NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

TIAGO ALEXANDRE LIBERATO MATOS OLIVEIRA

BSc in Computer Science and Engineering

A SUSTAINABILITY REQUIREMENTS CATALOGUE FOR THE ECONOMIC, ENVIRONMENTAL AND INDIVIDUAL DIMENSIONS

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

A SUSTAINABILITY REQUIREMENTS CATALOGUE FOR THE ECONOMIC, ENVIRONMENTAL AND INDIVIDUAL DIMENSIONS

TIAGO ALEXANDRE LIBERATO MATOS OLIVEIRA

BSc in Computer Science and Engineering

Adviser: João Baptista da Silva Araújo Júnior
Associate Professor, NOVA University Lisbon

Co-adviser: Ana Maria Diniz Moreira
Full Professor, NOVA University Lisbon

Examination Committee

Chair: Dr. Teresa Isabel Lopes Romão
Associate Professor, NOVA University Lisbon

Rapporteur: Dr. Isabel Sofia Sousa Brito
Coordinating Professor, Superior School of Technology and Management IPBeja

Member: Dr. João Baptista da Silva Araújo Júnior
Associate Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

A sustainability requirements catalogue for the economic, environmental and individual dimensions

Copyright © Tiago Alexandre Liberato Matos Oliveira, NOVA School of Science and Technology, NOVA University Lisbon.
The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.5) [1].

I would like to dedicate this dissertation to my family who unconditionally supported me throughout my life.

Acknowledgements

I would like to thank both of my advisors, Professor Ana Moreira and Professor João Araujo. They were fundamental to the success of my work and their guidance and dedication helped me during the many difficulties that I had during my time developing my thesis. I am glad that I had the opportunity to work on a topic that I genuinely believe will benefit society eventually in the future alongside colleagues and advisors that shared my enthusiasm. I would also like to thank my family for supporting me through my academic progress and for believing in me. It was thanks to them that I kept doing my best to improve at what I do and kept progressing through the course no matter the difficulties I faced.

”

“The greatest threat to our planet is the belief that someone else will save it.”

— Robert Swan

Abstract

Sustainability is increasingly recognized as a critical concern in software development, driven by challenges such as climate change, heightened social awareness, and economic pressures. Developers are adopting sustainability not only as an ethical imperative but also as a practical necessity, using proactive strategies to address these pressing issues. This dissertation centered on developing a sustainability catalog designed for the requirements engineering phase of software development. The research started with a systematic mapping study of existing tools, methodologies, and catalogs to identify widely recognized sustainability requirements and the current state of the art. These insights allowed us to create a streamlined catalog addressing sustainability across the individual, economic, and environmental dimensions. After creating our catalog, a supporting tool was developed to help use its information in real projects. The existing technologies found during our research aided the creation of a plugin for an already existing tool that helps use the catalog in real-world projects. After their creation, the catalog and tool were evaluated to assess their utility and impact. Feedback from this evaluation highlighted that combining the catalog and the supporting tool helps developers understand the sustainability implications of their requirements. The evaluation also revealed opportunities for refinement, as evidenced by several suggestions for improvement provided by participants. This dissertation contributed to the progression of software sustainability by introducing a comprehensive catalog and supporting tool, designed to serve as a foundation for integrating sustainability considerations into the software development process. These resources offer a practical framework to promote and implement sustainability in software projects.

Keywords: sustainability dimensions, sustainable development, requirements engineering, sustainable requirements, environmental dimension, individual dimension, economic dimension, sustainability catalog

Resumo

A sustentabilidade é cada vez mais reconhecida como uma preocupação crítica no desenvolvimento de software, motivado por desafios como as alterações climáticas, uma maior consciência social e pressões económicas. Os programadores estão a adotar a sustentabilidade não só como um imperativo ético, mas também como uma necessidade prática, usando estratégias proactivas para abordar estas questões prementes. Esta dissertação centrou-se no desenvolvimento de um catálogo de sustentabilidade concebido para a fase de engenharia de requisitos do desenvolvimento de software. A nossa pesquisa começou com um estudo de mapeamento sistemático de ferramentas, metodologias e catálogos existentes para identificar requisitos de sustentabilidade amplamente reconhecidos e o estado atual da tecnologia. Estes conhecimentos permitiram-nos criar um catálogo simplificado que aborda a sustentabilidade nas dimensões individual, económica e ambiental. Após a criação do nosso catálogo, foi desenvolvida uma ferramenta de apoio para ajudar a utilizar a sua informação em projectos reais. As tecnologias existentes encontradas durante a nossa investigação ajudaram na criação de um plugin para uma ferramenta já existente que ajuda a utilizar o catálogo em projectos reais. Após a sua criação, o catálogo e a ferramenta foram avaliados para determinar a sua utilidade e impacto. O feedback desta avaliação destacou que a combinação do catálogo e da ferramenta de apoio ajuda os programadores a compreender as implicações dos seus requisitos na sustentabilidade. A avaliação também revelou oportunidades de aperfeiçoamento, como evidenciado por várias sugestões de melhoria fornecidas pelos participantes. Esta dissertação contribuiu para a progressão da sustentabilidade do software ao introduzir um catálogo abrangente e uma ferramenta de apoio, concebidos para servir de base à integração de considerações de sustentabilidade no processo de desenvolvimento de software. Estes recursos oferecem uma framework prática para promover e implementar a sustentabilidade em projectos de software.

Palavras-chave: dimensões de sustentabilidade, desenvolvimento sustentável, engenharia de requisitos, requisitos sustentáveis, dimensão ambiental, dimensão individual, dimensão económica, catálogo de sustentabilidade

Contents

List of Figures x

List of Tables xii

1 Introduction 1
- 1.1 Context and Motivation . . . 1
- 1.2 Problem Statement . . . 1
  - 1.2.1 Challenges . . . 2
  - 1.2.2 Objectives . . . 3
- 1.3 Contributions . . . 3
- 1.4 Structure of the document . . . 3
- 1.5 Results . . . 4

2 Background 5
- 2.1 Requirements engineering . . . 5
  - 2.1.1 iStar 2.0 . . . 8
- 2.2 Sustainability . . . 9
  - 2.2.1 Sustainability dimensions . . . 9
  - 2.2.2 Sustainability requirements . . . 10
- 2.3 Systematic mapping studies . . . 11
  - 2.3.1 Definition of the research questions . . . 11
  - 2.3.2 Conducting the search . . . 12
  - 2.3.3 Screening the resulting studies and keywording of abstracts . . . 12
  - 2.3.4 Data Extraction and Mapping of Studies . . . 12
  - 2.3.5 Reporting . . . 12
- 2.4 Summary . . . 13

3 State of the Art 14
- 3.1 Research methodology . . . 14
  - 3.1.1 Planning phase . . . 14

3.1.2 Conducting stage . . . 17
3.1.3 Discussion of the results . . . 20
3.1.4 Threats to validity . . . 28
3.1.5 Results discussion . . . 29
3.2 Summary . . . 30

4 A Sustainability catalog **31**
4.1 Development and structure of the catalog . . . 31
4.1.1 Development of the catalog . . . 32
4.1.2 General structure of the catalog . . . 33
4.1.3 Categorization of the requirements . . . 34
4.1.4 Sustainability dimensions, requirements and catalogs . . . 37
4.2 The complete catalog . . . 40
4.3 Analysis of the catalogs . . . 40
4.3.1 The individual dimension . . . 41
4.3.2 The Economic and Environmental dimensions . . . 41
4.4 Summary . . . 42

5 Specification and supporting tool **43**
5.1 The piStar tool . . . 43
5.2 Plugin functionality and implementation . . . 44
5.2.1 Catalog within the plugin . . . 44
5.2.2 Implemented functionalities . . . 45
5.3 Application example . . . 54
5.3.1 ViaVerde system . . . 55
5.3.2 Choosing sustainability requirements . . . 55
5.4 Summary . . . 66

6 Evaluation **67**
6.1 Guides . . . 67
6.1.1 Video guide . . . 67
6.1.2 Written guide . . . 68
6.2 Questionnaire . . . 68
6.3 Participant selection . . . 69
6.4 Discussion of the Results . . . 70
6.4.1 Participant data . . . 70
6.4.2 Quality of the catalog . . . 72
6.4.3 Quality of the developed tool . . . 76
6.4.4 Guide evaluation . . . 78
6.4.5 Open feedback . . . 83
6.5 Threats to validity . . . 90
6.5.1 Internal validity . . . 91

6.5.2 External validity 91
6.5.3 Construct validity 92
6.5.4 Conclusion validity 92
6.6 Summary 92

7 Conclusion and Future work **93**
7.1 Contributions 93
7.2 Future work 94

Bibliography **96**

Appendices

A Data extraction form **103**

B Full sustainability catalog **104**

C ViaVerde example **106**

D Evaluation Questionnaire **108**

List of Figures

3.1 Economic dimension requirements present in the selected studies 23
3.2 Environmental dimension requirements present in the selected studies . . . 25
3.3 Individual dimension requirements present in the selected studies 27

4.1 Individual dimension sub-catalog 38
4.2 Economic dimension sub-catalog 38
4.3 First half of the economic dimension sub-catalog zoomed in 39
4.4 Second half of the economic dimension sub-catalog zoomed in 39
4.5 Environmental dimension sub-catalog 39
4.6 First half of the environmental dimension sub-catalog zoomed in 40
4.7 Second half of the environmental dimension sub-catalog zoomed in 40

5.1 Plugin button in the toolbar 45
5.2 Plugin's menu 46
5.3 Economic dimension selection 47
5.4 Individual dimension selection 47
5.5 Multi-dimension selection 48
5.6 Resulting model 48
5.7 Example of a list of relationships by type 49
5.8 Example of a list of relationships by dimension 50
5.9 Example of a list of refinements 50
5.10 Example of a tooltip in the plugin 51
5.11 First half of the ViaVerde example snippet zoomed in 64
5.12 Second half of the ViaVerde example snippet zoomed in 65

6.1 Percentage of participants by age 71
6.2 Percentage of participants by degree of education 71
6.3 Percentage of participants by degree type 72
6.4 Percentage of participants by years of experience 73
6.5 Participant opinions on the catalog's readability 73

