estimation procedure difficult due to their lack of knowledge. For example, UD2 also reported struggles with the legal dimension. Furthermore, UD2 added, “I had no experience at all in estimating software.”

Project Phase Dependent. PO2 pointed out that the estimation might require adjustments over time due to circumstance changes: “[...] what time frame are we looking at this? And what are actually the other metrics that are playing a role in the background that we can assume are also changing, like increasing user numbers [...].” PO2 wondered whether this example negatively influences the ecological dimension since the increased user number might require more energy. However, PO2 added whether this estimation would change if the system runs on renewable energy: “You could think about that again, where you draw the line then, up to where you want to evaluate that.”

Missing Requirement Details. The participants wished for more details about the provided requirement. For example, PO2 missed the reason why the registration needs a restriction. It would also have been helpful for PO2 if there had been mock-ups of the platform. D2 was unsure which perspective D2 should have taken: “There we have, basically, this point again, from which perspective do we see this? Are we looking only at the actual users of the platform or humanity as a whole?" D2 would have found it easier to perform the estimation if we had provided more information about the circumstances: "I would have actually quite liked to have a reference point, and where I know we are technically right now, I don't know [...] a value seven on some scale."

Card Design. The participant of the second iteration also emphasized that the cards contain too much information. UD2 stated, "Well. They are just slightly overwhelming. There's just a lot on them, and then there's super small text on the background up there." On the one hand, PO2 confirmed this statement, but on the other hand, PO2 also said: "[...] of course it works. I think if you want to read it and you're interested in it, it's not a hurdle at all." D2 added that some of the questions were too general for D2, and "maybe that makes it harder sometimes to really focus."

Insufficient Response Options. The participants also perceived that the range of response options was not sufficient. UD2 commented, "I think I struggled the most with saying it's a negative or positive influence. There I would have preferred to say how much influence it has, in some cases, instead of saying no influence. Because 'no influence,' I thought it was wrong then. But I didn't find the influence positive or negative." PO2 added that in principle, PO2 could imagine an influence but could not imagine the consequences: "[...] the dimension has an influence, but I don't know if it's negative."

Benefits

Diversified Thinking. Participants appreciated being able to take different perspectives, as reported by UD2: "But I thought it was super cool to look at something like this, from all these aspects." D2 agreed with UD2, "I also thought it was super inspiring; it was interesting to look at something like this." Furthermore, the participants also emphasized that the process brought them to sustainability aspects that they would not have thought of in the first place. PO2 reported: "And I also find it refreshing that it is not only ecological. I think that's good, too, but that's the first association." PO2 liked the questions on the card regarding the social dimension: "Above all, I found community spirit and trust to be good questions to help me think about it." D2 confirmed: "Yes, it is interesting to see what sustainability can actually mean. People always look at ecological influences. But that's nonsense. You can see it through that." UD2 agreed: "And it also helps to see these as sustainable aspects in a holistic way and to see them as sustainable aspects for one's daily work. Because otherwise you always think quickly only of ecological, but it is so much broader." UD2 summarized that the procedure motivates people to think about the effects of the desired requirement.

Saving Implementation Time. Through this multifaceted discussion, the participants assumed that projects could save unnecessary implementation time since they must analyze their projects extensively. However, PO2 noted that stakeholders must first decide whether they want to estimate a requirement at all.

Recommendations

Estimating Subdimensions. Initially, the procedure seemed very extensive to the participants. PO2 commented, “The first look was, wow, that’s a lot. Do I really have to read through everything now? That was kind of the initial reaction.” However, after further examination of the cards, UD2 said, “Even if there’s not much at all, I have right now here social right in front of me, I mean, there’s even under the points always just, once three paragraphs. It’s actually not that much, but it seems like that for now!” The participants also suggested estimating the subdimensions separately. D2 described: “So really, that you go through question by question and have this scale in front of you and estimate how this question can somehow have a negative or positive influence” Because of the seemingly enormous scope, UD2 also cannot yet imagine how stakeholder can apply this procedure in practice.

Sharing Responsibility. PO2 suggested assigning single dimensions to individual stakeholders: “[...] such an integration card I think; you can give just super well to product owner with a business owner [...] and say, this is actually your homework [...].” PO2 continued: “Make a card game out of it, and everyone gets assigned a card every week and just has to think about it and can say in the stand-up or wherever, somehow, what he thinks about it. Just to stimulate diverse thinking.”

Customizable. The participants imagined making the procedure customizable, as PO2 proposed: “[...] the recommendation is that each company, each project team, defines this for itself, also with its own examples. So that there is just the software name [...] remove some points perhaps [...] in order to sharpen the context then.” D2 suggested weighting the dimensions: “Because I think for me it is then a question of how much, weighs which dimensions? Are they really all equally important to me and my business? Or can it be that somehow something else is in the front for me? If I have an internal solution, for example, then I probably wouldn’t care about the aesthetics at first. Just that kind of aspect, how do I weight each category?”

Representation of Overall Estimates. The participants questioned whether a numerical value must emerge from the estimation procedure. PO2 commented, “Or is it just enough to just say, as UD2 also said, we just found a deviation, or there’s something there [...] by just finding it out, we can say, okay, we kind of have to watch out for mental health. So, that’s just learning that we have from that. We have to put that in there, and I don’t really care if it’s kind of worth 5 points to me or 7 points to me or whatever.” Still, PO2 continued that “If I say, I want to use this properly for tracking, and I want to know, how did our sustainability score actually change, then, sure, I kind of have to keep that in mind.” D2 had the impression that an overall conclusion might not be possible.

Applying it Quarterly. D2 suggested not applying it every day but several times during the year: “But if a quarter is planned, then you can definitely include something like that.”

Consider Third-Party Providers. Furthermore, D2 also mentioned that we did not consider third-party providers.

8.4.3 Iteration III

Difficulties

Lack of Knowledge and Experience. The participants also reported that they had difficulties estimating the requirement due to their lack of knowledge. For example, PO3 struggled to estimate the influence regarding the technical dimension: "I couldn't estimate it at all. No idea what it takes."

Missing Requirement Details. D3 would have liked more details: "Then I also lacked context from the network. What is our business case, our direction? And where do we stand?" Therefore, D3 was unsure whether D3 applied the cards correctly. However, it would improve over time once D3 used the cards more often: "That's probably where it helps just to use this as a team every now and then."

Card Design. D3 reported that "some examples were not clear" and suspected "overlap between sub-items." UD3 would have expected "partly different questions in the headings." Also, it was not clear to UD3 "whether individual points refer to end users or other stakeholders." Furthermore, UD3 noticed: "Not every dimension can be correctly estimated by every person (with a different background)."

Customer Dependent. Whether the process will actually have an impact on the product also depends on the customer, as PO3 noted: "Nevertheless, we decide to follow the wishes of the customer since he gives us millions every year so that we can continue here."

Benefits

Diversified Thinking. Participants acknowledged that the cards encouraged them to take different perspectives. For example, D3 reported, "I also thought it was nice to really look at a (requirement) from a variety of angles, which is something that often receives too little attention in software development." PO3 and UD3 also found it an interesting experience to work with the cards, as UD3 described: "I find it generally very interesting and get a starting point like that." Furthermore, PO3 emphasized that "please solution ideas (should be) already discussed in advance, to what extent they are also ethically justifiable [...] what potential impact this can have, also on human communities, also economically and ecologically."

Saving Implementation Time. By using this procedure, project participants would work on questions in advance that they might have forgotten or answered too late in the project. PO3 commented, "There are many people who forget to include, I think legal."

Providing Arguments. The participants reported that they could imagine identifying arguments for considering sustainability-related aspects. This could also reduce arbitrariness, as UD3 stated, "I can imagine that it helps [...]. From practice, I can say that decisions are often made very emotionally [...]."

Visibility of Participants' Views. Our participants perceived that the procedure's most crucial advantage is that it results in a shared discussion base. PO3 reported that the procedure makes it possible to discuss the potential characteristics of a project in advance. The project participants can express their experiences accordingly and make them visible to everyone: "[...] that you really see it all again, all together. That one could then say together at least, we have seen that." PO3 reinforces his statement, "The discussion is actually the valuable thing at the point, I think." D3 feels that the discussion resulting from the process is very important and could also imagine using the results for other methods: "I also think it's cool to have done this on my own and then use these different results as a basis for discussion for a backlog redefinition."

Card Design. D3 stated that the procedure "is visually very appealing," "(contains) good examples, can be easily extended for specific requirements/dimensions of the project," and provides a solid "framework for structured discussions." UD3 appreciated that it has a "simple presentation," and it seemed to be "understandable for everyone," and provided a "good overview of all important factors." PO3 stated that it "basically (are) helpful dimensions, gives structure to discussions and is visual" and further adds that it is "helpful for documentation of decisions, especially far-reaching ones."

Recommendations

Refining Dimensions. PO3 noted that some subdimensions should be revised again: "[...] there were a few sub-questions that seemed very similar to me" Furthermore, D3 was missing how they should consider end users: "For example, I was also somehow missing what the end user would like to see included as an additional dimension. How would they actually find the feature?"

Representation of Overall Estimates. PO3 was unsure whether an overall representation would add any value. In contrast, D3 said a visual presentation would be important: "Exactly, I would do the overall evaluation. It should be somehow visually summarized on one slide."

Applying at the Beginning. UD3 reported: "I agree that this should be done at the beginning of a project, no matter what it's called or if there is a product discovery in such a form." D3 followed with: "When you initially set up something new. It makes total sense to apply that, if you really do ideation, wildly brainstorming ideas, that you then apply them to these different dimensions and try out what comes out of it." In addition, D3 can also imagine applying the process to larger requirements: "And I also think if you have larger epics and then break them down in the refinements, you can still apply that there."

Customizable. Participants believed the procedure should be customizable to apply in everyday work. PO3 suggested that stakeholders should be able to adjust the procedure according to the company/business unit/case. Similarly, D3 also expressed, "What I just wonder, some cards or dimensions may not be useful for all projects somehow. If I'm developing a completely internal thing now, I don't know if I should somehow always consider the social component there."

Monitoring. Furthermore, a way to track the results would be helpful, as UD3 said: "Maybe somewhere you can also, somehow, steadily monitor that." Similarly, PO3 reported, "That's exactly what UD3 was talking about. [...] in terms of hard KPIs [...] at least take a look at it again, does it still fit?"

8.4.4 Iteration IV

Difficulties

Lack of Knowledge and Experience. Participants reported that adapting to the estimation procedure was initially difficult since sustainable software was an unfamiliar topic. PO4 commented: "But it was, I think, a difficult situation to get into at first, to slip into it from a completely different context, so to speak." UD4 had a similar experience: "First, I had to think my way into the topic. I haven't had much contact with it for a while." In addition to the lack of experience, the participants also stated that they had little knowledge of the requirement. UD4 reported, "I'm not that deep into this topic that I could judge that." Accordingly, UD4 frequently responded with "I don't know": "But yeah, I think I've used so frequently 'I don't know' because I don't want to make any claims that I can't substantiate specifically." This was also the case for D4: "And UD4, you had said earlier 'I didn't want to answer with half-knowledge or not even half-knowledge, [...] I just had that feeling a lot, too."

Missing Requirement Details. The participants would have liked more decisive information about the requirement. D4 reported: "Okay, we want to use blockchain somehow to achieve something. But how it will be implemented [...] is critical to many of those effects." UD3 wondered regarding the aspect of privacy: "How is that represented in the system? [...] what data is actually stored there? That's not at all clear to me, so I couldn't give an estimate now." Due to the lack of information, D4 could have imagined all possible influences, which made it difficult for D4 to provide an estimate: "Well, I can imagine all kinds of things, and that's why it was for me 'I don't know,' I can't judge, I don't know enough or it's too unspecific for me to be able to judge it." PO4 first needed to imagine the implementation of the requirement: "So first to understand what the exact example means, what it also has for effects, how you could design that."

Implementation Dependent. Furthermore, D4 states that the estimation also depends on how the requirement is implemented: "In the final analysis, I would say that it [...] initially does not influence privacy. How that is implemented, of course, impacts the privacy or can have one, and that can be both positive and negative, depending on the design."

Dimensions Influence Requirement. D4 found it challenging to give an estimation. D4 would have found it more appropriate to think about how the dimension influences the requirement and not the other way around: "I just think the requirement does not influence regional differences. Regional differences may influence the requirement, what I can do, for example, and how that then plays out in the software."

Benefits

Diversified Thinking. The participants appreciated that the estimation procedure has stimulated them to think more intensively about the requirement itself than just about its influence, as UD4 stated: "[...] because the cards already give you much more than that is rather positive or negative, but rather so that you think a little deeper." In this regard, UD4 likes that the process also stimulates thinking about the product's durability: "[...] I actually think that's cool, that it just stimulates this long-term thinking."

Visibility of Participants' Views. The Participants also valued the shared discussion base that resulted from the procedure. D4 said: "Otherwise, the discussion, the talking about it, makes it, I think, always easier to exchange ideas and to develop one's own thoughts." PO4 agreed with D4: "I thought the discussions were great and as D4 also said, you also come to a good result through the discussion, I think."

Card Design. The participants perceived the card design as very well done. PO4 described: "And the cards themselves are actually also well designed, in the sense of the text also [...] One can imagine something with it."

Recommendations

Refining Dimensions. Participants perceived the procedure as very time-consuming in estimating a requirement. Thus, PO4 said: "[...] I think it's just relatively [...] a lot of effort." Furthermore, PO4 suggested rethinking the direction of the influence. Some questions are adequate to estimate the requirements' influence on the respective dimension, and other questions should be used to answer what influence the dimensions have on the requirement. PO4 exemplified this observation by describing the legal dimension: "I think the first and second questions don't really fit together. The title says regional differences in laws, and in the first question, you are asking whether this new artifact will make the software available to other countries. [...] in the rarest cases, a (requirement) is responsible for the transition of the software to another country."

Estimation Example. For a reliable estimation, the participants would have liked an example, as D4 reported: "and then some examples where somehow this thought structure becomes clear how things should relate to each other and what the real questions are." UD4 and PO4 would also have liked an example. However, PO4 indicated that after a certain period of familiarization, the estimation became easier: "Well, a concrete example would have helped me a bit more. I then read it through a few more times; then it worked at some point."

Card Design. D4 reported regarding the icons, "when I see the picture, I just don't think directly about freedom of action and in that context. When I just skimmed headline text, I found the logo matched." UD4 made the following suggestion, "[...] so I imagine more like a choice of option, visually." For the user-centered aspect, D4 found the chosen icon irritating, "I actually think it was just the image again. Because it's user-centered, maybe a user in a circle like that, central in the middle. It was just the image again that didn't fit so spontaneously." With regard to the dimension Integrative, UD4 had a different idea: "Somehow the icon does not signal integration to me [...] Yes, well, for me it does not really symbolize an integration of something else, a system that integrates another system. Somehow it doesn't really reflect my idea." D4 agreed, "Yes, similar for me. I had thought of it so intuitively more along with third-party (providers)."

Involving More Domain Experts. D4 stated, "The bottom line is domain knowledge. A developer, when in doubt, cannot somehow estimate all dimensions." D4 later added, "So someone from the business development team, they have some domain knowledge. They're usually not the super experts, they have this high-level view, and that's often enough to get the business perspective in." Depending on the case, perhaps someone from the event team, as UD4 suggested: "Yes, in addition to the development team, you could perhaps also bring in event organizers as experts. In their network, they're often up to date on what's kind of being requested by customers right now." UD4 proposed that customer service should also be brought in. D4 agreed with UD4: "Customer service, they're often the people who know the most because they interact with customers on a daily basis."

8.5 Discussion

In the following section, we discuss the implications of our results in Subsection 8.5.1 and threats to validity in Subsection 8.5.2.

8.5.1 Implications

We aimed to encourage the participants to think deeply about the potential impacts of requirements on the sustainability of a product. To achieve this, we prepared an estimation procedure that consists of a requirement example, cards containing questions corresponding to the nine dimensions of the set Dim_{SUST}, and a rating scale.

The analysis of the difficulties revealed that participants' lack of expertise was the most challenging factor. This observation is in line with Seyff et al. [183]. In their study, participants had difficulty defining the potential long-term effects of a given requirement on sustainability. A procedure such as ours may require expert knowledge that the procedure itself cannot provide. However, it may reveal which expert knowledge a particular project requires. Furthermore, the participants would have appreciated more information regarding the requirements in each iteration. Considering that it is common in practice to have only a limited amount of information, we did not change the requirement. As a result, the procedure may not be able to compensate for missing details, but it may be able to reveal the gaps regarding the reasons why the project should or should not pursue the requirement in the first place.

Even though the participants lacked experience and information, they confirmed that the procedure became easier with practice. Sustainable software design may always require this effort but perhaps one day it will become self-evident, and stakeholders will not consider it an additional effort. The participants perceived the procedure as quite extensive and reflected that it might take too much time in their daily working routine. This may also be related to the fact that our requirement example was quite small. Nevertheless, the participants acknowledged that this kind of effort might be necessary to design sustainable software products and that they could apply it at specific points in the project rather than on a regular basis.

We considered frequent suggestions for customizability, such as weighting dimensions, a “double-edged sword.” The possibility of customizing might generate a greater level of acceptance. However, it also carries the risk that organizations will select only those aspects that they can quickly implement or that are beneficial to them.

The participants also reported difficulty applying the procedure because they were uncertain whether their estimations had been correct. In this instance, we should have clarified that there is no right or wrong approach. This observation indicates what the fundamental difficulty of sustainable software design is. Since stakeholders must make decisions based on assumptions, it is difficult to predict in advance which course of action will be appropriate.