6.6 Participant opinions on the catalog's utility 74
6.7 Participant opinions on the catalog's clarity 75
6.8 Participant interest in using the catalog 75
6.9 Participant opinions on the catalog's contribution towards understanding sustainability 76
6.10 Participant opinions on the tool's ease of use 77
6.11 Participant interest in using the tool in the future 77
6.12 Participant opinions on the tool's ability to ease the implementation of sustainability 78
6.13 Participant opinions on the text guide's clarity 79
6.14 Participant opinions on the video guide's clarity 79
6.15 Participant opinions on the text guide's quality 80
6.16 Participant opinions on the video guide's quality 80
6.17 Participant opinions on the video guide's quality 83

A.1 Data extraction form of the systematic mapping study 103

B.1 Complete sustainability catalog 105

C.1 ViaVerde example 107

List of Tables

2.1 PICOC structure 11

3.1 PICOC analysis table 15
3.2 Research sub-questions and their reasoning 16
3.3 Search string keywords 17
3.4 Inclusion and exclusion criteria 18
3.5 Results of the conduction stage 19
3.6 Selected studies 19
3.7 Economic dimension requirements 21
3.8 Environmental dimension requirements 24
3.9 Individual dimension requirements 26

5.1 Requirement definitions 52

1 Introduction

This work stemmed from growing concerns about sustainability, especially in Computer Science. While the importance of sustainability is recognized in this field, there's a lack of tools to help make software and software development more sustainable. The main goal of this paper was to make it easier for software developers to incorporate sustainability into their work, with a focus on economic, environmental, and individual aspects. This introduction lays the groundwork by providing context, explaining motivation, discussing challenges, and outlining the goals and expected outcomes of this research.

1.1 Context and Motivation

Over the past few decades, the consequences of insufficient sustainability measures have become increasingly apparent. Sustainability has shifted from a moral imperative to a critical necessity, particularly in a world marked by rapid change, substantial population growth, and the depletion of resources. Addressing sustainability has become crucial to ensuring the survival of future generations. In the context of our work, sustainability covers five key dimensions: environmental, social, economic, technical, and individual. This work addressed the growing need for sustainability by developing a requirements catalog that expands upon the sustainability catalog previously presented by Diogo Albuquerque, which focused on the social and technical dimensions [2]. The catalog we created addresses the three dimensions not covered in his work: individual, economic, and environmental. By building this catalog, our goal was to encourage the integration of sustainability into software development, contributing to the creation of a more resilient and sustainable future.

1.2 Problem Statement

This research was prompted by a gap in the existing methodologies, tools and reusable knowledge for creating sustainable software in software engineering. Finding problems early in software development makes them easier and cheaper to fix. While it's possible

to think about sustainability later on, it gets more difficult and more expensive due to the commitments made [3]. This can lead developers to give up on improving sustainability. So, it's crucial to consider sustainability early in the process to make sure it's implemented properly and not ignored as the project evolves. Many software developers also may not have an understanding of the impacts associated with their practices on the dimensions of sustainability. This lack of awareness makes it harder to integrate and prioritize sustainability concerns into their projects. Many software developers may lack awareness of how their practices affect sustainability, making it challenging to prioritize these concerns. Our problem statement, therefore, was **how we raise awareness of sustainability and what tools we could provide to help developers integrate sustainability into their software projects**.

1.2.1 Challenges

- **Challenge 1: Expose the current state of the art**
  To provide useful tools and raise awareness of sustainability, it's crucial to understand the latest methods, models and catalogs for making software projects more sustainable. To do this, we conducted a systematic mapping study on sustainability requirements in requirements engineering. The study involved an automated search on Google Scholar, a platform that compiles various online sources and libraries. This helped us gather insights from the most relevant sources on sustainability in software engineering.

- **Challenge 2: Identifying requirements that affect sustainability**
  As mentioned previously, sustainability is a broad concept that involves many aspects. It's crucial to identify factors contributing to or hindering sustainability, in our case requirements, and how they relate to the sustainability dimensions that we were studying. The systematic mapping study was a helpful tool in tackling this challenge as well as it allowed us to find as many requirements as possible along with their connections.

- **Challenge 3: Creating a generic catalog**
  A sustainability catalog should be as generic as possible to be applicable in a wide variety of situations. While it's possible to adapt a catalog from a specific problem domain to another to some extent doing so would heavily limit its effectiveness. The goal then was to develop a catalog that could be used by anyone involved in the requirements engineering phase of software development in an attempt to promote sustainability considerations as early as possible and help prevent it from being disregarded as a concern. This was the main challenge of the thesis.

- **Challenge 4: Tool support for the catalog**
  After developing our sustainability catalog, it was important to design a tool that facilitated the practical application of its knowledge in a more user-friendly manner,

for this the catalog had tool support implemented by extending the same tool already used by the catalog that addresses the technical and social dimensions.

1.2.2 Objectives

This dissertation addressed key challenges by gathering information on sustainability requirements related to the environment, economy, and individuals to create a comprehensive catalog. When combined with the existing catalog of social and technical sustainability requirements [2], this facilitated the use and reuse of these requirements. The catalog prioritized the most relevant requirements for various applications across our studied sustainability dimensions. Similar to previous research, the developed tool aimed to promote elicitation of sustainable software requirements in a user-friendly manner.

1.3 Contributions

This dissertation contributes to software development that takes into consideration sustainability. We hoped that by providing a tool and catalog that improve the ability to evaluate sustainability in the requirements engineering phase of development we can help move developers towards sustainable software solutions, as such we contributed with:

- A generic sustainability catalog that emphasizes the sustainability requirements that should be considered while creating software in a specific domain. This catalog also contains a sub-catalog for each dimension studied that addresses the properties of their specific dimension.
- A systematic mapping study on the sustainability requirements of the dimensions of environment, economy, and individual as well as sustainability requirement catalogs.
- Comparisons with several other tools and catalogs to understand how the current format of catalog can be improved as well as what the current catalogs lack the most.
- An evaluation of both the newly created sustainability catalog with the intent of gauging its ease of use usefulness.

1.4 Structure of the document

This document is structured as such:

- Chapter 2, Background, describes concepts present in the thesis highlighting their importance and relation with the work's objective and also gives a brief introduction to systematic mapping studies.

- Chapter 3, State of the art, presents the systematic mapping study and a discussion of its results, enabling an enhanced understanding of the current state of sustainability requirement catalogs.
- Chapter 4, Sustainability catalog, presents the sustainability catalog that was created taking into consideration the results of the research conducted during chapter 3.
- Chapter 5, Specification and supporting tool, describes the implemented supporting tool for using the sustainability catalog showing all of it's functionalities. It also has an application example where we applied our work to an already existing software project.
- Chapter 6, Evaluation, this chapter contains the assessment of our work and discussions regarding its results.
- Chapter 7, Conclusion and Future work, the last chapter exposes our conclusions and highlights opportunities for future research and improvements.

1.5 Results

We consider the work accomplished in this dissertation to be a meaningful success. Through our research, we developed a comprehensive catalog based on the requirements identified during our investigation, along with a tool designed to facilitate its practical application. The evaluation of these contributions provided valuable insights, demonstrating that our tools effectively support integrating sustainability principles into software development processes. Furthermore, they enhance users' understanding of how their software requirements influence the dimensions of sustainability that we focused on. While these outcomes highlight the potential of our approach to advance more responsible and sustainable software engineering practices, we recognize that there is room for improvement. The field of sustainability remains a rapidly evolving and expansive domain, presenting ongoing challenges and opportunities for further refinement and exploration.

2

Background

In this chapter, we provide essential background information and key concepts relevant to our dissertation. We will explain the process of requirements engineering and give some context on sustainability. Additionally, this section will provide an overview of systematic mapping studies as this method was used in the following chapter. This background information serves to establish an understanding of the setting of our research and the methods used during the requirements engineering phase of software development.

2.1 Requirements engineering

In the realm of software development, the success of a project hinges on the clarity, precision and alignment of its requirements [4]. If a project fails to align with the needs and perspectives of the client, its utility is significantly reduced. This is where requirements engineering plays an important role in the software development process. At its core, requirements engineering is to "discover, understand, formulate, analyze and agree on what problem should be solved, why a problem needs to be solved and who should be involved in the responsibility of solving that problem" [5]. By clearly defining and understanding the needs and expectations of all parties involved, requirement engineering establishes a solid foundation of the entire software development cycle by ensuring that the problem being solved and the method used to solve it are considered correct by both the developers and the stakeholders. Requirements engineering includes the following activities [4]:

- **Domain understanding:** Consists of studying the system-as-is within its organizational and technical context to obtain a good understanding of the domain in which the problem is rooted, and what the roots of the problem are.

- **Requirement elicitation:** Consists of discovering candidate requirements and assumptions that will shape the system-to-be, based on the weaknesses of the system-as-is as they emerge from the domain while acquiring information about the opportunities arising from the evolution of technologies and market conditions, improvements to the objectives of the system-to-be, organizational and technical constraints, boundaries between the software solution and the environment, typical scenarios involving the software, the assumptions necessary for the software to work properly.

- **Evaluation and agreement:** Exists to make informed decisions about issues raised during the elicitation process often based on what is considered the best trade-offs by the involved parties. It's in this activity that conflicting concerns are identified and resolved, the risks associated with the system are assessed, and requirement prioritization is performed.

- **Specification and documentation:** This activity consists of detailing, structuring and documenting the agreed characteristics of the system-to-be as they emerge from the evaluation activity which results in the requirements document, which will be used for a variety of purposes throughout the software's lifecycle.

- **Requirements consolidation:** This activity serves as quality assurance, where the specifications resulting from the specification and documentation activity are validated with stakeholders to pinpoint inadequacies, inconsistencies and omissions allowing errors to be treated at a decreased cost since the earlier in the development cycle it's found the cheaper it is to fix.

It's worth noting, however, that there is no ideal single process that is right for all organizations and that while each organization may base itself on currently accepted processes for requirements engineering it's common for organizations to adapt or create their processes based on the systems that they create, the organizational culture within, and the abilities of the people involved in the process [5].

Since the main objective was to make a catalog of reusable requirements, it's natural that the process of requirement elicitation would be one of the main focuses of this work. This activity is concerned with discovering requirements, analyzing them for incompleteness, inconsistency, relevance and practicality as well as negotiating the final requirements for the system [5]. During this process we obtain the following information for the system-to-be [4]:

- Technologies and market conditions exist that could address the weaknesses of the system-as-is while preserving its strengths.