Although the procedure enabled a diverse discussion, the overall estimation did not result in concrete action instructions. The results can serve as an information base for the person responsible for the product’s sustainability to understand how the project participants perceive a particular requirement or system. The participants also appreciated that the procedure revealed the differing perspectives of team members regarding the exact requirement. Our procedure can make voices visible that may not have been heard otherwise. Despite this, the participants believed that experiences and values influence estimation, which makes the process subjective.

However, participants indicated that the multifaceted procedure facilitated a holistic analysis of the requirement, reducing the risk of neglecting essential implementation aspects.

8.5.2 Threats to Validity

By using convenient sampling and involving acquaintances, perhaps the so-called *demand effect* [153] could have occurred. Since we received positive and negative feedback, we can assume that our participants might not have been too exposed to the effect. We offered the participants the opportunity to select an initiative we should donate to as an incentive. Furthermore, we randomized the sequence of the dimensions to ensure that the order does not influence the results.

Our pre-study was very small. Nevertheless, we used the gathered feedback to examine whether the participants correctly comprehended the assignments and questions. Only one facilitator conducted the discussion. However, we were two coders who performed the analysis separately. Our participants were very similar in their age. Additionally, they were all interested in the topic of sustainable software design. This interest might have influenced the results. Our study may not be replicable because we conducted it in German and translated the procedure and the results for this report. It might lead to different results if others conduct this study in a language other than German.

We conducted the procedure online. Therefore, we cannot completely exclude whether our participants asked for help during the self-estimation. However, since they were only supposed to give an estimation that had no consequence for the participants, we can assume that they did not bother to ask for help. Twelve participants are a relatively small size. Nevertheless, we gained insights that we treated as hypotheses rather than solid facts. We chose requirement examples that were relatively small granulated since they fit into the scope of the evaluation. In practice, stakeholders might select an entire product or a bigger intervention into an existing system.

8.6 Related Works

There has already been a considerable exploration of dimensions and corresponding questions to create a common understanding of the impact of software systems. For example, Lago et al. [112] and Duboc et al. [50] have researched this area. The latter authors proposed a “question-based framework for raising awareness of the potential effects of software systems on sustainability.” We used these questions as a fundamental part of our card-based communication and estimation approach.

The research presented in this thesis has not yet integrated these questions into established estimation procedures to obtain a comparable overall value. However, we already have established estimation procedures, such as *Planning Poker*. A survey by Usman et al. [199] shows that *Planning Poker* is one of the most popular estimation methods for estimating the implementation effort of requirements in practice. Gandomani et al. [68] observed in their case study that a consensus on effort estimation of user stories found with *Planning Poker* can be more accurate than an average size estimation. However, the application of *Planning Poker* assumes that experts are involved. Despite this, Alhamed and Storer [4] reported that when we carefully group non-experts together, they can arrive at an estimate similar to that of an individual expert. Furthermore, the research of Mahnič and Hovelja [114] indicates that *Planning Poker* structures communication and ensures that all team members can contribute equally while dominant characters have less influence on the outcome. In contrast, Curcio et al. [46] observed that it could polarize groups and lead to extreme estimations.

Thus, previous research indicates that the use of *Planning Poker* for effort estimating requirements is widespread and can be successful. We have taken this as an opportunity to use it for the estimation of requirements on sustainability. Nevertheless, there are other examples of using cards, such as the *Envisioning Cards* [67]. Stakeholders can apply these cards "to raise awareness of long-term and systemic issues in design [147].” The *Behavior Change Design Cards* [109] is another example that assists designers in choosing behavior-changing techniques to achieve their design goals. While these cards assist in evaluating the impact of a design, their application does not lead to result in a comparable value.

8.7 Summary

We prepared an estimation procedure based on our set Dim_{SUST}, which consists of an individual estimation and group estimation. To perform the estimation, we provided cards, a rating scale, and an example of a requirement. We evaluated our procedure with an online survey and focus group discussions. We described the research design, including the evaluation procedure, analysis approach, sample, and pre-study. Analyzing the results, we could identify what the participants found difficult and beneficial. Furthermore, the participants provided us with each iteration recommendations on improving the procedure, which we partly implemented according to our research goal.

The results revealed that our procedure promotes diverse thinking beyond the ecological interpretation of sustainability. Additionally, the procedure supported the participants in expressing their different views in a shared discussion based on their preceding individual estimation. However, the participants perceived the multifaceted analysis as time-consuming and challenging to apply daily. Nevertheless, the participants suggested applying the procedure at specific points in the project phase, e.g., before implementing the requirement.

In summary, we could observe that interdisciplinary teams were able to perform the estimation. Since it was able to assist the participants in expressing and understanding each other views, our procedure could contribute to fostering communication and raising the chance of enabling sustainable software design

PART III: SYNOPSIS

9 Sustainability-Centered Design

Publication. This chapter is partly based on our article *shapeRE-Framework: Forschungsvorhaben zur Ermittlung und Repräsentation von Anforderungen nachhaltiger Software-Innovationen* [161]. My contributions to this paper comprise the literature study, the framework design, the discussion, and leading the writing of the paper.

Contribution. This chapter introduces our framework, *Sustainability-Centered Design*, which includes proposing the new role of the *Sustainability Design Master* and a corresponding process. Our goal was to suggest a guiding design track that runs parallel to the implementation track. These tracks have specific intersections, where the *Sustainability Design Master* provides the implementation track stakeholders with the relevant requirements and evaluates whether those were sufficiently considered to facilitate a sustainable software product. To this end, we prepared a process that should guide the *Sustainability Design Master* in identifying and representing requirements according to the project's progress.

9.1 Motivation

While agile approaches suggest that it is easier to manage many small work cycles [95], the sum of their results does not necessarily create a coherent product. Moreover, we suggest that individual parts must be "composed" or designed with a specific willingness to shape future events or change current situations to ensure holistic and sustainable products. The most effective way to fulfill this design task is to delegate it to responsible people or roles. According to Schuler and Görlich [179], hiring creative employees is the best way to develop a company's creative potential. However, they also emphasize the importance of appropriate processes and resources. In terms of sustainable software design, this would mean that hiring innovative and sustainability-oriented people is the best way for a company to promote sustainable software design. To support companies to deploy these people in a meaningful way, we propose the framework *Sustainability-Centered Design*. This framework proposes establishing a new role and a requirements representation guideline integrated into a process extension that should complement and guide current development processes, such as agile approaches like *Scrum*.

We report the proposal of our framework in the following sections. In Section 9.2, we present the background and our theoretical foundation. In Section 9.3, we describe our framework and discuss our proposal in Section 9.4. We conclude with related works in Section 9.5 and a chapter summary in Section 9.6.

9.2 Background

Based on our comparison analysis in Chapter 4, we proposed the new role of the *Sustainability Design Master*. Additionally, we drew on the leadership concept of Hill et al. [93] and the description of an innovative person by Schuler and Görlich [179]. Since designing sustainable software products might require new ideas, as described in Section 2.2. Schuler and Görlich [179] describe a person who can implement and enforce innovations as, e.g., having an open mind, being able to combine opposing viewpoints, being realistic (recognizing possibilities and limitations in a given situation), being able to team up and co-operate.

The preparation of the process involved two additional approaches. Sys' process proposal [192] represents the first approach. It describes an iterative process in which design activities have their own track, which starts up front. The results of the design efforts then serve as a foundation for the development team in the implementation track.

The second approach is the communication method of building architecture [187]. In building architecture, descriptions and drawings serve as a means of communication between all project participants [180]. According to the course of the project and the addressees, different scales are available in building architecture, e.g., 1:500 for the site plan or 1:10 for the detailed planning. Those scales determine the content and level of detail. In line with this representation principle, we suggest a level-based guideline for expressing requirements based on the project's progress. Using this guideline, *Sustainability Design Masters* can assess the project status and initiate appropriate actions to advance the project.

Due to the limited guidance on representing requirements according to the project's progress [182], we prepared this representation guideline. Furthermore, most requirement documentation techniques, such as user stories, mainly use textual representations [46]. However, Wiegers and Beatty [211] stated, drawing on Davis [47], that it is necessary to have "a combination of textual and visual requirements representations at different levels of abstraction to paint a full picture of the intended system." There are many approaches to supplement, e.g., user stories [46], but only a few addresses how we can combine graphical and textual representations according to the project's progress.

9.3 Framework

In the following section, we present the *Sustainability Design Master* role in Subsection 9.3.1. In Subsection 9.3.2, we describe the corresponding development process, focusing on the activities of the new role and the project artifacts.

9.3.1 Sustainability Design Master

In reference to the *Scrum Master* [181], we have chosen the name *Sustainability Design Master* for the new role. Similar to the *Scrum Master* and product owner [181], we define the role as an all-encompassing authority that keeps the project together and guides it with a sustainability-driven attitude. In the following section, we describe the *Sustainability Design Master* in their liability, core activities, self-portrayal, and training.

Liability

A project can consist of the following roles: the clients, developers, UX/UI designers, and *Sustainability Design Masters*. Depending on the project, it is possible that all the roles listed are employed by one company, or each role participates in the project through its own company. However, the *Sustainability Design Masters* always act as the interface between the clients, the developers, and UX/UI designers.

Core Activities

Sustainability Design Masters can create a shared understanding of the stakeholders' business, design, and development needs. Accordingly, they know how to acquire insights into business demands, design approaches, and technical challenges.

They guide projects in a holistic and sustainable matter. To this end, they can apply graphical tools to create artifacts in which they present the ideas to approach the project goals from a sustainable perspective. At the beginning of a project *Sustainability Design Masters* work mainly with the involved UX/UI designers and later intensify the collaboration with the developers during the implementation phase. They consult and report to clients throughout the project. The goal of their efforts is to advance the project. Accordingly, *Sustainability Design Masters* should use the artifacts to summarize the current status and create a basis for discussion that makes it possible to decide on the next steps. Furthermore, they create or support the creation of time schedules and monitor their compliance. They are involved in preparing cost estimations and tracking the costs throughout the project.

Self-Portrayal

The *Sustainability Design Masters* are the driving force behind the design of sustainable solutions. Therefore, they need to be open to and interested in new ideas. They do not stick to proven ways of doing things. Nevertheless, they can also adapt ideas to given circumstances.

Furthermore, they accompany the implementation of the design approaches. Accordingly, they should have a sense of reality, which means they can recognize a situation's possibilities and limits. Hence, they can also plan and look for insights to proactively change the plan if necessary. They show entrepreneurial thinking and can anticipate problems that may arise during the implementation. Therefore, they can integrate competing ideas.

Since *Sustainability Design Masters* act as interfaces between the stakeholders, they should have an outgoing and communicative personality. They can explain ideas in a comprehensible way to the various stakeholders. They approach conflicts constructively and can build teams and coalitions.

Training

To pursue these activities, the *Sustainability Design Masters* should know how to acquire a particular knowledge base. This knowledge base enables them to guide the project with a sustainability-driven attitude. During their training, *Sustainability Design Masters* could acquire knowledge and experience in accordance with our set Dim_{SUST}, as presented in Subsection 6.2.3. Using these dimensions, *Sustainability Design Master*s can analyze projects and create solution proposals.

9.3.2 Process

As demonstrated in Figure 28, we suggest perceiving the development process of sustainable software products as two intertwined tracks. The first track is concerned with design activities and starts up front, while the second track, which represents the implementation activities, enters the development process later. The *Sustainability Design Masters* are involved in both tracks. The *Sustainability Design Masters* form with the designers and developers the project team. Based on the client's needs, they contribute to a coherent and sustainable software design and monitor its implementation later. To this end, we prepared a five-level-based guideline. We describe the purpose of each level, the activities the *Sustainability Design Masters* should perform, and artifacts suggestions they can provide. In the following section, we present *Level I – Context, Level II – Design, Level III – Details, Level IV – Details* and *Level V – Monitoring*.

LEVEL I – CONTEXT

Purpose

Clarifying the various stakeholder needs and creating a shared understanding of the project purpose

Activities

1. **Identify** the client's project requirements, using, e.g., interviews, surveys, or workshops applying Dim_{SUST}. This might include discussing the budget, involved stakeholders, the overall timeframe, specific company sustainability goals, and system context. *In case of repetition: Evaluate the identified requirements*.
2. **Define** the system and integration requirements of the future product.
3. **Prepare** the artifacts without or with UX/UI designers, summarizing the insights of activities 1 and 2. The artifacts should serve as a decision foundation.
4. **Provide** the artifacts to the stakeholders.
5. **Discuss** the artifacts and document the results.
6. **Decide** with the client whether to continue to the next level or repeat the current level's activities.

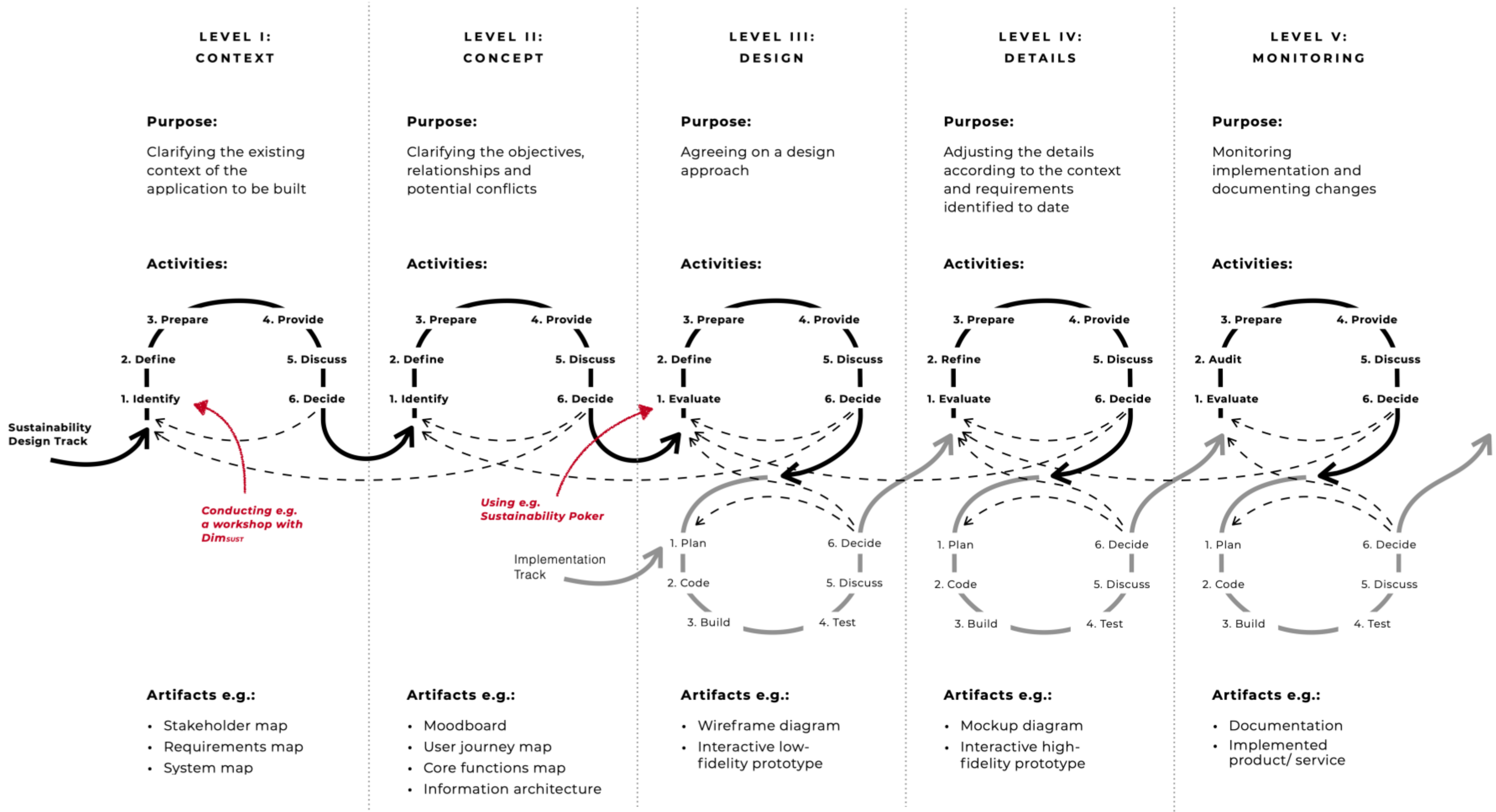


Figure 28: Sustainability-centered design process

Artifacts

Stakeholder Overview. Presents all stakeholders and, if necessary, all the people/institutions who indirectly influence the project course. This overview should represent the responsibilities, contact data, and intersections.

Product Requirement Diagrams. Shows the requirements of the client and their organizational system. These artifacts include requirements and goals that are not per se technical, e.g., integrating particular departments or changing the clients' image. The *Sustainability Design Masters* can visualize the requirements as a Mind-Map or another preferred representation.

System Overview. Creates a shared understanding of the technical environment. Based on this overview, the stakeholders can determine the technical "inventory," e.g., explore which systems, applications, and interfaces already exist and to what extent they have to consider them. Furthermore, stakeholders can gain an understanding of the technical conditions and might discover deficiencies and gaps.

LEVEL II – CONCEPT

Purpose

Clarifying objectives, relationships, and potential conflicts

Activities

1. **Identify** the most crucial problems, interesting insights, and promising suggestions from the previous level.
2. **Define** different solution approaches while considering the identified issues from activity 1.
3. **Prepare** the artifacts with or without UX/UI designers, which present the different solution approaches. The artifacts should serve as a decision foundation.
4. **Provide** the artifacts to the stakeholders.
5. **Discuss** the artifacts with the stakeholders and document the results.
6. **Decide** with the client whether the project team should pursue one or more of the presented solution approaches to continue to the next level. If the client is not satisfied, discuss whether the project team should repeat the current or previous level's activities.

Artifacts

Mood Board. Visualizes the "atmosphere" and the vision of the appearance of the future product. This allows the stakeholders to compare ideas and visions to evaluate whether their opinions diverge.

User Journey Map. Creates an overview of how and when future users will interact with the future product. The stakeholders should use it to assess whether the future product can address and evoke the desired behavior of the users.

Core Function Map. Presents the core functions, which address the primary needs of the users. The stakeholders can base their design and information architecture on these core functions.

Information Architecture. Presents, e.g., as a site map. Stakeholders can use this site map to reveal connections or gaps. Based on the representation of the information architecture, Stakeholders can discuss new ideas and how they can close gaps or create new connections.

LEVEL III – DESIGN

Purpose

Agreeing on a design approach

Activities

1. **Evaluate** whether the identified requirements are still up-to-date or lacking information and adjust them if necessary. Examine whether the chosen solution approaches contradict new emerging requirements or the sustainability of the future product, using, e.g., *Sustainability Poker*.
2. **Define** the chosen solution approaches in more detail, according to the required information depth of the respective artifact.
3. **Prepare** the artifacts with or without UX/UI designers, which present the different solution approaches. The artifacts should serve as a decision foundation.
4. **Provide** the artifacts to the stakeholders.
5. **Discuss** the artifacts with the stakeholders and document the results.
6. **Decide** with the client whether the project team should pursue one of the presented solution approaches to implement a low fidelity prototype or repeat the activities of Level III. The project team can develop a low-fidelity prototype as soon as the stakeholders agree on a solution approach. The stakeholders test the prototype and discuss the test results. Based on the discussion outcome, the stakeholders decide whether the project can proceed to the next level or needs to repeat the activities of the current level or previous levels.

Artifacts

Wireframe Diagram. Visualizes the navigation structure and control elements. Shows the relations of the individual screens to each other and the different states of the product, e.g., error message or waiting time. The illustration describes the purpose of each screen, the features, and how it has considered sustainability dimensions, as exemplified in Figure 29. Based on the wireframes, the stakeholders can evaluate the conceptual functionality and asses if the project is moving in the desired direction.

Interactive Low-Fidelity Prototype. The project team develops an interactive, low-fidelity prototype based on the wireframes. The stakeholders can evaluate the usability and the behavior of the users. If the behavior shown deviates from the desired interaction, the project can take corrective means at an early stage.

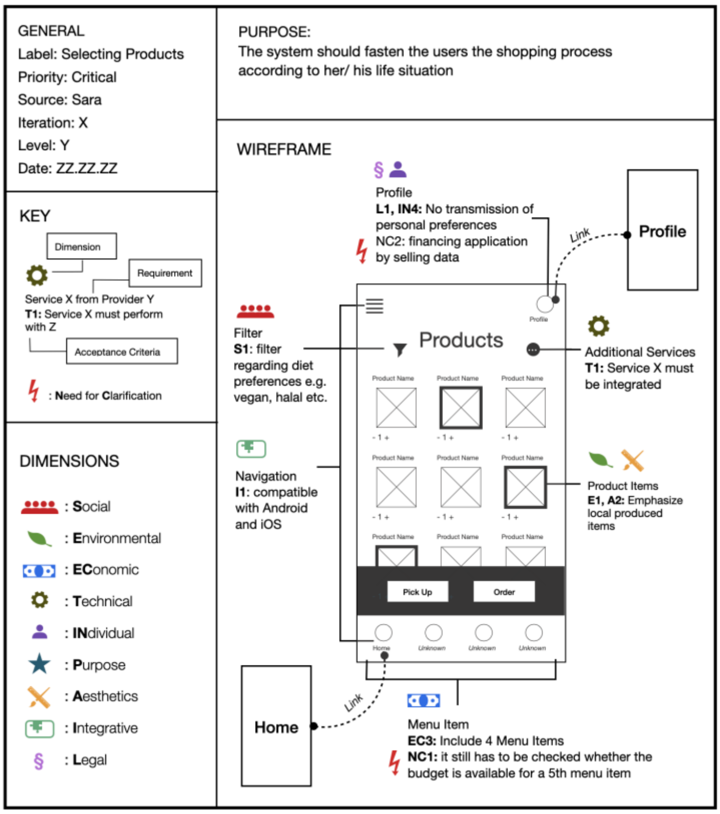


Figure 29: Level III - example wireframe

LEVEL IV: DETAILS

Purpose

Adjusting the details according to the contexts and conditions identified to date

Activities

1. **Evaluate** whether the identified requirements are still up-to-date or lacking information and adjust them if necessary. Examine whether the chosen solution approaches contradict new emerging requirements or the sustainability of the future product.
2. **Refine** the chosen design approach according to the required information depth of the respective artifact.
3. **Prepare** the artifacts with or without UX/UI designers, which present different solution approaches for particular details if necessary. The artifacts should serve as a decision foundation.
4. **Provide** the artifacts to the stakeholders.
5. **Discuss** the artifacts with the stakeholders and document the results.
6. **Decide** with the client whether to continue to the next level or to repeat the current or previous level's activities.

Artifacts

Mock-Up Diagram. Presents a detailed replica of the future product. The developers can use the mock-ups to prepare or, if necessary, adapt their code accordingly. As with the wireframe diagram, the illustration describes the purpose of each screen, the features, and how it has considered sustainability dimensions. The stakeholder can use the mock-ups to discuss the concepts and ideas in detail.

Interactive High-Fidelity Prototype. The project team develops an interactive, high-fidelity prototype based on the mock-ups. This is especially suitable for elaborating and testing novel details that need empirical proof of their impact. The stakeholder can solve many problems in advance through interactive prototypes and mock-ups before investing costly development time.

LEVEL V: MONITORING

Purpose

Monitoring implementation and documenting changes

Activities

1. **Evaluate** whether the identified requirements are still up-to-date or lacking information and adjust them if necessary. Examine whether the chosen solution approaches contradict new emerging requirements or the sustainability of the future product.
2. **Audit** and document the development process and changes.
3. **Prepare** documentation of the implemented product/service, including changes.
4. **Provide** implemented product (parts) to the client.
5. **Discuss** the implemented product (parts) with the stakeholders.
6. **Decide** with the client whether to continue the implementation according to the chosen design or to initiate changes.

Artifacts

Product Documentation. Uses the mock-up diagram and corresponding reference to the source code to document the implemented product. This keeps a record of whether and how the product has changed in the course of the implementation. This documentation is supposed to support future maintenance efforts to capture the entirety and limitations of the product.

9.4 Discussion

The role and activities of *Sustainability Design Masters* require a versatile study program and the desire to pursue sustainability-driven solutions. Consequently, it may be necessary to have years of in-depth training to perform those activities and contribute to creating adequate artifacts. In addition, the interdisciplinary nature of sustainability requires interdisciplinary training. Thus, implementing this training involves many resources, such as time, people, and money. However, it might be challenging to provide sufficient training in today's era of quick training and the need to make people available to the economy as soon as possible, as we can see by the goals of the Bologna Declaration [212]. Nevertheless, we can observe initial attempts to create alternative training programs from initiatives such as *Digital Design* [114].

Similar to our *Sustainability Poker* in Chapter 8 and our set of Dim_{SUST} in Subsection 6.2.3, our proposed framework may be more appropriate for larger projects. Larger projects, however, are likely to have a more significant impact, which should be guided in a sustainable manner regardless. In Chapter 5, we reported that practitioners perceive the topic of sustainability as an additional expense that only makes sense for large projects. In addition, our interviewees believed that implementing sustainability-related aspects must prove financially beneficial. It is, therefore, necessary for us to provide evidence in the future as to whether our framework can lead to sustainable software design and an increase in product sales.

Besides the project size, we prepared the framework for projects that require a graphical user interface. As a result, we provided a guideline that should assist the *Sustainability Design Masters* in using graphical means to present the current stage and advance the project. Using a graphical representation of requirements should enable projects to include stakeholders with less technical expertise, such as users or clients. The framework is, therefore, better suited to accompanying heterogeneous teams.

The *Sustainability Design Masters* can also alleviate the burden of documenting the development process on the developers by guiding the project with artifacts. In this manner, *Sustainability Design Masters* can establish communication bases that technically experienced and inexperienced individuals can use. Moreover, these artifacts can assist future maintenance activities in grasping the basic product structure more quickly. By facilitating the maintenance effort, clients can ensure that their product remains efficient and continues to serve users for as long as possible.

Despite the assumption that our framework implementation requires intensive training and a specific project type and size, we provided this proposal as a contribution to the discussion of whether we need new roles and a new understanding of designing and implementing software products to ensure their sustainability.

The proposal presented here may appear to be a mere addition to current implementation processes. In the future, we need to refine our framework *Sustainability-Centered Design,* so it becomes a given that it is a means to keep different interests together from the first concept idea to the final implementation.

Furthermore, we need to provide evidence that we have to “compose” requirements or explicit design activities to create a holistic product that is sustainable for our society and environment.

9.5 Related Works

In all the strategies that aim to incorporate different perspectives into software development, we can see that not only technical aspects are relevant for software products. One strategy example is *DevOps*, which aims to bring together business interests and technical capabilities. Another popular approach is adapting agile processes to create user-friendly products. However, agile working methods, e.g., *Scrum* or *Extreme Programming*, have the problem that they partly cannot answer what the right product is for the users [177].

According to Schön et al. [177], “These models lack in defining the right kind of product, which fulfills user needs and customer expectations. In order to fill in this gap and to develop products with a good user experience (UX), hybrid development approaches including Human-Centered Design [...] are applied.” Blomkvist [19] takes a similar view, writing, “From the perspective of usability and user-centered design, however, agile methods do not inherently provide the required support to the development process.” Consequently, practitioners and scholars have provided several proposals to address this problem. According to Zhong and Schmiedel [220], low user acceptance is mainly due to poor usability. Applying usability testing, they propose ways to integrate the user perspective into agile work processes to develop robot applications. Glomann [75] proposes the “Human-Centered Agile Workflow (HCAW),” an adaption of agile processes. Like Sy [192], Glomann suggests conducting a so-called *Sprint 0* before stakeholders start planning the implementation. *Sprint 0* aims to prepare a basic concept, including research and creating prototypes. Blomkvist recommends a “Balanced Integration Between Agile Development and User-Centered Design” to compensate for this lack of user-friendly design in software development.

However, the application of user-centered design methods and artifacts is not as straightforward as it might seem. Based on their mapping study, Garcia et al. [70] found that prototypes, user stories, and cards are the most commonly used artifacts. Nevertheless, Lucassen et al. [121] report that many user stories are not well written. To address this issue, they developed a tool capable of automatically identifying quality defects and suggesting improvements. Despite many techniques and methods, little guidance is available regarding which artifact is most appropriate for a particular project stage [182].

Furthermore, there has been little work on how sustainability can be considered procedurally in agile software development. Suggestions on considering sustainability were primarily related to requirements or software attributes [12]. As an example, Lago et al. [112] suggest treating sustainability as a quality attribute. Moreover, Lauenroth et al. [116] state that for sustainable software products to emerge, a new role is needed to be characterized by a new discipline. As a result, the authors propose a concept known as *Digital Design*, which does not see itself as a mere add-on to software development but as a driving and shaping force whose engagement should result in sustainable software products.

9.6 Summary

Current development processes, like agile approaches, are more likely to be technology focused and might not be able to consider sustainability-related aspects. Therefore, we proposed our framework *Sustainability-Centered Design* to foster sustainable software design. We present the background and foundation that we used to prepare the background. Our framework consists of a description of the new role of the *Sustainability Design Master* and a corresponding iterative process. This process includes how projects can organize design and implementation activities into two intertwined tracks. To this end, we suggest a five-level-based guideline that aims to represent the different project progress stages. This guideline describes the purpose of each level, the activities *Sustainability Design Masters* should perform, and suggestions on which artifacts they can provide. We discuss the conditions and limitations of implementing our framework and present related works.

10 Discussion

Albert Schweitzer is credited with the phrase, that *first people build houses, then houses build people* [44]. Nowadays, this statement might also apply to software products as human behavior, societies, and industries are changing due to advances in software technologies [101]. People may use software systems longer than initially intended [159], which can have an unforeseen negative impact. It is, therefore, more crucial than ever to address the issue of software sustainability.

However, the design of sustainable products may require novel ideas, as it cannot rely on previous, potentially harmful approaches. Hence, pursuing sustainable product design can both enable companies to innovate and move us towards a more sustainable future as a society. Today's creation of innovative and sustainable products ranges from the production of analog goods such as clothing [167] and houses [61] to digital goods such as software [203]. Various disciplines and institutions have taken on the topic of sustainability, demonstrating that sustainability is a complex concept that we cannot implement with a single generalizable approach [97]. Therefore, we followed Guldin's [81] interpretation of Schuler's and Görlich's multifactorial theory of innovation [179] to propose an extension of requirements engineering. We aimed to promote sustainable software design by providing this extension proposal. The authors' theory recommends considering the factors *inspiration, communication*, and *perspiration* to enable innovation.

Based on these factors, we identified problems in requirements engineering, such as the risk of neglecting sustainability-related requirements, insufficient communication bases, and missing adequate design frameworks. To address these shortcomings, we draw on insights from psychology, linguistics, software engineering, and building architecture, thus reflecting the interdisciplinary nature of sustainability. According to innovation factors, we provide approaches to identify sustainability-related requirements, communicate their relevance for the software's sustainability, and keep pursuing sustainability throughout the development process.

The design and promotion of sustainable software thus seem to be an interplay of many different factors. We must take responsibility for the products we create and their impact. Because what applies to our analogically created world applies just as much to our digitally created world: "First people shape software, then software shapes people.” In this chapter, we discuss our findings in Section 10.1 and limitations in Section 10.2

10.1 Promoting Sustainable Software Design

10.1.1 Diverse Thinking

We followed the approach of Becker et al. [12], assigning requirements an essential role in designing sustainable software products. Therefore, we have dealt with requirements engineering and its challenges, such as incomplete requirements [140]. We addressed this problem because we assumed it could also affect the identification of sustainability-related requirements, as presented in Section 1.2. Based on the linguistic relativity hypothesis, described in Subsection 2.2.3, we investigated whether we can alleviate incompleteness by using requirements dimensions, reducing the risk that development projects overlook sustainability-related requirements.

To this end, we suggest the requirements dimensions set Dim_{SUST}, as outlined in Subsection 6.2.3, and conducted several experiments, detailed in Chapter 6 and Chapter 7. Our results indicate that providing requirements dimensions, such as Dim_{SUST} or Dim_{ISO}, can guide the identification process. The experiments demonstrated that providing sets of dimensions results in more requirements than their absence. The practical implication of this observation is that we can likely address the recurring problem of missing requirements [140] by utilizing predefined requirements dimensions. Project participants might be enabled to identify specific requirements early, which they would otherwise not have identified until later. Regarding the identification of sustainability-related requirements, the results were not as significant. We observed that our control group performed weaker in identifying sustainability-related requirements than the groups with given sets of requirements dimensions. However, we did not observe such distinct differences between the given sets. Only in terms of ecological-related requirements were the participants able to identify more requirements with our proposed set Dim_{SUST} than the comparison groups. Whereas the participants with Dim_{ISO} that included the software quality dimensions of the standard *ISO 25010:11* were able to identify more technical-related requirements. Part of the reason for this observation might be that language has only an influencing effect after all and not a deterministic one. Another reason might lie in the coding of the identified requirements, resulting in partly moderate kappa values. This shows another complexity of the application of requirements dimensions.

With the term *requirement dimension*, we intended to create the possibility that we can identify requirements in relation to several dimensions. After all, it may well be that requirements can arise through several dimensions or affect several dimensions [183], [13]. However, we had to assign the identified requirements exclusively to one dimension for our statistical analysis. Through this analysis, we were able to evaluate whether we can enable people to take different perspectives and thus succeed in identifying more requirements by providing them with requirements dimensions.

Yet, in the subsequent implementation of the requirements, which may have resulted from the inspiration of multiple dimensions, a categorical abstraction must then occur again so that the computer can process them. This procedure could lead to less suited solutions for the “human world.”

While the “human world” appears in multiple dimensions, the machine world remains two-dimensional for the time being. To this day, computers are based on a binary system and thus rely on us humans to give them commands in “0 or 1” logic. While we can live with ambiguities among us humans, we cannot do so in communication with machines, as we can observe in the numerous measures against ambiguity [134], [195], [18]. It is still challenging to transfer multidimensional requirements into a binary system in such a way that real multidimensional problems can be solved. To meet this challenge, we have proposed our framework, *Sustainability-Centered Design*, in Chapter 9. This framework puts more emphasis on software design to enable people to conceptualize more freely before translating their ideas into written code. In doing so, we want to take advantage of the fact that humans can handle ambiguities as well as clear specifications [73]. A skill that is not yet sufficiently developed, let alone available, in most machines [73].

In summary, we cannot prove that our dimensions and the requirements identified with them lead to sustainable software products, let alone sustainable software innovations. The adjective *sustainable* can, by its very definition, only be attributed without reasonable doubt by retrospectively analyzing a product. Furthermore, we have not discussed whether pursuing economic growth with sustainable innovation might even contradict providing a sustainable future. Future research endeavors are challenged with finding ways to ex-ante increase the probability that a given software product designed today will turn out to have been sustainable and innovative in ex-post evaluation.