- Improvements to the objectives that the system-to-be should meet regarding weaknesses and opportunities and alternative options for satisfying them.

- The constraints that this system should take into account.

- Alternative approaches that can be considered when delegating which tasks will be handled by the system-to-be and which will be the responsibility of the environment that contains it.

- Typical scenarios illustrating desired interactions between the software-to-be and its environment.

- The essential properties of the domain and the assumptions about the environment that allow the software-to-be to work properly.

- Finally as mentioned before the requirements that the software-to-be should meet.

It is also worth mentioning that many common techniques are used during requirement elicitation to obtain all of this information each with its weaknesses and strengths. These techniques can be divided into categories: traditional, collaborative, cognitive and observational [6]. There are too many to list however these are some examples of techniques that are frequently used [6]:

- **Questionnaire:** An example of a traditional technique for requirement elicitation, this technique elicits requirements through a large number of people which results in less cost and less time spent. Well-designed questionnaires are useful to elicit consistent and unambiguous requirements from stakeholders but badly designed ones may induce a bias or give inaccurate results so its success is highly dependent on how well the questionnaire itself is designed.

- **Data gathering from an existing system:** It is a traditional technique wherein knowledge is collected by looking at an existing system. It's possible to collect a wealth of knowledge and requirements through this technique but incurs the risk of creating extra restraints if not done correctly.

- **Brainstorming:** It is our first example of a collaborative technique, it's based on providing an open environment for discussion where the participants are free to expose their ideas and expectations for the system which are then discussed and analyzed.

- **Use cases/Scenarios:** It is another collaborative technique that gives information on the interactions between users and the system. Each use case represents a functional requirement of the system and scenarios are examples of interactions between the user and the system. These scenarios should include information regarding the state of the system as the scenario starts, the state of the system after the completion of the scenario, what activities might be simultaneous, the normal flow of events and what exceptions to the events.

- **Document Analysis:** It is a cognitive technique where documents related to the problem domain are analyzed to gather information that flows within the organization often a useful technique to find information about a particular task.

- **Protocol Analysis:** Another cognitive technique where a meeting is conducted with the stakeholders to discuss the requirements of the system providing the required actions for fulfilling the user requirements.

- **Observation:** This technique involves the observation of a user's work and taking notes on them. Observation can be direct or indirect but allows the observer to view what users do in a real context overcoming potential problems caused by idealized or oversimplified stakeholder descriptions.

2.1.1 iStar 2.0

The iStar 2.0 (i* 2.0) modeling framework is a goal and actor oriented framework [7] designed to represent and analyze software systems by focusing on the relationships between actors and their goals. As an evolution of the original i* framework, iStar 2.0 is often applied in areas like requirements engineering offering an approach to identifying interactions and dependencies between stakeholders. The key elements of iStar 2.0 are [7]:

- **Actors** - entities aim to achieve their goals by leveraging their knowledge while collaborating with other actors. These entities can be further categorized into roles and agents. Roles represent an actor within a specific context, such as a student, while agents refer to actors with tangible physical manifestations, exemplified by institutions like FCT NOVA University.

- **Actor association links** - links that describe relationships between actors and can be categorized into two types: "participates-in" relationships, which indicate involvement in a particular context or activity, and "is-a" relationships, which define a hierarchical classification or type of actor.

- **Intentional elements** - these elements represent anything that actors seek to obtain or accomplish and are categorized as goals, qualities, tasks, and resources. Goals specify a state that an actor aims to achieve, accompanied by clear criteria for achieving it. Qualities refer to desirable attributes, with ambiguous achievement criteria. Tasks represent the actions that an actor wishes to execute, while resources encompass the physical or informational entities necessary for performing these tasks.

- **Social dependencies** - represent the social relationships between actors and their requirements, indicating that an element of a depender relies on an element of a dependee for to function properly or successful accomplish their objective.

- **Intentional element links** - lastly, intentional element links are classified into four types: refinement, needed-by, contribution, and qualification. Refinement links are used to decompose higher-level elements into specific sub-elements. Needed-by links indicate that one element must be completed before another can begin. Contribution links illustrate how intentional elements contribute to the achievement of other intentional elements, reflecting either a positive impact (help, make) or a negative impact (hurt, break). Finally, qualification links specify how an operation should be carried out, adding conditions or constraints to the intentional elements.

This framework is significant because it is widely utilized and serves as the foundation for our work, as it serves as the basis of one of the tools we use during the process of requirement elicicitation.

2.2 Sustainability

Sustainability, as defined by the UN in 1987, entails "meeting the needs of the present without compromising the ability of future generations to meet their own needs" [8]. To address these challenges, diverse measures have already been implemented, initiatives such as sustainability reports [9] and international treaties [10]. However, despite these efforts, achieving true sustainability remains a formidable task. The expansive nature of sustainability, even within the framework provided by the UN, the SDGs [11], demonstrates its complexity. In response to sustainability's complexity, it has been deconstructed into dimensions that better capture its diverse needs and challenges. One widely accepted decomposition involves five dimensions: environmental, individual, social, economic, and technical ([12]). Software that we build can affect any of these dimensions, both positively and negatively which brings us to what motivated this work: ensuring that developers have another tool that they can use to improve the sustainability during the requirements elicitation phase of their projects. An attempt has been made in the past to create a catalog of sustainability requirements [2]. This catalog only covers the technical and social sustainability dimensions, however, and none exist currently that cover the remaining three dimensions.

2.2.1 Sustainability dimensions

For this work it was important to define what the sustainability dimensions represent to be able to understand what our goal is. These dimensions can be described as shown below ([13]):

- **Individual sustainability:** Is concerned with individual freedom and agency, human dignity, and fulfillment allowing individuals to thrive and develop freely.
- **Social sustainability:** Consists of maintaining relationships between individuals and groups, in an attempt to balance conflicting social interests.

- **Economic sustainability:** Are the concerns related with financial and business aspects, often associated with business growth and financial operations.
- **Environmental sustainability:** Is described as the use and management of natural resources, often associated with the management of energy consumption and waste production, preserving the balance of the ecosystems and preventing climate change.
- **Technical sustainability:** Is described as the ability to ensure that artificial systems can evolve and be maintained over time.

2.2.2 Sustainability requirements

While we consider sustainability to be an essential requirement that should be prioritized, the current landscape of software development often treats it as a non-functional requirement [14]. Sustainability requirements in the context of software development are, therefore, non-functional requirements that are related to and affect sustainability. They can be threats or opportunities with effects that chain between each other and their dimensions with varying orders of effect [15]. As explained, sustainability encompasses five dimensions: environmental, social, economic, individual and technical. Each sustainability requirement corresponds to one or more dimensions, addressing specific concerns related to those dimensions. These requirements have effects that vary in quickness of effect; while the impact of some requirements is immediate, others may only have a long term impact. Systems that produce short-term benefits can sometimes lead to negative long-term consequences, emphasizing the need to consider all potential effects when developing software systems. The orders of effect are categorized as [15]:

- **Immediate** effects that are felt right away as the action is taken. An example of these would be an application that gives users access to coupons that give them a way to save money as soon as they start using said application.
- **Enabling** effects are, usually, a bit trickier to define, their effects are not felt immediately but don't take a long time to be seen. An example of this could be how some applications can inspire an increase in market competitiveness, while not felt right away it doesn't take a long time for it to be felt.
- **Systemic** effects are the ones that are felt only after a long time of the system's use. An example of this effect would be how software can influence positively or negatively energy consumption after a very long time has passed.

A great example of a sustainability requirement is "the software must optimize energy consumption over a long period of time.". The impact of this requirement would be systemic, could affect one or more sustainability dimensions, with the most obvious being the environmental dimension. This requirement would serve as an objective for the system to accomplish during development. Incorporating sustainability requirements

into the development process ensures that sustainability is taken into consideration early in the development cycle, serving as a guide for developers to integrate sustainability considerations into their decision-making and ensuring that the project aligns with the principles of sustainability.

2.3 Systematic mapping studies

A systematic mapping study is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest to identify any gaps in current research to suggest areas for further investigation, or to summarize the existing evidence concerning technology, or to provide a framework/background to appropriately position new research activities. This type of study allows us to reduce the likelihood that a study will be biased, combine data and provide evidence that a phenomenon is robust and transferable if it the studies analyzed provide consistent results [16]. These studies have five essential steps: definition of the research questions, conducting the research, screening the resulting studies and keywording of abstracts, and data extraction and mapping of studies [17]. These steps will be explained in the following subsections.

2.3.1 Definition of the research questions

The definition of the research questions is one of the most important parts of a systematic study [16] since they are responsible for giving a clear direction and focus to our research, without well-defined questions one may end up exploring irrelevant topics. Research questions can be structured according to a method called PICOC which is a method that defines what a research question's population, intervention, comparison, outcome and context will be. In Chapter 3 we will be using this structure allowing us to create more focused research questions. The PICOC structure can be described as such [18]:

Table 2.1: PICOC structure

Population	Group that we are interested in appealing to.
Intervention	Technology or process used to address a particular issue.
Comparison	Target that will be compared with the intervention.
Outcome	Significant outcomes resulting from the investigation.
Context	Context in which the investigation was performed focusing on what factors were present that may impact positively or negatively the intervention.

2.3.2 Conducting the search

The search for the primary studies can be performed automatically, by using search strings in digital libraries and manually by browsing through studies from reputable sources such as journal publications [17]. Search strings are a set of words that are based on our research questions [17] and are connected by logic operators that specify whether they should have an "AND" or "OR" operator. The "AND" operator ensures that all within the operator exist within the result and the "OR" operator ensures that at least one exists. By performing these searches we will obtain several studies that are seemingly relevant to the work which will then be screened in the next phase of the study.

2.3.3 Screening the resulting studies and keywording of abstracts

During this phase, a set of inclusion and exclusion criteria are created. These inclusion and exclusion criteria will define whether a research study will be considered relevant. To be considered relevant a study must comply with all of the inclusion criteria while simultaneously not having any of the exclusion criteria [17]. For automatic studies we have two steps to this process, the first being to read the title and the abstract and assess whether or not it's relevant, should a study's title and abstract not be enough to determine this it passes on to the next phase of the process. During the second phase, the studies selected in the first iteration are read in full to ensure that the studies selected will help with obtaining answers to our research questions.