At most, our results can be understood as an approach to increase the chance that future software product designs will also become sustainable innovations. One possible influencing factor could be inspiring requirements dimensions that enable people to identify sustainability-related requirements. Nevertheless, we must also ask whether it is desirable for us in this course to require external guidance to consider sustainability. If we do not wish to do so, we must decide what consequences we should impose to enable people to become more independent from external influences.

10.1.2 Interdisciplinary Communication

For the design of potentially innovative [91] and sustainable products [11] communication is necessary. Through communication, stakeholders can create a shared understanding of novel solution approaches for designing sustainable software products. This understanding can produce a collective effort [43] that leads to effective team collaboration while designing and implementing those novel ideas. However, novel ideas have the problem that they cannot rely on proven experience. Moreover, sustainability is still a complex concept with no universally accepted implementation principles [97], [11]. According to Briassoulis [25], we intuitively understand sustainability, but it can also be perceived differently due to cultural differences, as seen with environmental phenomena [76]. In addition, sustainability is also an interdisciplinary concept, as described in Section 2.1. This entails different interests that we need to bring together, making communication essential. The awareness of this issue has even led to the emergence of the designated research field *sustainability communication* [76]. This field distinguishes between internal and external communication [76]. Internal communication refers to all the communication within an organization [76].

In this thesis, we focused on internal communication and created *Sustainability Poker*, as outlined in Chapter 8. We aimed to provide a card-based communication approach for interdisciplinary teams to estimate requirements for their potential relevance and impact on product sustainability. We evaluated our approach in several iterations with practitioners to gain insights for improvement. Our sample included product owners, developers, and UX/UI designers. We observed that participants were able to point out and compare each other's perspectives. In doing so, the participants particularly valued the constructive discussion. They reported that our estimation procedure supported them in creating a shared understanding of potential requirements' impacts on the product's sustainability. This observation aligns with the goal of *sustainability communication* [76].

We derived our communication and estimation procedure from *Planning Poker* [79] and the *delphi method* [69], which uses expert evaluations to asses, e.g., the effort to implement a particular user story. These experts usually have in common that they all come from one discipline. In our case, we gathered experts from different fields and specialties who were not sustainability experts. However, this interdisciplinary composition allowed us to comply with the interdisciplinary concept of sustainability. Even though the method is based on subjective assessments, according to Granger [78], it can at times be more reliable than objective methods such as statistical analysis. However, Granger [78] also points out that objective tools are reliable when we pursue short-term predictions in stable markets and rely on a large amount of data. In contrast, for long-term predictions in unstable environments with little data at hand, subjective estimates may be more successful [78]. Therefore, our estimation procedure can be helpful since the design of sustainable software systems aims to create long-term products, which often enter or even create unstable markets. Nevertheless, our approach is currently time-consuming. Companies will probably need evidence that it can contribute to their competitive advantages. Additionally, external regulations could encourage companies to apply approaches, such as *Sustainability Poker*, to show how they have considered sustainability-related aspects in their software products.

10.1.3 Conversion Challenges

Mohanani et al. [143] report that the relationship between requirements engineering and design is still poorly understood. Furthermore, the belief still prevails that requirements are simply collected and then handed over to the design team, which then develops a suitable system based on them. They criticize viewing requirements engineering as a purely analytical activity and the representations of requirements in written form. They report, "Serious questions regarding how best to record and present desiderata remain unanswered." Following this statement, we have proposed our *Sustainability-Centered Design* framework in Chapter 9, which addresses this very relationship between requirements engineering and design. We described a process of how software development projects could organize design and implementation activities in two intertwined tracks. Furthermore, we introduced the new role of the *Sustainability Design Master* since people are still crucial for designing sustainable and innovative products [137].

The design of sustainable software products requires bringing together a wide variety of disciplines, as described in Chapter 2. The confluence of these different perspectives can give rise to various ideas or requirements that need to be refined and improved. However, trying out new ideas can inhibit conflicting demands and external time pressure. Therefore, we propose the role of the *Sustainability Design Master* to guide teams through these challenges. According to Gupta and Singhal, "innovation-based companies have learned how to manage, motivate and reward people" [83]. Hence, this leading role should combine analytical- and design-related activities with social skills. Based on the description of an innovative person by Schuler and Görlich [179] and the concept of leadership from Hill et al. [93] as well as the field of building architecture [150], the *Sustainability Design Master* should mind the future product as a whole as well as its individual parts. The role should ensure sufficient time for the design of new ideas while also driving the progress of the creative effort [93]. Consequently, our framework incorporates activities such as testing and designing new ideas while striving to move forward. We propose organizing iterative development procedures into five levels. Using this iterative level approach, we aimed to include the necessary design loops while emphasizing that the results of the loops should lead to project progress rather than stagnation. Therefore, we suggested several artifact recommendations that *Sustainable Design Masters* may use to represent the project status and provide their stakeholders with a solid basis for discussing the next steps. Furthermore, the resulting prototypes and designs can serve as documentation of the software development process, somewhat relieving the developers of the documentation burden.

Mohanani et al. [143] report that the representation of requirements, or as they call it, "desiderata," is critical to how the development team implements them. The authors reported that "templated requirements specification (TRS)" limits critical thinking, which they consider an essential part of creativity. They describe TRS as "requirements specification written in a specific syntactic structure using a restricted (controlled) natural language [8], in this case, for example, 'The system shall facilitate diet planning'." The authors, however, did not explore whether critical thinking is also restricted when the same people are both identifying requirements and creating the software designs. Our framework recommends the combination of requirements engineering and design by proposing mainly graphical representation methods, which include written information. This results in a range of tasks that need different skills that probably cannot be covered by the current profiles of designers, developers, or product owners. For *Sustainable Design Masters* to take over these different tasks and activities, we need to provide them a presumably comprehensive and time-consuming training. However, by establishing this role, we can introduce a leadership position that can contribute to designing and implementing potentially innovative products [111] while being careful to maintain a sustainability focus.

In summary, our framework aims to open up the possibility of considering a new idea for a sustainable product in its individual components and entirety. For the sake of maintenance, it will probably still be advantageous to think of software in modular terms, but only a holistic design approach can ensure that the individual components become a coherent whole. Although our proposal might appear to be an addition to current development approaches, we also believe that we need to do more than simply extend current development processes. To design sustainable software products, we must rethink the balance of power. However, this change requires us to resist putting ideas, solutions, or problems into hierarchical systems. Our responsibility should instead be to accept their existence and to consider them in relation to one another and the world at large. By adopting this approach, we can value, e.g., different disciplines equally rather than equalize or prioritize them. Similar to ensuring the operation of a clockwork, each gearwheel is equally important regardless of its size.

10.2 Limitations and Threats to Validity

The following section mainly summarizes the limitations and threats to the validity of our empirical studies, starting with our experiments, as presented in Chapter 6 and Chapter 7. We randomized the order of the requirements dimensions in our experiments to reduce the effects related to their sequence. We compensated the participants of the experiments with the opportunity to win a retail voucher in a raffle or credit points for their study if they were students. Participants also had to confirm that they had not participated in the study before. Given the time required to complete the study and that repeat participation does not result in additional credit points or higher chances of winning, we can assume that each participant has only participated once. The interviewees of our study in Chapter 5 were self-motivated, and we did not offer them any compensation. This indicates an active interest in the topic of sustainability and might therefore not be representative of all practitioners. We compensated the focus group participants, as reported in Chapter 8, with the opportunity to choose an initiative to which we should donate.

We used pre-studies to gather feedback on whether the participants comprehended the assignments and questions correctly. This enabled us to improve the design of our primary studies. In addition, we could test our coding guidelines by analyzing the data from the trial studies and applied researcher triangulation. Even though we carried out the qualitative assessment in pairs, we cannot exclude that other researchers would have coded specific data differently. The moderate results of the kappa test of the experiments indicate that an interpretation is not so straightforward.

For the experiments, we asked the participants to complete a trial task before we forwarded them to the main task. Through this, we enabled participants to focus on the primary task rather than spend time learning how the primary task works. Regarding the interviews, as reported in Chapter 5, and our focus groups, as described in Chapter 8, we sent sample questions and an overview of our research goals to potential interviewees and discussants. Before and at the end of each interview or discussion, we allowed the participants to ask clarifying questions to avoid misunderstandings. Nevertheless, we cannot exclude that there might have been misunderstandings.

The participants of Experiment II, as described in Chapter 7, were all students of the same study program. This might limit the external validity of the results to some extent since we assume that the participants have a similar background. However, it is common to use students as subjects in requirements engineering and software engineering research since it is more difficult to recruit professionals for such studies, especially if participation requires a particular completion time. We tried to mitigate this potential threat by recruiting participants with different occupations and fields of expertise for Experiment I. Furthermore, our experiments may not be easily replicable because only two participants conducted the study in English while the other 68 participants did so in German. Using other languages may lead to different results. This is also the case for the semi-guided interviews and focus group discussions, which we only conducted in German. Furthermore, most interviewees and discussants were neither requirements engineering nor software sustainability experts. They only had expertise in their respective fields. Additionally, our sample sizes are relatively small. Therefore, our quantitative reports should serve to derive further hypotheses rather than prove theories.

Furthermore, the small sample size bears the risk that our data is not normally distributed, which ideally should be the case for a regression analysis to be applied. However, Wooldridge [219] reports that regression analysis is robust enough under conditions that are not ideal. Furthermore, we included only a limited number of individual factors in Experiment I, as listed in Table 31. Accordingly, there may be other factors that could have influenced the dependent variables.

We conducted all the studies online. Regarding the experiments, the participants had to confirm that they would avoid distractions and not seek help before being forwarded to the study. Therefore, we can assume that our participants carried out the study correctly, but we cannot completely exclude any external influences.

We prepared our framework, *Sustainability-Centered Design*, as presented in Chapter 9, as a guideline that recommends representation methods of the product to be built according to the project progress. Using these representations, developers can form implementation strategies. However, this does not mean that developers cannot also think about social or ecological requirements. By relieving developers from documentation duties, we can enable them to dive deep into details, which in turn can lead to more technically sound and low-maintenance products. Since we have not implemented this framework in practice, future research has to evaluate its effect.

11 Conclusion

This chapter summarizes the thesis' contributions in Section 11.1 and discusses potential endeavors for future work Section 11.2.

11.1 Contribution Summary

Over the past years, consumer demand has increased, and political pressure has intensified for sustainable products. Consequently, software companies may have a competitive advantage in designing and offering sustainable products. Moreover, sustainable software design might provide us with an opportunity to leverage the already prevalent influence of software. Therefore, we may be able to ensure we do not compromise "the ability of future generations to meet their own needs" [198]. To this end, we propose extending requirements engineering to facilitate sustainable software design. Since requirements play an integral role in software design, we focused our efforts on requirements engineering. Their influence can determine whether we can consider a software product sustainable.

However, some problems or goals may not yet have sustainable solutions. Hence, software companies may be required to come up with novel ideas in order to design sustainable software products. For these sustainable products to have a meaningful impact, they must be widely adopted, just as innovative products are. Considering that innovations also arise from novel ideas, we have investigated which external drivers companies are exposed to and internal factors that they can influence to promote the emergence of innovative products.

Therefore, we examined in Chapter 2 the concept of sustainability and potential factors that may influence the emergence of innovations. From this foundation, we learned that sustainability is a century-old and ever-changing concept whose definition varies depending on the discipline. In light of the interdisciplinarity of sustainability, we examined how building architecture, requirements engineering, and software engineering have addressed sustainability in the past. By studying theories about innovation, we gained suggestions on how to use these interdisciplinary insights to foster sustainable software design. We learned that future innovation waves would be driven by the need to increase productivity and provide sustainable solutions. In order to benefit from these external drivers, we derived from Guldin's interpretation of Schuler's and Görlich's [179] theory that companies can influence the internal factors *inspiration, communication*, and *perspiration* to foster sustainable software design. However, before preparing our extension proposal on this foundation, we wanted to gain a preliminary understanding of current requirements engineering approaches and practitioners' perspectives on sustainability.

To this end, we conducted a design thinking workshop, as reported in Chapter 3. The workshop revealed that design thinking could generate new ideas but did not make participants consider the issue of sustainability. The linguistic relativity hypothesis, described in Subsection 2.2.3, provides a possible explanation for this observation. The hypothesis states that language influences our perception and thoughts. Building on the workshop observations and linguistic relativity hypothesis, we conducted a series of studies to explore the reasons behind the lack of inspiration regarding sustainability issues. In Chapter 4, we conducted a comparative analysis of the roles of product owners, requirements engineers, and building architects to identify potential synergies. Through this comparison, we identified several differences and similarities between the roles. Based on the result, we have derived hypotheses on how to adapt requirements engineering approaches to promote sustainable software design.

To explore the industry's perspective on software sustainability and on applying requirements dimensions to design sustainable software, we conducted semi-guided interviews with practitioners. In Chapter 5, we report that practitioners perceived our set of requirements dimensions as a promoting way to guide the identification process and appreciated its multifaceted approach. They reported that daily work routines do not always allow time to consider issues from multiple angles, which sets as ours could help. However, participants voiced that for this set to be used in a supportive manner, we have to take other factors into account, such as working-related aspects and essential prerequisites. Working-related aspects such as time pressure or profit orientation might hinder the application of our set. Furthermore, a prerequisite for such discussions, a common understanding of sustainable products would be helpful. With that and additional public support, lawmakers need to pass legislation that incentivizes companies and clients to act sustainably.

Our foundation and study insights from Chapter 3 to Chapter 5 enabled us to prepare the following three proposals on how to extend requirements engineering to foster sustainable software design. In Subsection 11.1.1, we summarize our observations regarding the potential of dimensions to inspire the identification of requirements. In Subsection 11.1.2, we describe how *Sustainability Poker* can facilitate communication within interdisciplinary teams toward a shared understanding of software requirements and their relationship to sustainability. In Subsection 11.1.3, we report how development projects can apply our framework *Sustainability-Centered Design* to consider sustainability throughout the development process.

11.1.1 Inspiration: Identifying More Requirements

Dim$_{\text{SUST}}$, a Set of Dimensions for Identifying Sustainability-Related Requirements. Besides the lacking consideration of sustainable aspects, another pressing problem in software engineering is incomplete requirements [140]. Based on the linguistic relativity hypothesis, we assumed that the "requirement language", or rather the requirements dimensions software developers or designers use, might restrict their inspiration possibilities. Our foundation in Chapter 2 and our analysis in Chapter 4 brought to our attention that the most well-known sets of requirements dimensions consist of either two dimensions, *functional* and *non-functional* requirements, or of the three dimensions *functional-, quality* requirements, and *constraints*. In contrast, building architects who aim to develop long-lasting and sustainable products work with a set of eight dimensions. Following these insights, we conducted an exploratory study presented in Chapter 6 to investigate the influence of different sets of requirements dimensions on identifying requirements. To this end, we suggested our own set of requirements dimensions [163]. In our exploratory study, we investigated the impact of personal factors such as age, work experience, or gender on identifying requirements. Furthermore, we aimed to evaluate our research design for the following main study presented in Chapter 7. We found that the explicit application of the requirements dimensions used in this study significantly affected the number of identified requirements compared to not using any requirements dimensions. This observation indicates that stakeholders might identify more requirements with requirements dimensions than without dimensions. We observed that applying our proposed set of requirements dimensions cannot guarantee the identification of all relevant sustainability-related requirements, but it increases the likelihood compared to not using any dimensions.

11.1.2 Communication: Creating a Shared Understanding

Sustainability Poker, a Requirements Estimation Approach for Facilitating Communication Within Interdisciplinary Teams. In Chapter 6 and Chapter 7, we suggested a set of requirements dimensions that might increase the likelihood of the future product being sustainable. However, according to our underlying innovation-promoting factors, it is not enough to identify relevant and inspiring requirements. We also must communicate them effectively, so they are not only seen and understood by the people who have identified them. If other people understand the relevance and influence of the requirements, they might be more motivated to support their implementation. To this end, we present our card-based communication procedure, *Sustainability Poker*, in Chapter 8. This procedure includes cards based on our set Dim_{SUST} and questions related to these dimensions. By answering these questions, we aimed to assist stakeholders in estimating and exchanging their thoughts regarding the impact a requirement might have on the sustainability of a software product. Furthermore, our procedure provides quantitative figures to compare requirements and dimensions that might conflict with each other. The procedure's results should help project teams or higher-level roles to make decisions that improve the product's sustainability.

11.1.3 Perspiration: Guidance Towards Sustainability

Sustainability-Centered Design, a Framework for Considering Sustainability-Related Requirements Throughout a Development Project. The development of new products can take a long time due to the different interests of the stakeholders that need to be involved. Depending on the stakeholder configuration, various interests may prevail more than others. To develop a product sustainably and holistically while surrounded by those diverse interests, we propose the framework *Sustainability-Centered Design* in Chapter 9. This framework includes the description of the *Sustainability Design Master* role and a corresponding guideline on how this new role can promote sustainable software design. Just as the product owners are responsible for the business interests, the developers for the technical implementation, and the designers for the product experience, the *Sustainability Design Masters* are responsible for ensuring that the product is heading in a sustainable direction. To achieve that, the *Sustainability Design Masters* ensure that all interests are considered sufficiently according to the project circumstances and their sustainability. The guideline addresses the challenge of projects having different widths and depths of information depending on the project's progress. Together with the other stakeholders, the *Sustainability Design Masters* are responsible for creating artifacts according to the project's progress and evaluating whether the product is developing in a sustainable direction. Our guideline recommends which artifacts *Sustainability Design Masters* can create throughout the project to advance the project's progress. By accompanying the project's process with artifacts, the *Sustainability Design Masters* can also relieve the developers of some documentation activities. Stakeholders can use these artifacts as a means of communication and documentation for later maintenance efforts to keep the product usable for as long as possible.