2.3.4 Data Extraction and Mapping of Studies

In this step, the data extraction takes place [17]. After being sorted their relevant data is then extracted according to a form. This data includes relevant information to our research questions and general information such as the title, authors and year of publication. The analysis that follows focuses then on verifying the categories of the publications allowing us to see which ones have been the most emphasized in previous research and what gaps or opportunities for future research exist [17].

2.3.5 Reporting

In the final phase, the main findings are presented and directly linked to the research questions. These findings are used to answer the research questions, based on the data extracted from the selected studies. Visual tools such as charts or tables are used employed to identify trends, gaps, and the distribution of research across different categories. The reporting phase also addresses any potential biases or limitations. This step is crucial for highlighting key insights, uncovering research gaps, and offering recommendations for future research.

2.4 Summary

In this chapter, we introduced the fundamental concepts for our dissertation. We began by discussing requirements engineering, followed by a brief overview of the iStar 2.0 framework. Next, we explored the concept of sustainability, its importance, and provided an explanation of the sustainability dimensions and sustainability requirements. Finally, we outlined the process of conducting a systematic mapping study, as carried out in Chapter 3, to gain insight into the state of the art in sustainability and the tools available to support its integration in software projects.

3

State of the Art

The changing landscape of requirements engineering and sustainability requirements reflects the quick progress in technology in software engineering. Significant improvements have been made in how we approach these areas to handle challenges from different situations and the growing need for high-quality, reliable and suitable software. It's crucial to have an understanding of the current state of requirements engineering to address the needs that our catalog aims to fulfill. This was done through a systematic mapping study, which allowed us to see the current methods, tools and frameworks that meet the demands of sustainability in software currently, as well as what requirements are most mentioned for each of the sustainability dimensions that we are studying.

3.1 Research methodology

Our research methodology was aimed at understanding the current state of the tools, methodologies and catalogs or lists that may help developers implement sustainability during the software engineering process. This involved conducting a three-stage systematic mapping study, the phases being: the planning phase, the conducting phase and the reporting phase.

3.1.1 Planning phase

This first phase of the systematic mapping study includes five sub-phases, all of which will be explained next.

3.1.1.1 Definition of the research question

To define our research questions we used the **PICOC** method, this method states that five criteria should be considered when building our research questions. By taking into consideration these five criteria (population, intervention, comparison, outcome and context) we can ensure that the we are creating the correct questions to solve the challenges of our work. Table 3.1 has these criteria defined in the context of our study.

Table 3.1: PICOC analysis table

Population	Scientific papers addressing sustainability requirements with proposals of catalogs, or other tools that address the need for these requirements.
Intervention	Searching in digital libraries for existing catalogs, tools or lists of sustainability requirements.
Comparison	Comparisons will be done between existing methodologies that are found during the search to gather as many of the sustainability requirements as possible.
Outcome	An overview analysis of the current approaches and tools that exist to help the elicitation of sustainability requirements.
Context	Preparation of a Master Thesis in Computer Science and Informatics Engineering.

This led us to create the following research question:
What are the existing methodologies that support software sustainability and sustainable requirements that cover the economic environmental and individual dimensions?
After obtaining the main research question, it was decomposed into four other sub-questions visible in Table 3.2, each with a different motivation behind its creation. By decomposing the main research question into several sub-questions we were able to analyze and categorize information according to the context of each of the questions which would then allow us to view which areas presented gaps or opportunities.

Search strategy

A systematic study's search strategy serves as a base to help identify and retrieve relevant studies and research to our research questions. The search strategy focused on the automatic due to the very large number of results. The automatic search is done by creating a search string using all the keywords present in Table 3.3.

Our automated search was conducted on Google Scholar, which scans numerous digital libraries and academic publishers simultaneously, including well-known ones such as IEEE, which led to a very large number of results, even with highly restrictive queries. It is important to highlight that the table's syntax does not align with Google Scholar's. To adapt it, we replaced *AND* with a space character, the *OR* with the vertical bar "|" and if a *NOT* was present a "-" would replace it. Following the automated search, the decision was made that a manual search was unnecessary as the results obtained were extensive enough that the results from a manual search were likely to not add anything of value and there may not have been enough time to process all articles had it been performed.

Table 3.2: Research sub-questions and their reasoning

Sub-question	Reason
RQ1. What are the tools and catalogs that exist for implementing sustainability in a software project?	While we are looking at three specific dimensions, a more general look at our topic may yield information that more focused approaches may not.
RQ2. What are the sustainability requirements in the economic dimension?	Allows the discovery of what sustainability requirements exist and are more often considered in the economic dimension.
RQ3. What are the sustainability requirements in the environmental dimension?	Allows the discovery of what sustainability requirements exist and are more often considered in the environmental dimension.
RQ4. What are the sustainability requirements in the individual dimension?	Allows the discovery of what sustainability requirements exist and are more often considered in the individual dimension.
RQ5. What are the relations between sustainability requirements in the studied dimensions?	Allows the discovery of positive and negative relations exist between requirements in the economic, environmental and individual dimensions.

Study selection strategy

The studies identified through our search underwent an evaluation based on predefined inclusion and exclusion criteria which determined whether or not a retrieved study was relevant to the questions that we had. This evaluation was executed in two iterations. Initially, during the first iteration, the title and abstract of each study were analyzed, and any study that met all inclusion criteria and none of the exclusion criteria was accepted for the next iteration as candidates for selection. In the second iteration, the full text of the candidates belonging to the reduced list from the first iteration was read. Similar to the first iteration, only studies that met all the inclusion criteria and none of the exclusion criteria were accepted as studies that were relevant to our search. The inclusion and exclusion criteria are outlined in Table 3.4:

Table 3.3: Search string keywords

Concept	Keywords
Methodologies	"model" OR "tool" OR "framework" OR "catalog" OR "metamodel" OR "ontology" OR "procedure"
Requirements engineering	"requirements engineering"
Software sustainability	"software sustainability"
Economic dimension	"economic"
Individual dimension	"individual"
Environmental dimension	"environmental"

3.1.1.2 Quality assessment of the studies

To ensure reliable outcomes in empirical studies we need to be sure of the quality of our research [19]. To assess this we took into consideration two factors in each study: the number of citations recorded in the digital library and its CORE Rank. The former was prioritized as a quality indicator because if a study has a high number of citations, then it is somewhat safe to assume that it's more trustworthy. The CORE Conference Ranking assigns a rank to conferences taking into consideration several indicators [20], with four possible rankings: C means that it meets the minimum standards, B is good, A is great and A* means that it's among the best. None of the studies were discarded due to the quality assessment as all passed the criteria.

3.1.1.3 Data collection and extraction

To enhance clarity and facilitate the extraction of information, we used an Excel spreadsheet to consolidate all the identified studies throughout both the first and the second iteration of the study selection. The document was structured with dedicated columns designated for specific types of information, such as title, year, authors, and columns specifically to address the research questions. Additional columns were also used to mark each of the inclusion and exclusion criteria that each study achieved. By organizing our data this way it was easier to get an overview of our gathered data.

3.1.2 Conducting stage

The search was conducted by searching through Google Scholar using the search string formulated during our planning phase. Our initial search yielded 417 potential studies,

Table 3.4: Inclusion and exclusion criteria

ID	**Study inclusion criteria**
IC1.1	The article is available in English.
IC1.2	The article is from a conference, workshop or journal
IC1.3	The article is available in full text
IC1.4	The title and/or abstract of the article is related to one of the research questions
IC2.1	The article talked about a tool, model, metamodel or catalog of sustainability requirements that could be useful for the creation of our own catalog
ID	**Study exclusion criteria**
EC1.1	The article is a duplicate
EC1.2	The article is informal (slides, extended abstracts, blogs)
EC1.3	The article is an introduction to special issues, books, workshops or posters
EC1.4	The title and/or abstract does not relate to any of the research questions
EC2.1	The request for a full-text of the document was not fulfilled on time
EC2.2	The article did not talk about a tool, model, metamodel or catalog of sustainability requirements that could be useful for the creation of our own catalog

this number was then refined to 88 after the first iteration concluded. It's worth noting that some results, including citations and articles lacking essential information such as titles, were cataloged during the first phase but immediately excluded due to their lack of usefulness and dubious origin. Moving into the second iteration, our objective was to further narrow down the studies by reading through them in full. Following this more in-depth analysis, 38 articles remained accepted by the time the process concluded on January 18, 2024. The outcomes of these iterations are presented in Table 3.5.The full list of seledcted studies can be seen in Table 3.6 and the data extraction form can be viewed in Appendix A.

Table 3.5: Results of the conduction stage

	First Iteration Studies
Exclusion criteria	417
EC1.1	-33
EC1.2	-2
EC1.3	-88
EC1.4	-55
EC1.5	-151
Total	88
	Second Iteration Studies
EC2.1	-5
EC2.2	-45
Total Selected	**38**

Table 3.6: Selected studies

Reference number	Study title
[15]	Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow
[21]	Characterizing the contribution of quality requirements to software sustainability
[2]	A social and technical sustainability requirements catalogue
[22]	Do we Really Know What we are Building? Raising Awareness of Potential Sustainability Effects of Software Systems in Requirements Engineering
[23]	Towards a Software Sustainability-Quality Model: Insights from a Multi-Case Study
[24]	Integrating Environmental Sustainability in Software Product Quality
[25]	Sustainability Quantification in Requirements Informing Design
[3]	A Green Model for Sustainable Software Engineering
[26]	Uncovering sustainability concerns in software product lines
[27]	Designing for Sustainability: Lessons Learned from Four Industrial Projects
[28]	ShapeRE: Towards a Multi-Dimensional Representation for Requirements of Sustainable Software
[29]	Interactions between environmental sustainability goals and software product quality: A mapping study
[30]	A Catalogue Supporting Software Sustainability Design
[31]	Incorporating sustainability into software projects: a conceptual framework
[32]	Sustainability requirements for eLearning systems: a systematic literature review and analysis
[33]	Requirements engineering for sustainable software systems: a systematic mapping study
[34]	Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects

Continued on next page

Table 3.6 – continued from previous page

Reference number	Study title
[35]	BERT-Based Approach for Greening Software Requirements Engineering Through Non-Functional Requirements
[36]	SQ-Framework for Improving Sustainability and Quality into Software Product and Process
[37]	GOSSEC: Goal Oriented Software Sustainability Evaluation Criteria
[38]	Sustainability requirements for connected health applications
[39]	Systematic mapping study on software engineering for sustainability (SE4S)
[40]	Towards a Systematic Process for the Elicitation of Sustainability Requirements
[41]	Karlskrona Manifesto: Software Requirement Engineering Good Practices
[42]	Sustainability debt: a portfolio-based approach for evaluating sustainability requirements in architectures
[43]	A concern-oriented sustainability approach
[44]	Classifying the Measures of Software Sustainability
[45]	An Exploration of Sustainability Thinking in Research Software Engineering
[46]	Software, Sustainability, and UN Sustainable Development Goals
[47]	Labelling Sustainable Software Products and Websites: Ideas, Approaches, and Challenges
[48]	A Unified Fuzzy-Based Symmetrical Multi-Criteria Decision-Making Method for Evaluating Sustainable-Security of Web Applications
[49]	Identification of Sustainability Characteristics and Sub-Characteristics as Non-Functional Requirement for Requirement Change Management in Agile
[50]	Green architecture for sustainable eLearning systems
[51]	A Comparison of Software Quality Characteristics and Software Sustainability Characteristics
[52]	Identification, Selection and Prioritization of Sustainable and Green Software Factors Using Fuzzy Set theory
[53]	Toward a Comprehensive Understanding and Evaluation of the Sustainability of E-Health Solutions
[54]	A Requirements Catalog of Mobile Personal Health Records for Prenatal Care
[55]	Quality Characteristics of Mobile Applications: A Survey in Brazilian Context

3.1.3 Discussion of the results

Having concluded our search we proceed to the discussion of the obtained results. Each of the following sub-sections is dedicated to analyzing the outcomes of this search in the context of our research question and its sub-questions. It's crucial to note once again that sustainability is a very broad subject, and because of that it is nearly impossible to explore in its entirety, especially in the context of a master's thesis. Therefore the findings related to our questions about what sustainability requirements exist (RQ1, RQ2, RQ3 and RQ4) will not be entirely complete. However, they represent the most acknowledged and frequently mentioned within our selected studies.

RQ1 - What are the tools and catalogs that exist for implementing sustainability in a software project?

The studies that we found showed us several tools and catalogs that help software developers implement sustainability in software projects. The following list is what we found:

- Frameworks [15][26][22][27][28][31][33][36],
- Models such as ISO/IEC 25010, Goal models as well and other model types [21][23][24][25][33][34][42],
- A manifesto called the Karlskrona Manifesto [30][33][40],
- Approaches such as INSURE/ENSURE, Crowd-focused/Seemi-automate, Negative UX-based Discovery, and AHP/ANP [33],
- Metrics that measure sustainability or how "green" software is [25][3][37],
- Language models such as BERT [35],
- General tools such as Fuzzy-AHP [48],
- Catalogs, lists, and checklists [2][30][33][38][44][40][56].

This lead us to the conclusion that there are many distinct tools and catalogs available that allow developers to integrate sustainability more easily into their projects, with some being generic [2], and some being domain-specific [38].

RQ2 - What are the sustainability requirements in the economic dimension?

In the studies that were selected, we found several requirement categories that are related to the economic dimension, these are seen in Table 3.7 .

Table 3.7: Economic dimension requirements

Documents references	Requirements
[21][27][37]	Effectiveness/impact
[21]	Availability
[21][49] [27][37]	Usefulness/usability
[49][21][27]	Risk mitigation
[21]	Recoverability
[49][21][27][37]	Functional appropriateness/suitability
[21][27]	Trust
[21]	Functional correctness
[21][27]	Flexibility
[21]	Completeness
[49][21][27][28][37]	Efficiency

Continued on next page

Table 3.7 – continued from previous page

Document references	Requirements
[21]	Functional completeness
[53][22][28][30][31][40][42]	Value
[22]	Supply chain management
[49][22][40]	Customer relationship management/satisfaction
[22]	Governance and processes
[22][40]	Innovation
[26]	Increased productivity/optimization
[26]	Time to market reduction
[53][50][26][40]	Business improvement/growth
[45][49][27][37][42]	Maintainability/longevity/supportability
[27]	Replaceability
[27]	User error protection
[50][28][30][40][42]	Cost minimization
[30]	Creation of new business opportunities
[31]	Business agility
[37][43]	Compatibility
[49][37]	Reliability/dependability
[37]	Security
[37][49]	Portability
[45]	Interdisciplinarity
[49]	Modifiability
[49]	Reusability
[49]	Performance
[49]	Predictability
[49]	Context coverage
[49]	Extensibility
[49]	Interoperability
[49]	Scalability

These results can be seen in a more visual format in the chart in Figure 3.1.

By looking at the list of requirements we can conclude that many of the requirements that are the most considered in these tools are related to creating value for the company and their customers, cost minimization for the company, company growth, maintainability, efficiency, usefulness/usability and appropriateness. The most striking omission is a lack of consideration for the economic dimension of the user, most of the economic perspectives present are focused on how the company can sustain itself with little consideration for the user's side. This is not to say that sustainability implementations do not consider this as pricing and monetization surely makes a company consider it, but there's certainly an opportunity for expansion in the economic requirements from the user's perspective.

RQ3 - What are the sustainability requirements in the environmental dimension?

This dimension, being the one most closely related to sustainability, had a great number of requirements made to implement it into systems the requirements found are in Table 3.8.

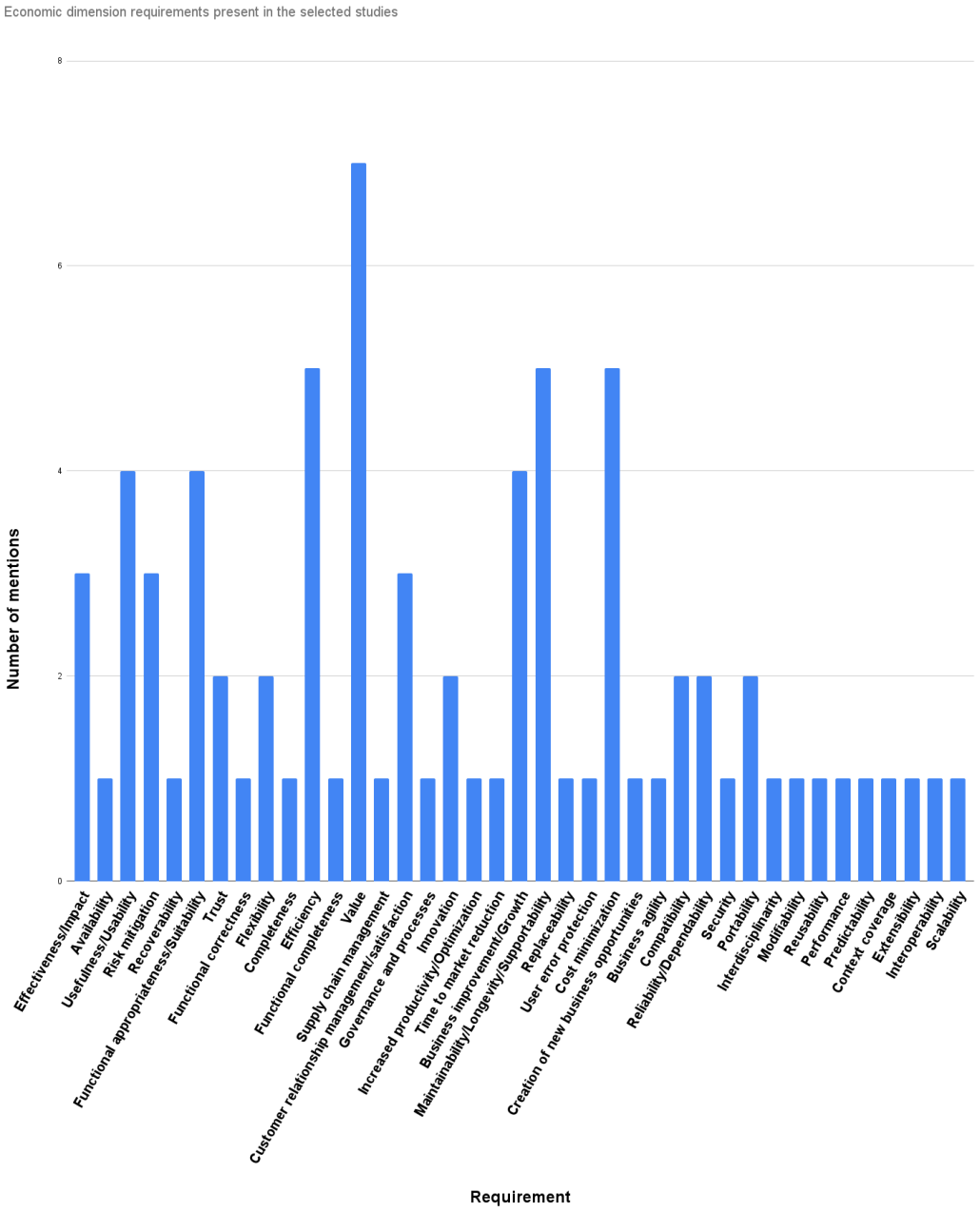


Figure 3.1: Economic dimension requirements present in the selected studies

Table 3.8: Environmental dimension requirements

Documents references	**Requirements**
[21][44]	Reusability
[37][35][27][23][21][44][52]	Performance
[27][21][49]	Modifiability
[21]	Co-existence
[21]	Availability
[42][23][21]	Environmental risk mitigation
[21]	Time behavior
[35][28][26][22][23][44]	Material and resource management/optimization
[22]	Soil, atmospheric, and water pollution reduction
[22]	Promotion of biodiversity and land use
[42][30][28][22][26][31][35][44][52]	Energy efficiency/energy consumption
[31][22]	Management of logistics and transportation
[23][37][44]	Maintainability
[27]	Flexibility
[27]	Effectiveness
[27]	Economic risk mitigation
[27]	Functional appropriateness
[27]	Functional correctness
[27]	Replaceability
[27]	Trust
[27]	Usefulness
[27]	User error protection
[43][42][31][28]	Ecosystem protection
[32]	Greenability
[35]	Infrastructure management and optimization
[35][52]	Reduction of disposal and waste
[37][44]	Functional suitability
[37]	Compatibility
[37]	Portability
[37][49]	Impactibility
[44]	Environmental footprint reduction
[44]	Supportability
[44]	Capacity optimization
[44]	Predictability
[44]	Usability
[44]	Accessibility
[44]	Perdurability
[44]	Dependability
[44]	Integrity
[44]	Evolvability
[44]	Freedom from risk
[44]	Satisfaction
[44]	Scalability
[44]	Extensibility
[44]	User conformity