11.2 Future Work

This section gives an outlook on future work. We suggest further research endeavors in Subsection 11.2.1 and potential improvements in Subsection 11.2.2.

11.2.1 Research

In order to promote sustainable software design, we propose an extension of requirements engineering. To this end, we examined the relationship between applying requirements dimensions and requirements identification, reported in Chapter 6 and Chapter 7. Future studies could investigate the way development teams address the identified requirements. For example, future studies could evaluate whether software designs differ if one team does the identification, design, and implementation, and another only receives the identified requirements. However, in practice, it will probably be unavoidable to divide tasks. Accordingly, examining the relationship between requirements engineering, design, and implementation, is particularly important.

An influencing factor in this relationship could be the representation of requirements. According to Mohanani et al. [143], a strong pre-structuring and prioritization of requirements can limit creativity. Since we assumed an influence of representation on the implementation, we proposed our framework, *Sustainability-Centered Design*, in Chapter 9, which suggests representations according to the project progress. However, future studies need to investigate which representation is well suited for implementing requirements. Furthermore, future studies could investigate the difference it could make if stakeholders represent requirements according to their priority or specific topics derived from a set of requirements dimensions, such as our Dim_{SUST}. Moreover, future studies could explore how we can extend existing tools with our proposed set Dim_{SUST}. For example, we could add *sustainability epics* or tags in issue tracking systems such as Jira. Another possibility could be structuring meetings or prioritization procedures according to these dimensions.

We based our framework on real-world approaches, but future research needs to evaluate its possible effects. Since the complete implementation of our framework would require much training upfront, future studies could test only some parts of our framework, such as using the representation methods. Future studies could also investigate whether the representation methods could possibly save implementation time and increase user satisfaction. Saved time reduces the consumption of resources, and increased user satisfaction makes it more likely that the product will be used and spread in the long term.

Our card-based procedure *Sustainability Poker* aims to assist stakeholders in estimating a requirement's impact on a software product's sustainability and communicating their estimates among them. However, we also see the potential to apply the card set for other purposes. For example, future studies could investigate whether our method support stakeholders to empathize with other disciplines and develop an understanding of their challenges. Throughout our study from Chapter 8, we refined the granularity of the cards. We learned that smaller cards are more appropriate for our estimation procedure. However, future studies could explore whether stakeholders might require a larger granularity of cards like in our first iterations to identify requirements or create new ideas. Furthermore, stakeholders can add insights from our proposed estimation procedure to requirements specifications or software architecture documents.

11.2.2 Improvements

Our set Dim_{SUST} could be further adjusted, for example, by combining them with the quality dimensions Dim_{ISO}. Some of our participants in Chapter 5 reported that the technical dimension of Dim_{SUST} was too broad. Since there might be no linear relation between the number of dimensions and identified requirements, future work would have to examine the maximum number of dimensions stakeholders can manage until they reach a certain saturation. Furthermore, future work could investigate whether it might be advantageous to restructure the dimensions. In our study from Chapter 8, we noticed that some dimensions might have intrinsic and others extrinsic influences. For example, the dimension ecological is suitable to think about which requirements might support the ecological effect of a product. Meanwhile, extrinsic dimensions like the legal dimension would directly provide the requirements the product must include. If restructuring the dimensions, future work should make sure to avoid hierarchization, which could lead to the neglect of specific dimensions.

Regarding our *Sustainability-Poker*, participants asked for the possibility of giving more weight to specific dimensions in order to customize them. This weighting, however, would come with the risk of companies focusing on specific dimensions in such a way that the outcome is more profit-maximizing than sustainable. Nevertheless, we need to consider the companies' individual needs to promote our procedure's adaptation. Future work should pursue an approach that will not allow a dimension to fall below a minimum value while considering these individual needs. The underlying calculation model should assure that stakeholders will not neglect dimensions like the social dimension because they might not contribute to a profit-maximizing outcome.

APPENDENCIES

Appendix A – Interview Guideline

Introduction

- Facilitators name, age, motivation
- Summary of the general conditions (duration, anonymizing transcript)
- Asking for permission to record

Working Experiences

- What is your job title?
- What are your typical tasks?
- How long have you been working in your profession?
- Can you roughly estimate how many software applications you have worked on during this time?
- What does your company do?
  - How many employees does your company have?
  - How long have you been working at your current company?
- Do you work in a team? If yes,
  - How many members does your team have?
  - What role do you have in your team?
  - What are your responsibilities?
  - What are the responsibilities of your team members?
  - How do you work with your team? What is your process or framework?
- Do you work directly with the customer? If yes,
  - How much contact do you have with your customers?
  - Can you describe the relationship to your customers?
  - What do you expect from your customers?
  - How do you work with your customers? What is your process or framework?

Requirements Engineering

- What role do requirements engineering play in your work?
- How do you identify requirements?
- Who is involved in the identification?
- How do you communicate requirements?
- What else comes to mind when you think of software requirements?
- Are you aware of any software requirement types (or dimensions)?
- What role do these requirement types (or dimensions) play in your projects?
- How do you have learned about these types (or dimensions)?

Software Sustainability

- How do you define sustainability?
- How would you describe sustainable software?
- How long do you think software applications should remain in use?

- How old is the oldest application you have ever worked on?
    - Would you describe the software as sustainable?
    - If yes/no, why?
- What was the shortest runtime of an application you can remember?
- How relevant do you find the topic of sustainability to software development?
- What is the current importance of sustainability in your software projects?
    - If yes, how is it implemented in your projects?
    - If none, why not? Should it play a role?
- How could the importance be increased?

Introduction of the dimensions set

- What do you think about these requirements dimensions?
- What do you think about their relevance to sustainable software design?
- Which of the sets do you find suitable for sustainable software design and why?
- Which set would you choose to identify requirements with users or customers for sustainable software design?
- Which prerequisites do we need to design sustainable software?

Personal Details

- How old are you?
- Would you consider yourself to be of a particular gender? Female, Male, Diverse, Not Specified
- What is your highest education degree?
- What did you study, or what training did you complete?
- How important is sustainability to you in your personal life? Scale of 1-5 (unimportant - somewhat unimportant - neutral - somewhat important - important)
- Are you involved in a socially relevant (e.g., social, political, or ecological) association, project, group, or initiative?

Outro

- Remaining Questions of the participants
- Conclusion and thanking for participation

Appendix B – Study Questionnaire

<table>
<tr><th>EXPLANATION</th><th>RESPONSE OPTIONS</th></tr>
<tr><td colspan="2">Introduction</td></tr>
<tr><td>Description of the study purpose, the participation conditions, the estimated completion time, and the data protection guidelines, no distraction, opportunity to take part in a raffle</td><td></td></tr>
<tr><td colspan="2">Questionnaire - Part I</td></tr>
<tr><td>Please select an occupation</td><td>• Pupil, student, apprentice
• Professional (employee, worker, self-employed)
• Other</td></tr>
<tr><td>Please select the sector of your education or main occupation</td><td>• Informatics or related to IT
• Other</td></tr>
<tr><td>How much professional experience in the software development field do you have?</td><td>• None
• less than 1 year
• 1-5 years
• 6-10 years
• More than 10 years</td></tr>
<tr><td colspan="2">Transition</td></tr>
<tr><td>Each assignment consists of a trial task and a main task.
YOUR ROLE
Imagine you are part of a software development team. Your task in this team is to name software requirements for new smartphone apps.
WHAT ARE SOFTWARE REQUIREMENTS?
Software requirements comprise wishes and needs of everyone participating in a project, being affected by, or interested in it, as well as the conditions and external restrictions which are given by the planned context of use of the app. A software requirement can be written as follows: ‘In rough sea, the Radar System shall detect targets at ranges up to 100 nautical miles.’
REQUEST
Please work on the tasks alone and without using additional help. There are no right or wrong answers.
Click ‘Next’ to read the introduction of the trial task.</td><td>-</td></tr>
<tr><td colspan="2">Assignments Group Dim_{NO}</td></tr>
<tr><td>ASSIGNMENT - Trial task - Introduction
By working on the following trial task, you can get to know the task procedure. Your answers to the trial task will not be included in the evaluation of the main task. For the following trial task your team is expected to develop a fitness app. This app is supposed to enable users to document their physical activities and at the same time motivate them to do more sports. Your task is to name as many of the app's software requirements as possible. You will have 2 minutes to work on this task. Write down each software requirement as a separate bullet point.
Click ‘Next’ to start the timer and begin the practice task.</td><td>-</td></tr>
<tr><td>ASSIGNMENT - Trial task
Which software requirements should the fitness app meet?
Software requirements comprise wishes and needs of everyone participating in a project, being affected by, or interested in it, as well as the conditions and external restrictions which are given by the planned context of use of the app.
Example: ‘- the app should be able to be used for many different sports’</td><td>Free text – bullet points</td></tr>
</table>

ASSIGNMENT - Main task By completing the trial task, you had the opportunity to try out the task's procedure. Now the main task begins, which means the answers to the following tasks will be included in the evaluation. For the main task your team is asked to develop another app. This time the app is meant to support grocery shopping in the supermarket. Your task is once again to name as many of the app's software requirements as possible. You will have 18 minutes to work on this task. Write down each software requirement as a separate bullet point. Click 'Next' to start the timer and begin with the main task.	-
ASSIGNMENT - Main task Which software requirements should the grocery shopping app meet? Software requirements comprise wishes and needs of everyone participating in a project, being affected by, or interested in it, as well as the conditions and external restrictions which are given by the planned context of use of the app. Example: '- the app should speed up the shopping process'	Free text – bullet points
Assignments Group Dim$_{ISO}$ – Trial Task	
ASSIGNMENT - Trial task By working on the following trial task, you can get to know the task procedure. Your answers to the trial task will not be included in the evaluation of the main task. For the following trial task your team is expected to develop a fitness app. This app is supposed to enable users to document their physical activities and at the same time motivate them to do more sports. Your task is to name as many of the app's software requirements as possible on the basis of 8 dimensions. The 8 dimensions are: 8 REQUIREMENTS DIMENSIONS Functional suitability: This dimension represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. Performance efficiency: This dimension represents the performance relative to the amount of resources used under stated conditions. Compatibility: This dimension describes the degree to which a product, system or component can exchange information with other products, systems, or components, and/or perform its required functions while sharing the same hardware or software environment. Usability: This dimension describes the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. Reliability: This dimension describes the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. Security: This dimension describes the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. Maintainability: This dimension describes the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. Portability: This dimension describes the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. PROCEDURE For the following practice task you will only be given ONE of the dimensions. You will then be tasked to write as many software requirements as possible. You will have two minutes to do so. Write down each software requirement as a separate bullet point. Click 'Next' to start the timer and begin with the warm-up task.	-
ASSIGNMENT - Trial task: Dimension - Functional Suitability Which software requirements regarding the functional suitability should the fitness app meet? This dimension represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. Example: '- The app should enable the creation of a training plan'	Free text – bullet points

ASSIGNMENT - Trial task: Dimension - Performance Efficiency Which software requirements regarding the performance efficiency should the fitness app meet? This dimension represents the performance relative to the amount of resources used under stated conditions. Example: '- The app should not take up much hard drive space'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Compatibility Which software requirements regarding the compatibility should the fitness app meet? This dimension describes the degree to which a product, system or component can exchange information with other products, systems, or components, and/or perform its required functions while sharing the same hardware or software environment. Example: '- The app should be able to import data from a smartwatch'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Usability Which software requirements regarding the usability should the fitness app meet? This dimension describes the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. Example: '- important functionality should be able to be controlled from the phone's lock screen'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Reliability Which software requirements regarding the reliability should the fitness app meet? This dimension describes the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. Example: '- important functionality should be usable without internet'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Security Which software requirements regarding the security should the fitness app meet? This dimension describes the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. Example: '- the app should protect the users' private health information'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Maintainability Which software requirements regarding the maintainability should the fitness app meet? This dimension describes the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. Example: '- the app should be built to allow addition of new features'	Free text – bullet points
ASSIGNMENT - Trial task: Dimension - Portability Which software requirements regarding the portability should the fitness app meet? This dimension describes the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. Example: '- the app should work on iOS and Android'	Free text – bullet points
Assignments Group Dim_{ISO} – Main Task	
ASSIGNMENT - Main task - Introduction By completing the trial task, you had the opportunity to try out the task's procedure. Now the main task begins, which means the answers to the following tasks will be included in the evaluation. For the main task your team is asked to develop another app. This time the app is meant to support grocery shopping in the supermarket. Your task is once again to name as many of the app's software requirements as possible, this time for ALL eight dimensions. You will have 2.25 minutes per dimension to write down your ideas. Write down each software requirement as a separate bullet point. Click 'Next' to start the timer and begin with the main task.	-

ASSIGNMENT - Main task: Dimension - Usability Which software requirements regarding the usability should the grocery shopping app meet? This dimension describes the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. Example: '- important functionality should be able to be controlled from the phone's lock screen'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Portability Which software requirements regarding portability should the grocery shopping app meet? This dimension describes the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. Example: '- The app should work on iOS and Android'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Security Which software requirements regarding security should the grocery shopping app meet? This dimension describes the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. Example: '- the app should protect the users' payment information'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Maintainability Which software requirements regarding maintainability should the grocery shopping app meet? This dimension describes the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. Example: '- the app should be built to allow addition of new features'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Efficiency Which software requirements regarding performance efficiency should the grocery shopping app meet? This dimension represents the performance relative to the amount of resources used under stated conditions. Example: '- the app should not take up much hard drive space'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Functional suitability Which software requirements regarding functional suitability should the grocery shopping app meet? This dimension represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. Example: '- the app should remind me about used up supplies'	Free text – Bullet Points
ASSIGNMENT - Main task: Dimension - Reliability Which software requirements regarding reliability should the grocery shopping app meet? This dimension describes the degree to which a system, product or component performs specified functions under specified conditions for a specified period. Example: '-important functionality should work even without internet access'	Free text – bullet points
ASSIGNMENT - Main task: Dimension - Compatibility Which software requirements regarding the compatibility should the grocery shopping app meet? This dimension describes the degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment. Example: '- the app should be able to import products from online recipes'	Free text – bullet points
Assignments Group Dim$_{SUST}$ – Trial Task	
ASSIGNMENT - Trial task By working on the following trial task, you can get to know the task procedure. Your answers to the trial task will not be included in the evaluation of the main task. For the following trial task your team is expected to develop a fitness app. This app is supposed to enable users to document their physical activities and at the same time motivate them to do more sports.	-

Your task is to name as many of the app's software requirements as possible on the basis of 9 dimensions. The 9 dimensions are: 9 REQUIREMENTS DIMENSIONS Design-aesthetic: This dimension includes requirements that ensure a pleasing and beneficial usage for the entire life of a product. Individual: This dimension refers to the individual human and their opportunities to exercise their freedom, agency, and human rights. Integrative: This dimension includes requirements that regard the integration into existing systems, processes, organizations, contexts etc. Ecological: This dimension includes requirements that are related to the protection of the global and local ecosystem and the saving of natural resources. Economic dimension: This dimension includes the minimization of life cycle costs, improvement of economic efficiency and protection of capital and product value. Legal: This dimension includes the legal requirements and standards which are given by the usage, official regulations, company, or domain. Social: This dimension includes requirements which are concerned with interhuman relationships between individuals or groups, and which support a structure of trust and communication, as well as enabling the balance between conflicting interests. Technical: This dimension includes requirements that refer to the maintenance, evolution, resilience, and the changeableness of artificial systems, such as soft- and hardware. Purpose: This dimension includes requirements that enable an application to fulfil its purpose. PROCEDURE For the following trial task, you will only be given ONE of the dimensions. You will then be tasked to write as many software requirements as possible. You will have 2 minutes to do so. Write down each software requirement as a separate bullet point. Click 'Next' to start the timer and begin with the warm-up task.	
ASSIGNMENT - Trial task: Design-Aesthetic Dimension Which design-aesthetic software requirements should the fitness app meet? This dimension includes requirements that ensure a pleasing and beneficial usage for the entire life of a product. Example: '- the app should graphically show training results'	Free text – bullet points
ASSIGNMENT - Trial task: Individual Dimension Which individual software requirements should the fitness app meet? This dimension refers to the individual human and their opportunities to exercise their freedom, agency, and human rights. Example: '- the app should offer individually customizable training programs'	Free text – bullet points
ASSIGNMENT - Trial task: Integrative Dimension Which integrative software requirements should the fitness app meet? This dimension includes requirements that regard the integration into existing systems, processes, organizations, contexts etc. Example: '- the app should synchronize with my sports equipment'	Free text – bullet points
ASSIGNMENT - Trial task: Ecological Dimension Which ecological software requirements should the fitness app meet? This dimension includes requirements that are related to the protection of the global and local ecosystem and the saving of natural resources. Example: '- the app should give tips for saving resources during sports'	Free text – bullet points
ASSIGNMENT - Trial task: Economic Dimension Which economic software requirements should the fitness app meet? This dimension includes the minimization of life cycle costs, improvement of economic efficiency and protection of capital and product value. Example: '- the app should be financed by ads'	Free text – bullet points
ASSIGNMENT - Trial task: Legal Dimension Which legal software requirements should the fitness app meet? This dimension includes the legal requirements and standards which are given by the usage, official regulations, company, or domain. Example: '- the app should keep personal data confidential'	Free text – bullet points