Continued on next page

Table 3.8 – continued from previous page

Document references	Requirements
[44]	Interoperability
[52]	Distribution or deployment of software
[52]	Dematerialization
[52]	Process efficiency

This list can be seen in chart format in Figure 3.2. After an analysis of this list, we can

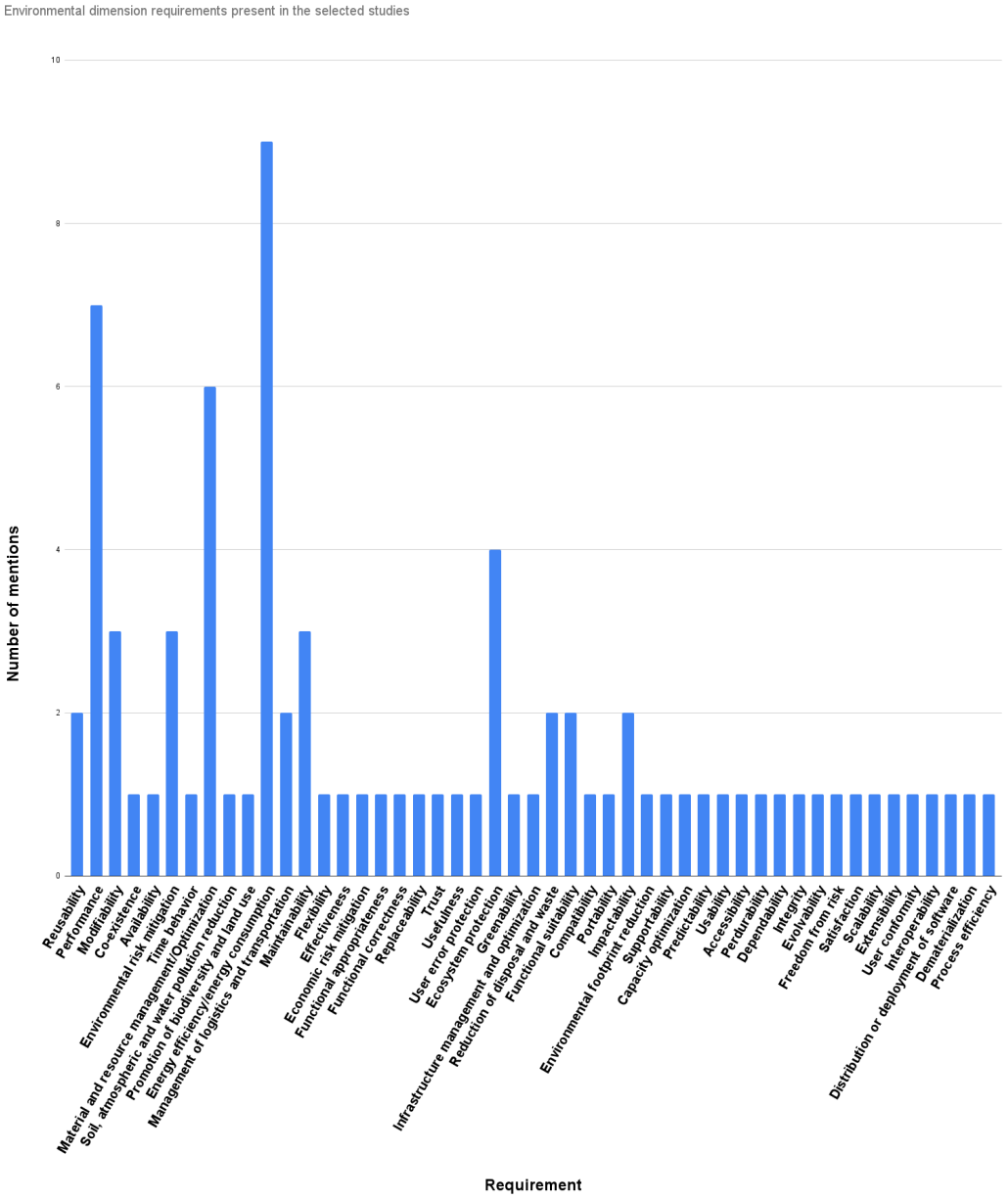


Figure 3.2: Environmental dimension requirements present in the selected studies

observe that the environmental dimension has some of the most agreed-upon requirements having more studies mentioning them than the other two dimensions, there was a heavy focus on resource management/optimization, performance, reduction of energy consumption and protection of the ecosystem. One of the papers stated through their list that all sustainability requirements that belong to the environmental dimension since the requirements of other dimensions are all included in the environmental dimension. We discarded this notion due to going against previously shown definitions and conventions for software requirements and sustainability requirements. The dimensions have been divided this way because as stated in Chapter 2, sustainability is a large subject and the division helps focus our efforts in certain areas of this topic, by assuming this paper's idea we'd be complicating an already complicated process, none the less these were included as there is some intersection between dimensions and it may help when attempting to find this intersection. One other observation that can be made is that while it is mentioned, infrastructure and electronic device disposal were mentioned very little, which we found to be an aspect that can be tackled in the future since infrastructure such as data centers are becoming an increasing concern when it comes to the sustainability of applications that require them due to their impact on the environment [57].

RQ4 - What are the sustainability requirements in the individual dimension?

Out of the three dimensions studied this one was the one that was the least considered, still some requirements that were found are in Table 3.9.

Table 3.9: Individual dimension requirements

Documents references	**Requirements**
[22]	Health
[22][50]	Privacy
[22][50][45]	Lifelong learning/education
[22]	Safety
[22][28][40]	Agency
[28][40]	Freedom
[28][40]	Human dignity
[42][28][40]	Fulfillment/satisfaction
[40]	Trust in the system
[40]	Equality of access to the system
[40]	Fair treatment of the user
[42]	Time and resource optimization
[44]	Usability
[50]	Personalization
[50]	Collaboration
[50]	Leadership development
[50]	Security

Figure 3.3 shows the chart for the requirements of this dimension.

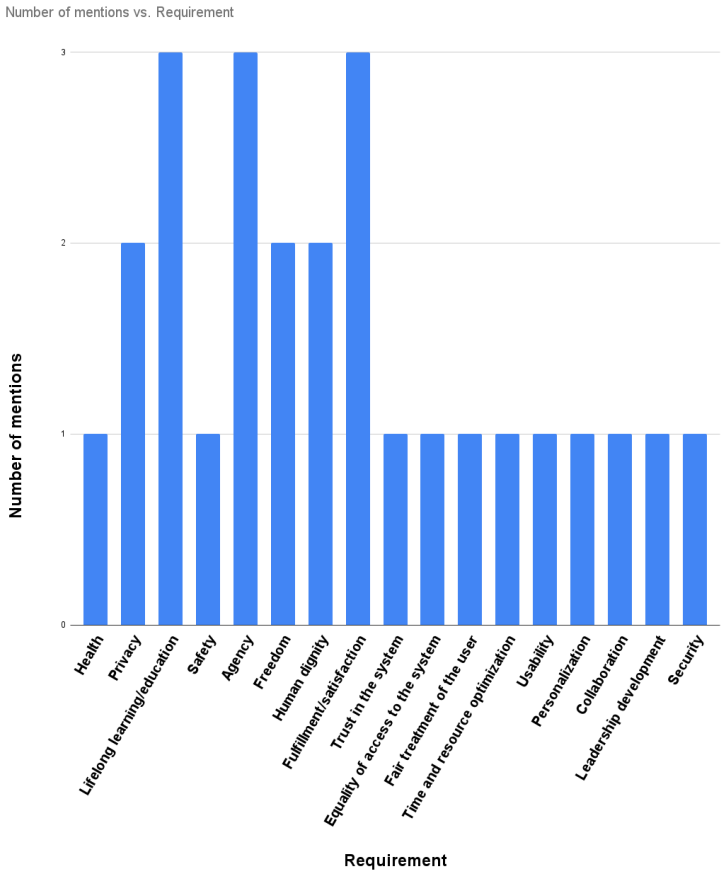


Figure 3.3: Individual dimension requirements present in the selected studies

As we can see these requirements base themselves on attempting to improve the users in some aspect, as well as enforcing human rights and giving the users a good experience. We can also observe that there is more ambiguity when it comes to defining the individual dimension as there are very few requirements that are mentioned more than once in the studies. The discrepancy between this dimension and the other two regarding the number of requirements and ambiguity has a justification, that being that some consider that the individual dimension and the social dimension share the same requirements and as such only the social dimension needs to be considered [21][27][32] leading to a decrease in considerations for this particular dimension.

RQ5 - What are the relations between sustainability requirements in the studied dimensions?

During our search we found many relations between the sustainability dimensions and requirements. Notably most of these were positive ones and a very limited number of negative relations. The results in RQ2, RQ3 and RQ4 all listed requirements that had positive relations towards the sustainability dimensions that they were associated to,

so we won't repeat them here. Negative relations on the other hand were limited very limited. They only got mentioned between requirements and never when affecting a particular sustainability dimension. The following list highlights the requirements found that negatively impact each other:

- Functional suitability and compatibility [24]
- Performance and security [24]
- Reliability and compatibility [24]
- Cost and performance [24]
- Performance and reliability [24]
- Security and energy efficiency [29]

We can observe that the majority of the studies we selected heavily focused on positive relations while often neglecting the negative ones. This presents another gap that could be addressed.

Main Research Question - What are the existing methodologies that support software sustainability and sustainable requirements that cover the economic, environmental and individual dimensions?