ASSIGNMENT - Trial task: Social Dimension Which social software requirements should the fitness app meet? This dimension includes requirements that cover relationships between individuals and groups, their structures of trust and communication and the balance between conflicting interests. Example: '- the app should compare the user's training performance with friends or acquaintances'	Free text – bullet points
ASSIGNMENT - Trial task: Technical Dimension Which technical software requirements should the fitness app meet? This dimension includes requirements that refer to the maintenance, evolution, resilience, and the changeableness of artificial systems, such as soft- and hardware. Example: '- the app should be able to save training data in a cloud'	Free text – bullet points
ASSIGNMENT - Trial task: Purpose Dimension Which software requirements need to be met for the fitness app to fulfil its purpose? This dimension includes requirements that enable an application to fulfil its purpose. Example: '- the app should have push notifications as reminder to train'	Free text – bullet points
Assignments Group Dim_{SUST} – Main Task	
ASSIGNMENT - Main task: Design-aesthetic dimension Which design-aesthetic software requirements should the grocery shopping app meet? This dimension includes requirements that ensure a pleasing and beneficial usage for the entire life of a product. Example: '- the app should show the groceries as icons'	Free text – bullet points
ASSIGNMENT - Main task: Individual dimension Which individual software requirements should the grocery shopping app meet? This dimension refers to the individual human and their opportunities to exercise their freedom, agency, and human rights. Example: '- the app should prevent external influence on the purchase decision'	Free text – bullet points
ASSIGNMENT - Main task: Integrative dimension Which integrative software requirements should the grocery shopping app meet? This dimension includes requirements that regard the integration into existing systems, processes, organizations, contexts etc. Example: '- the app should adapt to the visited supermarket'	Free text – bullet points
ASSIGNMENT - Main task: Ecological dimension Which ecological software requirements should the grocery shopping app meet? This dimension includes requirements that are related to the protection of the global and local ecosystem and the saving of natural resources. Example: '- the app should be as energy saving as possible'	Free text – bullet points
ASSIGNMENT - Main task: Economic dimension Which economic software requirements should the grocery shopping app meet? This dimension includes the minimization of life cycle costs, improvement of economic efficiency and protection of capital and product value. Example: '- the app should be able to be bought with a one-time payment'	Free text – bullet points
ASSIGNMENT - Main task: Legal dimension Which legal software requirements should the grocery shopping app meet? This dimension includes the legal requirements and standards which are given by the usage, official regulations, company, or domain. Example: '- the customers' purchase history must be confidential'	Free text – bullet points
ASSIGNMENT - Main task: Social dimension Which social software requirements should the grocery shopping app meet? This dimension includes requirements which are concerned with interhuman relationships between individuals or groups and which sup- port a structure of trust and communication, as well as enabling the balance between conflicting interests. Example: '- the app should enable sharing shopping lists with friends'	Free text – bullet points
ASSIGNMENT - Main task: Technical dimension Which technical software requirements should the grocery shopping app meet?	Free text – bullet points

This dimension includes requirements that refer to the maintenance, evolution, resilience, and the changeableness of artificial systems, such as soft- and hardware. Example: '- the app should be developed with Java'	
ASSIGNMENT - Main task: Purpose dimension Which software requirements need to be met for the grocery shopping app to fulfil its purpose? This dimension includes requirements that enable an application to fulfil its purpose. Example: '- the app should remind me about used up supplies'	Free text – bullet points
Questionnaire - Part II	
Age	Free text - Years
Gender	• Female • Male • Diverse
I knew about software quality criteria according to ISO 25010: (functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, portability) before the study.	• Agree • Somewhat agree • Neither agree nor disagree • Somewhat disagree • Disagree
How important is sustainability to you in your daily life?	• Important • Somewhat important • Neither important nor unimportant • Somewhat unimportant • Unimportant
Do you use an app to buy groceries?	• No • Yes • No information
If you have any further comments on the survey, you can leave them here.	Free text
Conclusion	
Clarification about the experiment, summary of our research endeavor, instruction to participate in the raffle, thanking for participation	

Appendix C – Focus Group Discussion Guideline

Adopted from Faltin [59]

Introduction

- Name, job-title, responsibilities of each participant
- Facilitator summarizes the estimation procedure, its background and motivation

Simulation of Group Estimation

- Facilitator introduces the respective requirement example
- Facilitator presents estimation results and arguments that participants provided during the self-estimation
- Guiding questions:
  - Can each of you explain your estimation?
  - Would you change an estimation after hearing the other arguments?
  - Can you agree on an estimate?

From Iteration III:

  - Can you explain why you rated this dimension with “I don’t know”

Evaluation of the Estimation Procedure

- What do you think about the procedure in general?
- What do you think about the rating scale?
- What do you think about using this procedure to design sustainable software?
- What do you think about sustainability and software design after performing this procedure?
- Can you comment on the procedure's suitability for estimating requirements regarding their impact on a software product's sustainability?
- Could you integrate this procedure into your working life?

Evaluation of the *Sustainability Cards*

- What do you think about the cards?
- What do you think about the comprehensibility?
- How did the cards contribute to your estimation?
- How did you perceive the size?
- What do you think about the design?

List of Figures

List of Tables

List of Publications

The thesis is based on the following publications. We used parts of them verbatim in the corresponding chapters:

- Pham, Y. D., Bouraffa, A., Hillen, M., and Maalej, W., "The Role of Linguistic Relativity on the Identification of Sustainability Requirements: An Empirical Study," in *2021 IEEE 29th International Requirements engineering Conference (RE)*, Sep. 2021

- Pham, Y. D., "shape*re*-Framework: Forschungsvorhaben zur Ermittlung und Repräsentation von Anforderungen nachhaltiger Software-Innovationen, [shape*re*-Framework: Research Project on the Identification and Representation of Requirements of Sustainable Software Innovations]" in *Jahrbuch Digital Design 2021*, Berlin: Bitkom Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., 2021

- Pham, Y. D., Bouraffa, A., and Maalej, W., "ShapeRE: Towards a Multi-Dimensional Representation for Requirements of Sustainable Software," in *2020 IEEE 28th International Requirements engineering Conference (RE)*, Aug. 2020

- Pham, Y. D., Montgomery, L., and Maalej, W., "Renovating Requirements Engineering: First Thoughts to Shape Requirements Engineering as a Profession," in *2019 IEEE 27th International Requirements engineering Conference Workshops (REW)*, Sep. 2019

- Pham, Y. D., Fucci, D., and Maalej, W., "A First Implementation of a Design Thinking Workshop during a Mobile App Development Course Project," in *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*, Jun. 2018

Additional, co-authored and peer-reviewed publications not included in this thesis:

- Maalej, W., Pham, Y. D., and Chazette, L., "Tailoring Requirements Engineering for Responsible AI," *Computer*, 2023 (to appear)

Bibliography

[1] Abidin, S. Z., Sigurjónsson, J., Liem, A., and Keitsch, M., "On the role of formgiving in design," in *DS 46: Proceedings of E&PDE 2008, The 10th International Conference on Engineering and Product Design Education*, Aug. 2008, pp. 365–370.

[2] Albayrak, O., Albayrak, D., and Kilic, T., "Are software engineers' responses to incomplete requirements related to project characteristics?" in *2009 Second International Conference on the Applications of Digital Information and Web Technologies*, Aug. 2009, pp. 124–129.

[3] Alexander, C., "The origins of pattern theory: the future of the theory, and the generation of a living world," *IEEE Softw.*, vol. 16, no. 5, pp. 71–82, Oct. 1999.

[4] Alhamed, M. and Storer, T., "Playing Planning Poker in Crowds: Human Computation of Software Effort Estimates," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, May 2021, pp. 1–12.

[5] Alharthi, A. D., Spichkova, M., and Hamilton, M., "SuSoftPro: Sustainability Profiling for Software," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Aug. 2018, pp. 500–501.

[6] Alrajeh, D., Kramer, J., van Lamsweerde, A., Russo, A., and Uchitel, S., "Generating obstacle conditions for requirements completeness," in *2012 34th International Conference on Software Engineering (ICSE)*, Jun. 2012, pp. 705–715.

[7] Amici, F., Sánchez-Amaro, A., Sebastián-Enesco, C., Cacchione, T., Allritz, M., Salazar-Bonet, J., and Rossano, F., "The word order of languages predicts native speakers' working memory," *Sci. Rep.*, vol. 9, no. 1, p. 1124, Dec. 2019.

[8] Arora, C., Sabetzadeh, M., Briand, L., and Zimmer, F., "Automated Checking of Conformance to Requirements Templates Using Natural Language Processing," *IEEE Trans. Softw. Eng.*, vol. 41, no. 10, pp. 944–968, Oct. 2015.

[9] Asdrubali, F., D'Alessandro, F., and Schiavoni, S., "A review of unconventional sustainable building insulation materials," *Sustain. Mater. Technol.*, vol. 4, pp. 1–17, Jul. 2015.

[10] Bambazek, P., Groher, I., and Seyff, N., "Sustainability in Agile Software Development: A Survey Study among Practitioners," in *2022 International Conference on ICT for Sustainability (ICT4S)*, Jun. 2022, pp. 13–23.

[11] Barendsen, W., Muß, A. C., and Silvius, G., "Exploring team members' perceptions of internal sustainability communication in sustainable project management," *Proj. Leadersh. Soc.*, vol. 2, p. 100015, Dec. 2021.

[12] Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S. M., Penzenstadler, B., Seyff, N., and Venters, C. C., "Requirements: The Key to Sustainability," *IEEE Softw.*, vol. 33, no. 1, pp. 56–65, Jan. 2016.

[13] Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., and Venters, C. C., "Sustainability Design and Software: The Karlskrona Manifesto," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE)*, May 2015, pp. 467–476.

[14] Benington, H. D., "Production of Large Computer Programs," *IEEE Ann. Hist. Comput.*, vol. 5, no. 4, pp. 350–361, Oct. 1983.

[15] Berry, D. M., "Software and House Requirements Engineering: Lessons Learned in Combating Requirements Creep," *Requir. Eng.*, vol. 3, no. 3–4, pp. 242–244, Mar. 1998.

[16] Berry, D. M., “More requirements engineering adventures with building contractors,” *Requir. Eng.*, vol. 8, no. 2, pp. 142–146, Jul. 2003.

[17] Berry, D. M., “Appliances and Software: The Importance of the Buyer’s Warranty and the Developer’s Liability in Promoting the Use of Systematic Quality Assurance and Formal Methods,” in *2000 Monterey Workshop on Modelling Software System Structures in a fastly moving scenario*, 2000, p. 38.

[18] Berry, D. M. and Kamsties, E., “Ambiguity in Requirements Specification,” in *Perspectives on Software Requirements*, Prado Leite, J. C. S. and Doorn, J. H., Eds. Boston, MA: Springer US, 2004, pp. 7–44.

[19] Blomkvist, S., “Towards a Model for Bridging Agile Development and User-Centered Design,” in *Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle*, Seffah, A., Gulliksen, J., and Desmarais, M. C., Eds. Dordrecht: Springer Netherlands, 2005, pp. 219–244.

[20] Blutstein, H. and Rodger, A., “The sustainable tall building of the third millennium,” in *Tall Buildings and Urban Habitat*, CRC Press, 2001, pp. 153–164.

[21] Boroditsky, L., “How Language Shapes Thought,” *Sci. Am.*, vol. 304, no. 2, pp. 62–65, Feb. 2011.

[22] Boroditsky, L., Schmidt, L. A., and Phillips, W., “Sex, syntax, and semantics,” *Lang. Mind Adv. Study Lang. Thought*, pp. 61–79, 2003.

[23] Bourque, P., Fairley, R. E., and IEEE Computer Society, *Guide to the software engineering body of knowledge*. 2014.

[24] Brhel, M., Meth, H., Maedche, A., and Werder, K., “Exploring principles of user-centered agile software development: A literature review,” *Inf. Softw. Technol.*, vol. 61, pp. 163–181, May 2015.

[25] Briassoulis, H., “Sustainable Development and its Indicators: Through a (Planner’s) Glass Darkly,” *J. Environ. Plan. Manag.*, vol. 44, no. 3, pp. 409–427, May 2001.

[26] Brito, I. S., Conejero, J. M., Moreira, A., and Araujo, J., “A concern-oriented sustainability approach,” in *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, May 2018, pp. 1–12.

[27] Broy, M., “Rethinking Functional Requirements: A Novel Approach Categorizing System and Software Requirements,” in *Software Technology: 10 Years of Innovation in IEEE Computer*, Hinchey, M., Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018, pp. 155–187.

[28] Budgen, D., *Software design*, 2nd ed. Harlow, England ; New York: Addison-Wesley, 2003.

[29] Buhl, A., Schmidt-Keilich, M., Muster, V., Blazejewski, S., Schrader, U., Harrach, C., Schäfer, M., and Süßbauer, E., “Design thinking for sustainability: Why and how design thinking can foster sustainability-oriented innovation development,” *J. Clean. Prod.*, vol. 231, pp. 1248–1257, Sep. 2019.

[30] Burnay, C., Horkoff, J., and Maiden, N., “Stimulating Stakeholders’ Imagination: New Creativity Triggers for Eliciting Novel Requirements,” in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, Sep. 2016, pp. 36–45.

[31] Cabot, J., Easterbrook, S., Horkoff, J., Lessard, L., Liaskos, S., and Mazon, J.-N., “Integrating sustainability in decision-making processes: A modelling strategy,” in *2009 31st International Conference on Software Engineering - Companion Volume*, 2009, pp. 207–210.

[32] Calero, C., García-Rodríguez De Guzmán, I., Moraga, M. A., and García, F., “Is software sustainability considered in the CSR of software industry?,” *Int. J. Sustain. Dev. World Ecol.*, vol. 26, no. 5, pp. 439–459, Jul. 2019.

[33] Calero, C. and Piattini, M., *Green in software engineering*, vol. 3. New York, NY: Springer Berlin Heidelberg, 2015.

[34] Cambridge University Press, “Definition of inspiration,” *Cambridge Dictionary*. 2022. Accessed: Jan. 05, 2022. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/inspiration

[35] Charter, M. and Tischner, U., Eds., *Sustainable Solutions: Developing Products and Services for the Future*, 1st ed. Routledge, 2001.

[36] Chitchyan, R., Becker, C., Betz, S., Duboc, L., Penzenstadler, B., Seyff, N., and Venters, C. C., “Sustainability design in requirements engineering: state of practice,” in *Proceedings of the 38th International Conference on Software Engineering Companion*, May 2016, pp. 533–542.

[37] Clark, R. M., Stabryla, L. M., and Gilbertson, L. M., “Sustainability coursework: student perspectives and reflections on design thinking,” *Int. J. Sustain. High. Educ.*, vol. 21, no. 3, pp. 593–611, Feb. 2020.

[38] Clarke, V. and Braun, V., “Thematic Analysis,” in *Encyclopedia of Quality of Life and Well-Being Research*, Michalos, A. C., Ed. Dordrecht: Springer Netherlands, 2014, pp. 6626–6628.

[39] Cohn, M., *Agile estimating and planning*. Pearson Education, 2005.

[40] Condori-Fernandez, N. and Lago, P., “Towards a Software Sustainability-Quality Model: Insights from a Multi-Case Study,” in *2019 13th International Conference on Research Challenges in Information Science (RCIS)*, May 2019, pp. 1–11.

[41] Constantine, L. L. and Lockwood, L., “Process agility and software usability: Toward lightweight usage-centered design,” *Inf. Age*, vol. 8, no. 8, pp. 1–10, 2002.

[42] Costanzo Sow, S., “Sustainable Development - What is there to know and why should we care?,” Oct. 17, 2016. https://www.unssc.org/news-and-insights/blog/sustainable-development-what-there-know-and-why-should-we-care (accessed Jul. 17, 2022).

[43] Craig, C. A. and Allen, M. W., “Sustainability information sources: employee knowledge, perceptions, and learning,” *J. Commun. Manag.*, vol. 17, no. 4, pp. 292–307, Nov. 2013.

[44] Cramer, A., *Zur Lage der Familie und der Familienpolitik in der Bundesrepublik Deutschland [On the Situation of the Family and Family Policy in the Federal Republic of Germany]*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1982.

[45] Creswell, J. W. and Creswell, J. D., *Research design: Qualitative, quantitative, and mixed methods approaches*, 5th ed. Los Angeles: Sage Publications, 2018.

[46] Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S., “Requirements engineering: A systematic mapping study in agile software development,” *J. Syst. Softw.*, vol. 139, pp. 32–50, May 2018.

[47] Davis, A. M., *201 principles of software development*. New York, McGraw-Hill Inc., 1995.

[48] DeVries, B. and Cheng, B. H. C., “Automatic detection of incomplete requirements via symbolic analysis,” in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, Oct. 2016, pp. 385–395.

[49] Duboc, L., Betz, S., Penzenstadler, B., Akinli Kocak, S., Chitchyan, R., Leifler, O., Porras, J., Seyff, N., and Venters, C. C., “Do we Really Know What we are Building? Raising Awareness of Potential Sustainability Effects of Software Systems in Requirements Engineering,” in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Sep. 2019, pp. 6–16.

[50] Duboc, L., Penzenstadler, B., Porras, J., Akinli Kocak, S., Betz, S., Chitchyan, R., Leifler, O., Seyff, N., and Venters, C. C., “Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow,” *Requir. Eng.*, vol. 25, no. 4, pp. 469–492, Dec. 2020.

[51] Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., and Leifer, L. J., “Engineering Design Thinking, Teaching, and Learning,” *J. Eng. Educ.*, vol. 94, no. 1, pp. 103–120, Jan. 2005.

[52] Eckhardt, J., Vogelsang, A., Femmer, H., and Mager, P., “Challenging Incompleteness of Performance Requirements by Sentence Patterns,” in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, Sep. 2016, pp. 46–55.

[53] Eckhardt, J., Vogelsang, A., and Méndez Fernández, D., “On the Distinction of Functional and Quality Requirements in Practice,” in *Product-Focused Software Process Improvement*, vol. 10027, Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., and Mikkonen, T., Eds. Cham: Springer International Publishing, 2016, pp. 31–47.

[54] Economist Intelligence Unit (EIU), “An Eco-Wakening: Measuring global engagement, awareness and action for nature,” World Wide Fund for Nature (WWF), 2021.

[55] Ekardt, F., *Sustainability: Transformation, Governance, Ethics, Law*. Cham: Springer International Publishing, 2020.

[56] Eloranta, V.-P., Koskimies, K., Mikkonen, T., and Vuorinen, J., “Scrum Anti-Patterns - An Empirical Study,” in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, Dec. 2013, pp. 503–510.

[57] Enfield, N. J., “Linguistic Relativity from Reference to Agency,” *Annu. Rev. Anthropol.*, vol. 44, no. 1, pp. 207–224, Oct. 2015.

[58] European Comission, “Delivering the European Green Deal,” *European Comission*. https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal/delivering-european-green-deal_en (accessed Jul. 05, 2022).

[59] Faltin, T., “Entwicklung eines Verfahrens zur Einschätzung der Softwarenachhaltigkeit in agilen Projekten [Design of an Estimation Procedure for Software Sustainability in Agile Projects],” M.S. Thesis, University of Hamburg, Hamburg, 2022.

[60] Federal Association of German Architects, “Architects in Germany.” Federal Association of German Architects, May 2011. Accessed: Jun. 02, 2020. [Online]. Available: https://www.bda-bund.de/wp-content/uploads/2016/05/CoFA_Flyer.pdf

[61] Federal Ministry of the Interior, Building and Community (BMI), “Guideline for Sustainable Building,” Federal Ministry of the Interior, Building and Community (BMI), Apr. 2019.

[62] Ferrari, A., dell’Orletta, F., Spagnolo, G. O., and Gnesi, S., “Measuring and Improving the Completeness of Natural Language Requirements,” in *Requirements Engineering: Foundation for Software Quality*, vol. 8396, Salinesi, C. and van de Weerd, I., Eds. Cham: Springer International Publishing, 2014, pp. 23–38.

[63] Ferreira, J., Noble, J., and Biddle, R., “Agile Development Iterations and UI Design,” in *AGILE 2007 (AGILE 2007)*, Aug. 2007, pp. 50–58.

[64] Forsberg, K. and Mooz, H., “The Relationship of Systems Engineering to the Project Cycle,” *Eng. Manag. J.*, vol. 4, no. 3, pp. 36–43, Sep. 1992.

[65] Fowler, M. and Highsmith, J., “The agile manifesto,” *Softw. Dev.*, vol. 9, no. 8, pp. 28–35, 2001.

[66] Fox, D., Sillito, J., and Maurer, F., “Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry,” in *Agile 2008 Conference*, 2008, pp. 63–72.

[67] Friedman, B. and Hendry, D., “The envisioning cards: a toolkit for catalyzing humanistic and technical imaginations,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, May 2012, pp. 1145–1148.

[68] Gandomani, T. J., Faraji, H., and Radnejad, M., “Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus,” in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, Feb. 2019, pp. 066–071.

[69] Gandomani, T. J., Wei, K. T., and Binhamid, A. K., “A Case Study Research on Software Cost Estimation Using Experts’ Estimates, Wideband Delphi, and Planning Poker Technique,” *Int. J. Softw. Eng. Its Appl.*, vol. 8, no. 11, p. 11, 2014.

[70] Garcia, A., Silva da Silva, T., and Selbach Silveira, M., “Artifacts for Agile User-Centered Design: A Systematic Mapping,” in *Hawaii International Conference on System Sciences*, 2017, pp. 5859–5868.

[71] García-Mireles, G. A. and Villa-Martínez, H. A., “Practices for Addressing Environmental Sustainability through Requirements Processes,” in *Trends and Applications in Software Engineering*, vol. 688, Mejia, J., Muñoz, M., Rocha, Á., Quiñonez, Y., and Calvo-Manzano, J., Eds. Cham: Springer International Publishing, 2018, pp. 61–70.

[72] German Environment Agency, “Resources and Energy-Efficient Software Products (DE-UZ 215),” *Blue Angel - The German Ecolabel*. https://www.blauer-engel.de/en/productworld/resources-and-energy-efficient-software-products (accessed Jun. 06, 2022).

[73] Ghazaei, G., Laina, I., Rupprecht, C., Tombari, F., Navab, N., and Nazarpour, K., “Dealing with Ambiguity in Robotic Grasping via Multiple Predictions,” in *Computer Vision – ACCV 2018*, vol. 11364, Jawahar, C. V., Li, H., Mori, G., and Schindler, K., Eds. Cham: Springer International Publishing, 2019, pp. 38–55.

[74] Glinz, M., “On Non-Functional Requirements,” in *15th IEEE International Requirements Engineering Conference (RE 2007)*, Oct. 2007, pp. 21–26.

[75] Glomann, L., “Introducing ‘Human-Centered Agile Workflow’ (HCAW) – An Agile Conception and Development Process Model,” in *Advances in Usability and User Experience*, vol. 607, Ahram, T. and Falcão, C., Eds. Cham: Springer International Publishing, 2018, pp. 646–655.

[76] Godemann, J. and Michelsen, G., “Sustainability communication–an introduction,” in *Sustainability communication*, Springer, 2011, pp. 3–11.

[77] Goldschmidt, G. and Weil, M., “Contents and Structure in Design Reasoning,” *Des. Issues*, vol. 14, no. 3, p. 85, 1998.

[78] Granger, C. W. J., *Forecasting in business and economics*. Academic Press, 2014.

[79] Grenning, J., “Planning poker or how to avoid analysis paralysis while release planning,” *Hawthorn Woods Renaiss. Softw. Consult.*, vol. 3, pp. 22–23, 2002.

[80] Groher, I. and Weinreich, R., “An Interview Study on Sustainability Concerns in Software Development Projects,” in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Aug. 2017, pp. 350–358.

[81] Guldin, A., “Führung und Innovation [Leadership and Innovation],” in *Die Zukunft der Führung*, Springer, 2012, pp. 213–233.

[82] Guldin, A. and Gelléri, P., “Förderung von Innovationen [Promoting Innovation],” in *Lehrbuch der Personalpsychologie*, Hogrefe, 2014, pp. 607–646.

[83] Gupta, A. K. and Singhal, A., “Managing Human Resources for Innovation and Creativity,” *Res.-Technol. Manag.*, vol. 36, no. 3, pp. 41–48, May 1993.

[84] Hamberger, J., “Von der Sylvicultura zur Waldkultur,” *LWF Wissen*, vol. 72, pp. 15–23, 2013.

[85] Hargroves, K. and Smith, M. H., Eds., *The natural advantage of nations: business opportunities, innovation, and governance in the 21st century*. London ; Sterling, VA: Earthscan, 2005.

[86] Hasso Plattner Institute of Design at Stanford University, “Design Thinking Bootcamp Bootleg,” *d.school*. https://dschool.stanford.edu/resources/the-bootcamp-bootleg (accessed Nov. 30, 2017).

[87] Hasso Plattner Institute of Design at Stanford University, “How to Kick Off a Crash Course,” *d.school*. https://dschool.stanford.edu/resources/gear-up-how-to-kick-off-a-crash-course (accessed Nov. 30, 2017).

[88] Hauschildt, J., Salomo, S., Schultz, C., and Kock, A., *Innovationsmanagement*. Vahlen, 2016.

[89] Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., and Broy, M., “On Integrating Design Thinking for Human-Centered Requirements Engineering,” *IEEE Softw.*, vol. 37, no. 2, pp. 25–31, Mar. 2020.

[90] Hehn, J. and Uebernickel, F., “The Use of Design Thinking for Requirements Engineering: An Ongoing Case Study in the Field of Innovative Software-Intensive Systems,” in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Aug. 2018, pp. 400–405.

[91] Henry, A. D., “Learning sustainability innovations,” *Nat. Sustain.*, vol. 1, no. 4, pp. 164–165, Apr. 2018.

[92] Hess, A., Trapp, M., Karras, O., Seyff, N., and Lauenroth, K., “Welcome to the 3rd International Workshop on Learning from Other Disciplines for Requirements Engineering (D4RE’19),” in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, Sep. 2019, pp. 1–2.

[93] Hill, L. A., Brandeau, G., Truelove, E., and Lineback, K. L., *Collective genius: The art and practice of leading innovation*. Harvard Business Review Press, 2014.

[94] Hillen, M., “Sustainability in Requirements Engineering: Investigating the Use of Requirements Dimensions,” B.S. Thesis, University of Hamburg, Hamburg, 2021.

[95] Hoda, R., Salleh, N., and Grundy, J., “The Rise and Evolution of Agile Software Development,” *IEEE Softw.*, vol. 35, no. 5, pp. 58–63, Sep. 2018.

[96] Horkoff, J., Maiden, N. A., and Asboth, D., “Creative goal modeling for innovative requirements,” *Inf. Softw. Technol.*, vol. 106, pp. 85–100, Feb. 2019.

[97] Huemann, M. and Silvius, G., “Projects to create the future: Managing projects meets sustainable development,” *Int. J. Proj. Manag.*, vol. 35, no. 6, pp. 1066–1070, Aug. 2017.

[98] International Organization for Standardization, “ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.” International Organization for Standardization, 2011.

[99] Jacobsen, I., Booch, G., and Rumbaugh, J., *The unified software development process*. Reading (Mass.) : Addison-Wesley, 1999.

[100] Jaspers, M., Steen, T., Bos, C., and Geenen, M., “The think aloud method: a guide to user interface design,” *Int. J. Med. Inf.*, vol. 73, no. 11–12, pp. 781–795, Nov. 2004.

[101] Johann, T. and Maalej, W., “Position Paper: The Social Dimension of Sustainability in Requirements Engineering,” presented at the 2nd

International Workshop on Requirements Engineering for Sustainable Systems, Jul. 2013.

[102] Johansson-Sköldberg, U., Woodilla, J., and Çetinkaya, M., “Design Thinking: Past, Present and Possible Futures,” *Creat. Innov. Manag.*, vol. 22, no. 2, pp. 121–146, Jun. 2013.

[103] Kapor, M., “A Software Design Manifesto,” in *Bringing Design to Software*, Winograd, T., Ed. ACM, 1996.

[104] Karita, L., Mourão, B. C., and Machado, I., “Software industry awareness on green and sustainable software engineering: a state-of-the-practice survey,” in *Proceedings of the XXXIII Brazilian Symposium on Software Engineering - SBES 2019*, 2019, pp. 501–510.

[105] Keitsch, M., “Sustainable Architecture, Design and Housing: Some general thoughts on Sustainable Architecture, Design and Housing,” *Sustain. Dev.*, vol. 20, no. 3, pp. 141–145, May 2012.

[106] Kempf, A. and Osthoff, P., “The Effect of Socially Responsible Investing on Portfolio Performance,” *Eur. Financ. Manag.*, vol. 13, no. 5, pp. 908–922, Nov. 2007.

[107] Kitchenham, B. and Pfleeger, S. L., “Principles of survey research: part 5: populations and samples,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 27, no. 5, pp. 17–20, Sep. 2002.

[108] Koerner, E. F. K., “The Sapir-Whorf Hypothesis: A Preliminary History and a Bibliographical Essay,” *J. Linguist. Anthropol.*, vol. 2, no. 2, pp. 173–198, Dec. 1992.

[109] Konstanti, C., Karapanos, E., and Markopoulos, P., “The Behavior Change Design Cards: A Design Support Tool for Theoretically-Grounded Design of Behavior Change Technologies,” *Int. J. Human–Computer Interact.*, pp. 1–17, Nov. 2021.

[110] Koziolek, H., “Sustainability evaluation of software architectures: a systematic review,” in *Proceedings of the joint ACM SIGSOFT conference--QoSA and ACM SIGSOFT symposium--ISARCS on Quality of software architectures--QoSA and architecting critical systems--ISARCS*, 2011, pp. 3–12.

[111] Kremer, H., Villamor, I., and Aguinis, H., “Innovation leadership: Best-practice recommendations for promoting employee creativity, voice, and knowledge sharing,” *Bus. Horiz.*, vol. 62, no. 1, pp. 65–74, Jan. 2019.

[112] Lago, P., Koçak, S. A., Crnkovic, I., and Penzenstadler, B., “Framing sustainability as a property of software quality,” *Commun. ACM*, vol. 58, no. 10, pp. 70–78, Sep. 2015.

[113] Lago, P., Verdecchia, R., Condori-Fernandez, N., Rahmadian, E., Sturm, J., van Nijnanten, T., Bosma, R., Debuysscher, C., and Ricardo, P., “Designing for Sustainability: Lessons Learned from Four Industrial Projects,” in *Advances and New Trends in Environmental Informatics*, Kamilaris, A., Wohlgemuth, V., Karatzas, K., and Athanasiadis, I. N., Eds. Cham: Springer International Publishing, 2021, pp. 3–18.

[114] Lauenroth, K., “From Design Thinking in Software Engineering to Digital Design as a New Profession: An Essay on Methods and Professions for Shaping Digital Solutions and Systems,” in *Design Thinking for Software Engineering*, Springer, 2022, pp. 61–72.

[115] Lauenroth, K., Kamsties, E., and Hehlert, O., “Do Words Make a Difference? An Empirical Study on the Impact of Taxonomies on the Classification of Requirements,” in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Sep. 2017, pp. 273–282.

[116] Lauenroth, K., Lehn, K., Trapp, M., and Schubert, U., “Digital Design - Der nächste Schritt für das Requirements Engineering im Kontext der Digitalen

Transformation [Digital Design - The Next Step for Requirements Engineering in the Context of Digital Transformation],” *Softwaretechnik-Trends*, vol. 38, pp. 47–48, 2018.

[117] Lea, D., “Christopher Alexander: an introduction for object-oriented designers,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 19, no. 1, pp. 39–46, Jan. 1994.

[118] Leavy, P., *Research design: Quantitative, qualitative, mixed methods, arts-based, and community-based participatory research approaches*. Guilford Publications, 2017.

[119] Lietz, P., “Research into Questionnaire Design: A Summary of the Literature,” *Int. J. Mark. Res.*, vol. 52, no. 2, pp. 249–272, Mar. 2010.

[120] Lindberg, T., Noweski, C., and Meinel, C., “Evolving discourses on design thinking: how design cognition inspires meta-disciplinary creative collaboration,” *Technoetic Arts*, vol. 8, no. 1, pp. 31–37, May 2010.

[121] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., and Brinkkemper, S., “Improving agile requirements: the Quality User Story framework and tool,” *Requir. Eng.*, vol. 21, no. 3, pp. 383–403, Sep. 2016.

[122] Lucero, A., Dalsgaard, P., Halskov, K., and Buur, J., “Designing with Cards,” in *Collaboration in Creative Design*, Markopoulos, P., Martens, J.-B., Malins, J., Coninx, K., and Liapis, A., Eds. Cham: Springer International Publishing, 2016, pp. 75–95.

[123] Maalej, W., Kurtanović, Z., Nabil, H., and Stanik, C., “On the automatic classification of app reviews,” *Requir. Eng.*, vol. 21, no. 3, pp. 311–331, Sep. 2016.

[124] Maalej, W. and Thurimella, A. K., Eds., *Managing Requirements Knowledge*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[125] Mahaux, M., Heymans, P., and Saval, G., “Discovering Sustainability Requirements: An Experience Report,” in *Requirements Engineering: Foundation for Software Quality*, vol. 6606, Berry, D. and Franch, X., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 19–33.

[126] Mahnič, V. and Hovelja, T., “On using planning poker for estimating user stories,” *J. Syst. Softw.*, vol. 85, no. 9, pp. 2086–2095, Sep. 2012.

[127] Maier, G. W., Hülsheger, U. R., Anderson, N., and Steinmann, B., “Innovation und Kreativität in Projekten [Innovation and creativity in projects],” in *Angewandte Psychologie für das Projektmanagement: Ein Praxisbuch für die erfolgreiche Projektleitung*, Wastian, M., Braumandl, I., von Rosenstiel, L., and West, M. A., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 249–266.

[128] Mairiza, D., Zowghi, D., and Nurmuliani, N., “An investigation into the notion of non-functional requirements,” in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC ’10*, 2010, p. 311.

[129] Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., Pollock, L., and Clause, J., “An empirical study of practitioners’ perspectives on green software engineering,” in *Proceedings of the 38th International Conference on Software Engineering*, May 2016, pp. 237–248.

[130] Marckmann, B., Gram-Hanssen, K., and Christensen, T. H., “Sustainable Living and Co-Housing: Evidence from a Case Study of Eco-Villages,” *Built Environ.*, vol. 38, no. 3, pp. 413–429, Jul. 2012.

[131] Marquis, C., “Searching For A Solution: Ecosia Is Taking On Google And Aiding Reforestation Efforts In The Process,” *Forbes*. https://www.forbes.com/sites/christophermarquis/2021/09/01/searching-for-a-solution-ecosia-is-taking-on-google-and-aiding-reforestation-efforts-in-the-process/ (accessed Jul. 05, 2022).

[132] Marshall, M. N., "Sampling for qualitative research," *Fam. Pract.*, vol. 13, no. 6, pp. 522–526, 1996.

[133] Martens, D. and Maalej, W., "Extracting and Analyzing Context Information in User-Support Conversations on Twitter," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Sep. 2019, pp. 131–141.

[134] Massey, A. K., Rutledge, R. L., Anton, A. I., and Swire, P. P., "Identifying and classifying ambiguity for regulatory requirements," in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Aug. 2014, pp. 83–92.