The methodologies found were listed in the section about RQ1, and the requirements and relationships in RQ2, RQ3, RQ4 and RQ5 so we'll not repeat them here but the conclusion that we can take from our search is that while there exist a variety of lists, tools and a few catalogs that help with implementing these dimensions of sustainability into software projects. There are not any that any lists of catalogs of reusable requirements for our dimensions. As proven by a different work focusing on the social and technical aspects of sustainability a catalog of this style helps a lot with the process of implementing sustainability during the requirements elicitation phase of the project [2]. In conclusion, while several methodologies exist that help with implementing sustainability, as shown in RQ1's answer, there is potential for a vast improvement in the requirements elicitation phase especially concerning the individual, economic and environmental dimensions due to a distinct lack of support for this particular niche.

3.1.4 Threats to validity

Threats to validity will always exist in any study regardless of how well they are mitigated. One of the threats to the study is the accuracy of the automated search that was used, while the string used was heavily optimized it can still be incomplete or provide inaccurate results. Some studies that could be relevant can also have been left out either due to the lack of keywords during the first iteration of the choosing process or the search string may

have been more restrictive than we initially thought or they could have been discarded due to not being in English or being unavailable in full text. The last factor that may be a thread was the fact that this was a single-person executed study with a high number of analyzed studies. This might have introduced some bias or errors in choosing the relevant studies during the choosing phase that may have otherwise have been corrected. Another measure that was taken was the use of the digital library Google Scholar. By using this library over others we reduced the chance of missing studies that would otherwise be in unchecked alternative libraries since it searches through all of them. The final measure that was used to combat this was the organization of the search's results in a spreadsheet with the inclusion and exclusion criteria marking each study, this allowed us to keep the results as they were eliminating the risk of searching through the digital library multiple times and receiving different results each time while also ensuring that they got excluded and included as we found them.

3.1.5 Results discussion

The systematic mapping study that was performed gave us some very necessary insights into the state of sustainability in the area of software development. One of these was the discrepancy between the sustainability dimensions that we studied regarding the quantity of information that exists. This can be seen by comparing the results of RQ2, RQ3 and RQ4 while there was a large number of results for what sustainability requirements exist for the economic and environmental dimensions with environmental having slightly more, the individual dimension had very few results in comparison. This bias was already justified in the answer to RQ4, but to reiterate this bias seems to stem from the idea that some people in the field have that the social and the individual dimensions are not distinct enough that they should be considered as different dimensions [21][27][32]. This has been something that other papers and tools consider incorrect. The Karlskrona Manifesto [58] and more recent tools such as SusAf [15] are some examples that agree that the individual dimension is necessary. We agree with this perspective, when looking up the definitions of these dimensions such as the ones presented in Chapter 2, there is always the emphasis that the social dimension is concerned with groups of people that function as a society whether they are big such as a country or small companies both of these could be considered societies and they both have different requirements when compared to a single person. The individual dimension places a focus on personal growth and satisfaction, as can be seen by their definitions they are not mutually inclusive, one individual may be happy with the way a system works but the society that the individual belongs to may not be sustained, the same way that just because society is working well it doesn't mean that its individuals are growing or happy with their conditions. It's because of this that we believe that it is important to keep this separation between these dimensions.

One other conclusion that we can take from looking at RQ2, RQ3 and RQ4 is that there is a significant crossover between the requirements of the sustainability dimensions.

Some papers have pointed this out in the past [21] showing that there are requirements that even affect more than one dimension at once. The results of the three questions ended up reinforcing this as we have requirements that exist in all dimensions these were: trust, satisfaction and usability. However, one detail that can be pointed out is that these are very generic, and because of that perhaps they need to be refined to fit the sustainability dimension that they are being considered when constructing lists catalogs or tools.

Regarding relationships between requirements, through RQ5 it was possible to see that there is a big focus on positive relations. Negative relations got very neglected, this may have been due to the studies selected, however, this neglect could also be indicative of a trend within the area of sustainability. It's essential to acknowledge both the positive and negative relations between requirements to ensure that sustainability is enforced as expected by the developers.

In terms of tools we found that many tools help to implement sustainability into software projects, most of these tools as seen in RQ1 were models, frameworks and checklists, most of which help introduce sustainability into projects at various stages of development. If we look at all our research questions' answers we can see that there is a lack of investment in the requirements engineering phase of software development. While it's difficult to get a completely accurate generic tool or catalog for this phase of a project especially considering that every project is contained in a different problem domain and the vastness of sustainability as a requirement, there don't seem to be many that have attempted to make such a tool or catalog that cover sustainability requirements.

3.2 Summary

In this chapter, we investigated the current state of the art regarding tools for implementing sustainability in software projects, the requirements associated with each of our studied dimensions, and how these dimensions relate to one another. By analyzing the selected studies, we gathered the information needed to answer our research question and sub-questions. We compiled a list of tools that support sustainability implementation and identified several sustainability requirements, and relationships among some of them. Our analysis revealed deficiencies in existing tools and catalogs concerning the individual sustainability dimension, often combined with the social dimension due to a recurring bias or a misunderstanding of the dimensions. Another issue we identified is the lack of attention to the negative interactions between sustainability requirements, which can lead to inaccuracies in assessing a project's true impact on sustainability. Lastly, we observed a significant gap in tools and catalogs specifically designed to facilitate the elicitation of sustainability requirements during the requirements gathering phase of software projects.

4

A Sustainability catalog

Catalogs, in the area of software development, are tools that consolidate a broad array of pre-made developmental decisions and reusable information. This information can be extremely useful for developers, providing a foundation that streamlines the development process. Our catalog, in particular, has been designed to deliver a set of sustainability requirements. The primary objective of our catalog is to offer a reusable set of requirements and to facilitate an in-depth exploration of how these requirements interact within and across different sustainability dimensions. This is crucial, as it allows users to understand the impact of their decisions on sustainability and view the broader context in which these requirements operate, and as a consequence make more informed decisions during the development process. The catalog was structured to accommodate the refinement of requirements. When a requirement can be broken down into more granular components, this relationship is contained within the catalog. This enables developers and future contributors to adjust the catalog to fit their project's specific needs or incorporate enhancements. This chapter will explore the concepts that form our catalog.

4.1 Development and structure of the catalog

As previously stated, the primary objective of this work is to develop a catalog that organizes sustainability requirements according to the dimensions they pertain to. To construct this catalog, we collected and analyzed non-functional requirements related to sustainability as addressed in Research Questions RQ2, RQ3, RQ4, and RQ5. This catalog provides a view of each dimension through an iStar model that adheres to the iStar 2.0 standards. The iStar model was constructed from the ground up. Initially, all requirements that we found were integrated into the model. These requirements were then mapped to their respective dimensions, and the model was refined to enhance its accuracy and utility. In the following sections, we will detail the structure and development process of the catalog.

4.1.1 Development of the catalog

The catalog contains the sustainability requirements collected in Chapter 3. As highlighted in the previous section, the aim was to consolidate all these requirements into a single model, showcasing their relationships with the various dimensions and exposing any refinements creating a catalog that is comprehensive and easy to use for software developers. We used iStar 2.0 standards, because they provided a structured framework for organizing the requirements. In this context, we used PiStar as the primary tool for building our catalog. PiStar is well-suited for this purpose because it inherently supports iStar 2.0 standards, ensuring that the models created comply with these guidelines. By basing our work on iStar 2.0, we ensured the catalog was comprehensive and consistent while adhering to established modeling practices. This method allowed us to develop a detailed and well-organized catalog, enhancing its usefulness and applicability for software developers focusing on sustainability-related requirements.

4.1.1.1 Mapping catalog concepts to iStar elements

Once we decided to use piStar, the first step was to map the concepts of our catalog to iStar 2.0. The iStar framework has several elements. Here's how we used each iStar element:

- **Actors:** Actors represent entities with specific goals and have two sub-types: *Role*, which denotes an actor within a particular domain, and *Agent*, referring to more concrete, physical actors such as individual humans or organizations [7]. In our catalog, there is only one regular actor the catalog itself—because our focus is on the requirements and relationships within the catalog rather than interactions with external entities. The goal is to provide a model that users can modify as needed and use as a starting point for implementing their system or a component of their system. Although the catalog itself does not benefit from having actors, iStar 2.0 requires that all qualities be associated with an actor [7], so we included the catalog actor as a foundational element.

- **Qualities:** In our catalog, iStar qualities are non-functional requirements derived from earlier stages of the work. These qualities include the initial requirements and their refined versions. They are interconnected through contribution links (such as break, hurt, help, and make) to illustrate their relationships.

- **Goals:** These represent objectives in a system that an actor wants to achieve [7]. While the catalog does have the goal of helping implement sustainability, sustainability itself is also a non-functional requirement. The decision was to represent it as a quality rather than a goal. During the usage of the catalog, the user should connect the qualities from the catalog with the defined goals of the system through relationship links or contribution links.

- **Tasks:** Are concrete actions required to achieve a goal or quality. They link to goals through refinement links, specifically *AND* or *OR*. The *AND* link indicates that all associated tasks must be complete, while the *OR* link signifies that only one of the tasks needs to be accomplished [7].

- **Resources:** Represent static entities needed to complete certain task elements, often including physical items or types of information. They must link to task elements they support via the *NeededBy* link [7].

- **Actor links:** Represent relationships between actors, specifically the *is-a* and *participates-in* relations [7]. Since our catalog contains only one actor, these links are not applicable.

- **Dependencies:** Represent the relationships between two actors [7]. Since our catalog includes only one actor, these dependencies do not exist in the catalog. However, in actual projects, dependencies are expected to exist. Since qualities cannot connect to elements outside the actor that contains them, users should replace qualities with quality dependency links.

- **Contribution links:** Illustrate the effects that elements have on qualities [7]. These links will be among the most frequently used in the catalog, as each quality will connect with others in some manner.

- **Refinement links:** Connect goals to the tasks required to achieve them and come in two types: *AND* and *OR* [7]. As mentioned earlier, *AND* links indicate that all related tasks must be completed to accomplish the goal. Meanwhile, *OR* links establish that at least one of the tasks must be implemented for the goal to be achieved.

- **Qualification links:** These links illustrate how a goal should be achieved [7]. The catalog itself does not use them; they are, however, employed by the users of our tool to connect their elements with those in the catalog.