[135] Mauger, C. and Berry, D. M., "Lessons Learned from and for Requirements Engineering and Building Construction: A Case Study of Requirements Engineering for a Synagogue Kitchen with Use Cases and Scenarios," in *2014 IEEE International Conference on Software Science, Technology and Engineering*, Jun. 2014, pp. 67–76.

[136] McCabe, D., Langer, K. G., Cornwell, M. A., Borod, J. C., and Bender, H. A., "Practice Effects," *Encyclopedia of Clinical Neuropsychology*. Springer International Publishing, Cham, 2018.

[137] McEntire, L. E. and Greene-Shortridge, T. M., "Recruiting and Selecting Leaders for Innovation: How to Find the Right Leader," *Adv. Dev. Hum. Resour.*, vol. 13, no. 3, pp. 266–278, Aug. 2011.

[138] McHugh, M. L., "Interrater reliability: the kappa statistic," *Biochem. Medica*, pp. 276–282, 2012.

[139] Meinel, C. and Leifer, L., "Introduction – Design Thinking Is Mainly About Building Innovators," in *Design Thinking Research*, Plattner, H., Meinel, C., and Leifer, L., Eds. Cham: Springer International Publishing, 2015, pp. 1–11.

[140] Méndez Fernández, D. *et al.*, "Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice," *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, Oct. 2017.

[141] Menzel, I., Mueller, M., Gross, A., and Doerr, J., "An Experimental Comparison Regarding the Completeness of Functional Requirements Specifications," in *2010 18th IEEE International Requirements Engineering Conference*, Sep. 2010, pp. 15–24.

[142] Mishra, P., Zhao, Y., and Tan, S., "From Concept to Software: Developing a Framework for Understanding the Process of Software Design," *J. Res. Comput. Educ.*, vol. 32, no. 2, pp. 220–238, Dec. 1999.

[143] Mohanani, R., Ralph, P., Turhan, B., and Mandic, V., "How Templated Requirements Specifications Inhibit Creativity in Software Engineering," *IEEE Trans. Softw. Eng.*, pp. 1–1, 2021.

[144] Moløkken-Østvold, K., Haugen, N. C., and Benestad, H. C., "Using planning poker for combining expert estimates in software projects," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2106–2117, Dec. 2008.

[145] Morgan, D. L. and Hoffman, K., "Focus groups," in *The sage handbook of qualitative data collection*, 2018, pp. 250–263.

[146] Nadin, M., "Interface design and evaluation–semiotic implications," in *Advances in human-computer interaction*, vol. 2, Hartson, H. R. and Hix, D., Eds. Ablex Publishing Norwood, 1988, pp. 45–100.

[147] Nathan, L. P., Friedman, B., Klasnja, P., Kane, S. K., and Miller, J. K., "Envisioning systemic effects on persons and society throughout interactive system design," in *Proceedings of the 7th ACM conference on Designing interactive systems - DIS '08*, 2008, pp. 1–10.

[148] Naumann, S., Dick, M., Kern, E., and Johann, T., “The GREENSOFT Model: A reference model for green and sustainable software and its engineering,” *Sustain. Comput. Inform. Syst.*, vol. 1, no. 4, pp. 294–304, Dec. 2011.

[149] Naumann, S., Kern, E., Guldner, A., and Birkenfeld, U.-C., “Umweltzeichen Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte,” Umweltbundesamt, 2021.

[150] Neufert, E. and Neufert, P., *Architects’ data.*, 5th ed. John Wiley & Sons, 2019.

[151] Newman, P., Ferrario, M. A., Simm, W., Forshaw, S., Friday, A., and Whittle, J., “The Role of Design Thinking and Physical Prototyping in Social Software Engineering,” in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, May 2015, pp. 487–496.

[152] Olsen, C. and Mac Namara, S. C., *Collaborations in architecture and engineering*. New York, NY: Routledge, 2014.

[153] Orne, M. T., “On The Social Psychology of the Psychological Experiment: With Particular Reference to Demand Characteristics and Their Implications,” in *Sociological Methods*, Routledge, 2017, pp. 279–299.

[154] Owen, C. and Dovey, K., “Fields of sustainable architecture,” *J. Archit.*, vol. 13, no. 1, pp. 9–21, Feb. 2008.

[155] Oxford University Press, “Definition of holism,” *Lexico Dictionaries*. 2022. Accessed: Jun. 30, 2022. [Online]. Available: https://www.lexico.com/en/definition/holism

[156] Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H., and Franch, X., “Systematic mapping study on software engineering for sustainability (SE4S),” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE ’14*, 2014, pp. 1–14.

[157] Penzenstadler, B., Raturi, A., Richardson, D., and Tomlinson, B., “Safety, Security, Now Sustainability: The Nonfunctional Requirement for the 21st Century,” *IEEE Softw.*, vol. 31, no. 3, pp. 40–47, May 2014.

[158] Pereira, L., Parizi, R., Prestes, M., Marczak, S., and Conte, T., “Towards an understanding of benefits and challenges in the use of design thinking in requirements engineering,” in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, Mar. 2021, pp. 1338–1345.

[159] Pérez-Castillo, R., de Guzmán, I. G.-R., and Piattini, M., “Knowledge Discovery Metamodel-ISO/IEC 19506: A standard to modernize legacy systems,” *Comput. Stand. Interfaces*, vol. 33, no. 6, pp. 519–532, Nov. 2011.

[160] Petersen, K. and Wohlin, C., “The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study,” *Empir. Softw. Eng.*, vol. 15, no. 6, pp. 654–693, Dec. 2010.

[161] Pham, Y. D., “shapere-Framework: Forschungsvorhaben zur Ermittlung und Repräsentation von Anforderungen nachhaltiger Software-Innovationen [shapere-Framework: Research Project on the Identification and Representation of Requirements of Sustainable Software Innovations],” in *Jahrbuch Digital Design 2021*, Berlin: Bitkom Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., 2021, p. 15.

[162] Pham, Y. D., Bouraffa, A., Hillen, M., and Maalej, W., “The Role of Linguistic Relativity on the Identification of Sustainability Requirements: An Empirical Study,” in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, Sep. 2021, pp. 117–127.

[163] Pham, Y. D., Bouraffa, A., and Maalej, W., “ShapeRE: Towards a Multi-Dimensional Representation for Requirements of Sustainable Software,” in

2020 IEEE 28th International Requirements Engineering Conference (RE), Aug. 2020, pp. 358–363.

[164] Pham, Y. D., Fucci, D., and Maalej, W., “A first implementation of a design thinking workshop during a mobile app development course project,” in *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*, Jun. 2018, pp. 56–63.

[165] Pham, Y. D., Montgomery, L., and Maalej, W., “Renovating Requirements Engineering: First Thoughts to Shape Requirements Engineering as a Profession,” in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, Sep. 2019, pp. 7–11.

[166] Pohl, K. and Rupp, C., *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam, foundation level, IREB compliant*, Second edition. Santa Barbara, CA: Rocky Nook, 2015.

[167] Pufé, I., “Was ist Nachhaltigkeit? Dimensionen und Chancen [What is Sustainability? Dimensions and Opportunities],” *Aus Polit. Zeitgesch.*, vol. 31–32, no. Nachhaltigkeit, 2014.

[168] Raturi, A., Penzenstadler, B., Tomlinson, B., and Richardson, D., “Developing a sustainability non-functional requirements framework,” in *Proceedings of the 3rd International Workshop on Green and Sustainable Software - GREENS 2014*, 2014, pp. 1–8.

[169] Rauth, I., Köppen, E., Jobst, B., and Meinel, C., “Design thinking: An educational model towards creative confidence,” presented at the DS 66-2: Proceedings of the 1st international conference on design creativity (ICDC 2010), 2010.

[170] Reips, U.-D. and Krantz, J. H., “Conducting true experiments on the Web.,” in *Advanced methods for conducting online behavioral research.*, Gosling, S. D. and Johnson, J. A., Eds. Washington: American Psychological Association, 2010, pp. 193–216.

[171] Robinson, A., *Sudden genius? the gradual path to creative breakthroughs*. Oxford, NY: Oxford University Press, 2010.

[172] Royce, W. W., “Managing The Development Of Large Software Systems,” in *Proceedings IEEE Wescon*, Aug. 1970, pp. 1–9.

[173] Ruparelia, N. B., “Software development lifecycle models,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 35, no. 3, pp. 8–13, May 2010.

[174] Salam, M. and Khan, S. U., “Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey,” *J. Softw. Evol. Process*, vol. 30, no. 8, p. e1939, Aug. 2018.

[175] Sapir, E., *Culture, language and personality selected essays*, 8. printing. Univ. of California Press, 1966.

[176] Schmidt, N.-H., Ruch, T. J., Decker, J., and Kolbe, L. M., “Ecosia.org: The Business Case of a Green Search Engine,” in *Green Business Process Management*, vom Brocke, J., Seidel, S., and Recker, J., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 217–232.

[177] Schön, E.-M., Thomaschewski, J., and Escalona, M. J., “Agile Requirements Engineering: A systematic literature review,” *Comput. Stand. Interfaces*, vol. 49, pp. 79–91, Jan. 2017.

[178] Schuler, H., “Communication rather than inspiration and perspiration?. On performance requirements in highly qualified occupations,” in *Competence and responsibility. The Third European Conference of the European Council for High Ability held in Munich (Germany)*, 1994, pp. 112–116.

[179] Schuler, H. and Görlich, Y., *Kreativität: Ursachen, Messung, Förderung und Umsetzung in Innovation [Creativity: Causes, Measurement, Promotion and Implementation in Innovation]*. Hogrefe Verlag, 2007.

[180] Schulz, A. and Schulz, B., *Perfect Scale*, 2nd edition. Munich: DETAIL -Institut für internationale Architektur- Dokumentation GmbH & Co. KG, 2016.

[181] Schwaber, K. and Sutherland, J., "The scrum guide," *Scrum Alliance*, vol. 21, no. 1, 2017.

[182] Selic, B., "Agile Documentation, Anyone?," *IEEE Softw.*, vol. 26, no. 6, pp. 11–12, Nov. 2009.

[183] Seyff, N., Betz, S., Duboc, L., Venters, C., Becker, C., Chitchyan, R., Penzenstadler, B., and Nobauer, M., "Tailoring Requirements Negotiation to Sustainability," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Aug. 2018, pp. 304–314.

[184] Shamshiri, H., "Supporting sustainability design through agile software development," in *Evaluation and Assessment in Software Engineering*, Jun. 2021, pp. 300–304.

[185] Simon, H. A., *The sciences of the artificial*, Third edition [2019 edition]. Cambridge, Massachusetts: The MIT Press, 2019.

[186] Sonnentag, S., Niessen, C., and Volmer, J., "Expertise in Software Design," in *The Cambridge Handbook of Expertise and Expert Performance*, 1st ed., Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., Eds. Cambridge University Press, 2006, pp. 373–388.

[187] Springer Fachmedien Wiesbaden, Ed., *HOAI 2013-Textausgabe =: HOAI 2013-text edition ; Honorarordnung für Architekten und Ingenieure vom 10. Juli 2013 = Official scale of fees for services by architects and engineers dated July 10, 2013*, 5., Vollst. aktualisierte Aufl. Wiesbaden: Springer Verlag, 2013.

[188] Stanik, C., Haering, M., and Maalej, W., "Classifying Multilingual User Feedback using Traditional Machine Learning and Deep Learning," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, Sep. 2019, pp. 220–226.

[189] Stettina, C. J. and Heijstek, W., "Necessary and neglected?: an empirical study of internal documentation in agile software development teams," in *Proceedings of the 29th ACM international conference on Design of communication - SIGDOC '11*, 2011, p. 159.

[190] Stevens, J. P., *Intermediate Statistics: A Modern Approach*. Routledge, 2007.

[191] Stevens, S. S., "Introduction: A Definition of Communication," *J. Acoust. Soc. Am.*, vol. 22, no. 6, pp. 689–690, Nov. 1950.

[192] Sy, D., "Adapting Usability Investigations for Agile User-centered Design," *J. Usability Stud.*, vol. 2, no. 3, pp. 112–132, 2007.

[193] Tang, A., Aleti, A., Burge, J., and van Vliet, H., "What makes software design effective?," *Des. Stud.*, vol. 31, no. 6, pp. 614–640, Nov. 2010.

[194] Thierry, G., "Neurolinguistic Relativity: How Language Flexes Human Perception and Cognition," *Lang. Learn.*, vol. 66, no. 3, pp. 690–713, Sep. 2016.

[195] Umber, A. and Bajwa, I. S., "Minimizing ambiguity in natural language software requirements specification," in *2011 Sixth International Conference on Digital Information Management*, Sep. 2011, pp. 102–107.

[196] United Nations, "Sustainable Development Goals." https://www.un.org/sustainabledevelopment/ (accessed Jul. 05, 2022).

[197] United Nations System Staff College, “Primer on Sustainable Development (English).” https://www.unssc.org/news-and-insights/resources/primer-sustainable-development-english (accessed Jul. 17, 2022).

[198] United Nations World Comission on Environment and Development, “Report of the World Commission on Environment and Development: Our Common Future.” UN Conference on Environment and Development, 1987.

[199] Usman, M., Mendes, E., and Börstler, J., “Effort estimation in agile software development: a survey on the state of the practice,” in *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, Apr. 2015, pp. 1–10.

[200] Vassigh, S., “Complexity of Sustainable and Resilient Building Design and Urban Development,” in *Sustainability and Social Responsibility of Accountability Reporting Systems: A Global Approach*, Çalıyurt, K. T. and Said, R., Eds. Singapore: Springer Singapore, 2018, pp. 65–79.

[201] Venkataraman, M. and Durg, K., “Research Idea on How Language and Symbols (Semantics and Semiotics) Affect Emotions of Software Engineers,” in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, Jun. 2020, pp. 577–580.

[202] Venters, C. C., “Software Sustainability: The Modern Tower of Babel,” in *CEUR Workshop Proceedings*, 2014, vol. 1216, pp. 7–12.

[203] Venters, C. C., Capilla, R., Betz, S., Penzenstadler, B., Crick, T., Crouch, S., Nakagawa, E. Y., Becker, C., and Carrillo, C., “Software sustainability: Research and practice from a software architecture viewpoint,” *J. Syst. Softw.*, vol. 138, pp. 174–188, Apr. 2018.

[204] Verdecchia, R., Lago, P., Ebert, C., and de Vries, C., “Green IT and Green Software,” *IEEE Softw.*, vol. 38, no. 6, pp. 7–15, Nov. 2021.

[205] Vetterli, C., Brenner, W., Uebernickel, F., and Petrie, C., “From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking,” *IEEE Internet Comput.*, vol. 17, no. 2, pp. 91–94, Mar. 2013.

[206] Von Carlowitz, H. C., *Sylvicultura oeconomica, oder haußwirthliche Nachricht und naturmaeßige Anweisung zur Wilden Baum-Zucht: Von allerhand Arten und Geschlechtern der innländischen Wilden Baeume [Home Economics Information and Natural Instructions for Wild Tree Cultivation: Of All Kinds and Sexes of the Domestic Wild Trees]*, vol. 1. Bey Johann Friedrich Brauns sel. Erben, 1732.

[207] Wagemans, F. A. J., Koppen, C. S. A. (Kris) van, and Mol, A. P. J., “The effectiveness of socially responsible investment: a review,” *J. Integr. Environ. Sci.*, vol. 10, no. 3–4, pp. 235–252, Dec. 2013.

[208] Wang, N. and Adeli, H., “Sustainable Building Design,” *J. Civ. Eng. Manag.*, vol. 20, no. 1, pp. 1–10, Mar. 2014.

[209] Whelan, T. and Kronthal-Sacco, R., “Research: Actually, Consumers Do Buy Sustainable Products,” 2019. https://hbr.org/2019/06/research-actually-consumers-do-buy-sustainable-products (accessed Jul. 06, 2022).

[210] Whorf, B. L., *Language, thought, and reality selected writings of Benjamin Lee Whorf*, 2. ed. MIT Press, 2012.

[211] Wiegers, K. E. and Beatty, J., *Software Requirements*, Third edition. Redmond, Washington: Microsoft Press, A Division of Microsoft Corporation, 2013.

[212] Winkel, O., “Higher education reform in Germany: How the aims of the Bologna process can be simultaneously supported and missed,” *Int. J. Educ. Manag.*, vol. 24, no. 4, pp. 303–313, May 2010.

[213] Winograd, T., *Bringing Design to Software*. ACM, 1996.

[214] Winograd, T., “From programming environments to environments for designing,” *Commun. ACM*, vol. 38, no. 6, pp. 65–74, Jun. 1995.

[215] Winograd, T. and Tabor, P., “Software Design and Architecture,” in *Bringing Design to Software*, Winograd, T., Ed. ACM, 1996.

[216] Wittgenstein, L., *Tractatus Logico-Philosophicus*. London: Kegan Paul, 1922.

[217] Wohlin, C., *Experimentation in Software Engineering*. New York: Springer, 2012.

[218] Wolff, P. and Holmes, K. J., “Linguistic relativity,” *Wiley Interdiscip. Rev. Cogn. Sci.*, vol. 2, no. 3, pp. 253–265, May 2011.

[219] Wooldridge, J. M., *Introductory econometrics a modern approach*, 5. ed., International ed. South-Western Cengage Learning, 2013.

[220] Zhong, V. J. and Schmiedel, T., “A User-Centered Agile Approach to the Development of a Real-World Social Robot Application for Reception Areas,” in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2021, pp. 76–80.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, den ____________________ Unterschrift:__________________________