4.1.2 General structure of the catalog

The catalog was built with a specific structure to make it easy to read, easy to expand and easy to use. The catalog is composed of the sustainability requirements that were found divided into categories with each level of the catalog representing something different, note that the categorizations mentioned on the following list will be explained more in depth in later sections:

With the mapping complete, the next step was to create the catalog within these constraints. The catalog was designed with a specific structure, ensuring it is easy to read, expand, and use. It is composed of the sustainability requirements identified and organized into distinct categories. Each level of the catalog represents a different aspect of these requirements, providing a clear and hierarchical view of the information.

The catalog was structured to facilitate navigation and usability. Each category and subcategory also contributes to a structure that allows straightforward expansion and updates as new requirements or insights emerge.

The categorizations used in the catalog are detailed in the following list. However, a more detailed explanation of these categorizations is provided in the subsequent sections.

- **The top item** represents our main objective: to make the system sustainable. So, this level is composed of a single quality: Sustainability. Which then branches into lower levels based on our categorizations.

- **The second level** represents one of the categorizations that was applied during development. This categorization separates requirements into ones that affect a single dimension and the ones that involve more than one dimension simultaneously.

- **The third level** contains the sustainability dimensions. At this level, each of the individual dimensions is represented by its quality element. We also have as a consequence of our categorization quality elements, which represent combinations of sustainability dimensions.

- **The fourth level** contains a categorization of the sustainability requirements created by Ian Sommerville. Each of our sustainability requirements belongs to one of the categories listed in his book [59].

- **The fifth level** contains our sustainability requirements, organized according to two distinct categorizations. First, it includes a categorization specifically created and proposed by this work, tailored to address the unique aspects of sustainability in the context of our project. Second, it aligns the requirements with the ISO/IEC 25010 standards [60], a widely recognized framework for software quality.

- **The sixth level and below** represent additional layers of refinement within the catalog. Any levels beyond the fifth were considered refinements of the preceding levels. In the catalog we developed, there are six levels in total. These are sufficient to capture the categorizations and sustainability requirements identified in this work.

Once the categorizations were complete, contribution links connecting the sustainability requirements were added to indicate how they influence each other.

4.1.3 Categorization of the requirements

As previously stated, the catalog incorporates several layers of categorization, each used to enhance its structure and functionality. In this section, we will explain how each categorization was used to organize our requirements and explain the significance of these categorizations.

4.1.3.1 Single and Multi dimension categorization

During the development of our catalog, we encountered a significant challenge: the catalog became difficult to read and manage due to the high number of links between requirements and sustainability dimensions. Many requirements were connected to multiple sustainability dimensions, creating a cluttered and unwieldy catalog. To address this issue and enhance the catalog's readability, we divided it into two distinct sections: one for requirements that affect a single dimension and another for those that impact multiple dimensions. This division meant that the section for single-dimension requirements was straightforward. Each of the requirements is assigned to only one of the dimensions: "Individual Dimension," "Economic Dimension," and "Environmental Dimension". For the multi-dimensional section, we adopted a more complex categorization system. Given that requirements could influence multiple dimensions, we considered all possible combinations. For instance, requirements affecting the economic and environmental dimensions are considered "Environmental and Economic Dimensions" requirements. Likewise, if a requirement impacted all three dimensions—economic, environmental, and individual it was labeled an "Environmental, Economic, and Individual Dimensions" requirement. This approach to categorization ensures that each requirement is accurately represented according to its scope of impact while facilitating navigation and improving the readability of the catalog.

4.1.3.2 Ian Sommerville categorization

After improving readability, we applied another categorization: Ian Sommerville's categories as detailed in his book [59]. Sommerville's categorization provides another approach to classifying requirements. This categorization is organized into several levels, each increasing in specificity. At the top of Sommerville's classification there are the most general categories. These include "External Requirements", "Organizational Requirements", and "Product Requirements". Each of these broad categories encompasses a wide range of requirements that influence various aspects of the system from different perspectives. Moving down the hierarchy, we encounter more specific categories. These include specialized requirements such as "Space Requirements" and "Safety Requirements." These lower-level categories address particular aspects of the system's performance or constraints, providing detailed insights into specific needs. By incorporating Sommerville's categorization, we added another layer of organization to the catalog. This allowed us to refine and categorize the sustainability requirements, ensuring that they align with established standards in software engineering. For this work the decision was made to use exclusively the most specific categorizations which are: usability requirements, performance requirements, space requirements, reliability requirements, portability requirements, delivery requirements, implementation requirements, standards requirements, ethical requirements, interoperability requirements, privacy requirements, and safety requirements.

4.1.3.3 ISO/IEC 25010 categorization

Another categorization applied to our catalog was the ISO/IEC 25010 standard [60], a framework for classifying software quality requirements. We identified several sustainability requirements that aligned closely with the categories defined by the ISO/IEC 25010 standard. Given this alignment, we integrated the categorization into our catalog. Integrating ISO/IEC 25010 into our work ensured the catalog's structure remained true to pre-established software engineering standards. Applying the ISO/IEC 25010 categorization brought one other advantage: it provides a consistent and structured approach to organizing requirements, making it easier for users to identify refinements. The following is a list of changes that this categorization made to our requirements:

- "Capacity Optimization" was renamed to "Capacity".
- "Resource Optimization" was renamed to "Resource Utilization".
- "Functional Appropriateness," "Functional Correctness," and "Functional Completeness" were consolidated under "Functional Suitability".
- "Co-existence" and "Interoperability" became refinements of "Compatibility".
- "Recoverability" and "Availability" became a part of "Reliability."
- "Integrity" was placed as a part "Security."
- "Usefulness" and "Trust" were combined under "Satisfaction."
- "Health" and "Safety Qualities" were merged and renamed as "Health and Safety Risk Mitigation."
- "Flexibility" and "Completeness" were grouped under a new quality attribute called "Context Coverage".
- "Modifiability" and "Reusability" were placed under "Maintainability".
- "Resource Utilization," "Time Behavior," and "Capacity" became a part of "Performance Efficiency".

4.1.3.4 Other categorizations

In addition to the standard categorizations applied, we introduced several custom categorizations to enhance the catalog's organization and readability. These changes were designed specifically to make the catalog less cluttered or to remove redundancy when found. Our adjustments created new requirements, reorganizing some of the existing ones as part of the new categorization and also removed a requirement that was considered redundant. The complete changes were:

- “Human Dignity” and “Equality of Access” were incorporated into “Fair Treatment of the User”.
- “Ecosystem Protection”, “Promotion of Biodiversity and Land Use”, “Greenability”, “Reduction of Waste and Disposal”, “Energy Efficiency”, “Environmental Footprint Reduction”, and “Pollution Reduction” were consolidated under “Environmental Effects”.
- “Business Agility”, “Business Growth”, and “Creation of Business Opportunities” were grouped under “Business Effects”.
- “Environmental Risk Mitigation”, “Economic Risk Mitigation”, and “Health and Safety Risk Mitigation” were turned into part of “Freedom from Risk”.
- The “Quality Requirement” was removed due to its ambiguity.

4.1.4 Sustainability dimensions, requirements and catalogs

While the primary objective of this work was to develop a comprehensive catalog of requirements encompassing the sustainability dimensions studied, the catalog we created can also be divided into individual sub-catalogs, each dedicated to a specific sustainability dimension viewable in Figures 4.1, 4.2 and 4.5. The sub catalogs created have fewer levels than the main catalog but have a similar structure:

- **The top item** represents the dimension that the catalog belongs to.
- **The second level** contains the categorization of Ian Sommerville used the same way as the main catalog.
- **The third level** contains the sustainability requirements using the categorization proposed during this work and also the ISO/IEC 25010 standards like the main catalog.
- **The fourth level and below** represent more layers of refiniment. Any requirements at this level are considered refinements of the requirements from the levels above. The sub catalogs only have four levels since these were enough to capture all the requirements that were identified.

The next sections will examine these individual dimensions and their corresponding sub-catalogs.

4.1.4.1 Individual Dimension

As previously mentioned, the "Individual Dimension" is the one for which we identified the fewest requirements. This dimension primarily focuses on creating systems that support and enable the development and thriving of individuals [13]. It addresses aspects

related to personal growth and the ability of individuals to thrive within the system. For this dimension, we ended up with the catalog present in Figure 4.1.

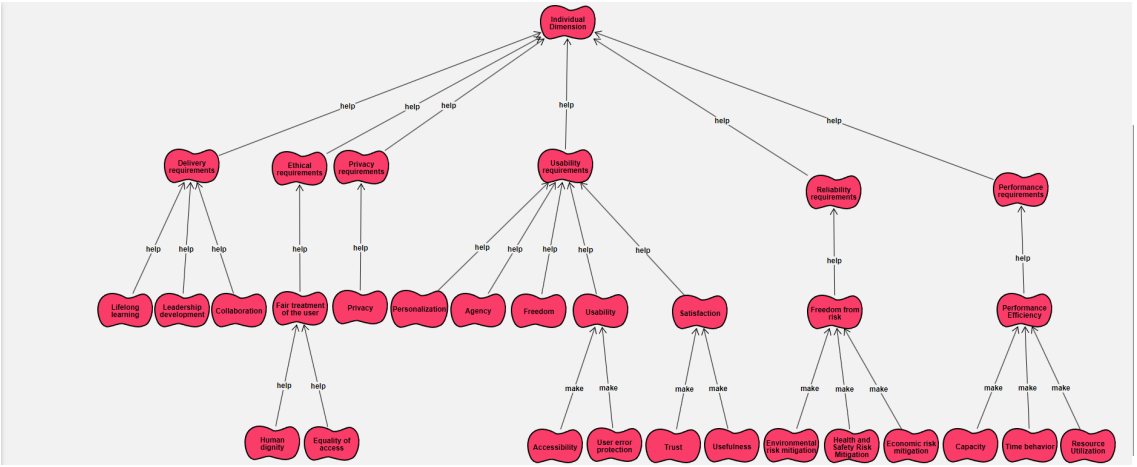


Figure 4.1: Individual dimension sub-catalog

4.1.4.2 Economic Dimension

The Economic Dimension, which focuses on financial and business aspects of a project [13], had the second highest number of requirements in our catalog. Although it's not the most expansive dimension, it still resulted in an extensive catalog. Figures 4.2, 4.3 and 4.4 show the catalog from this dimension.

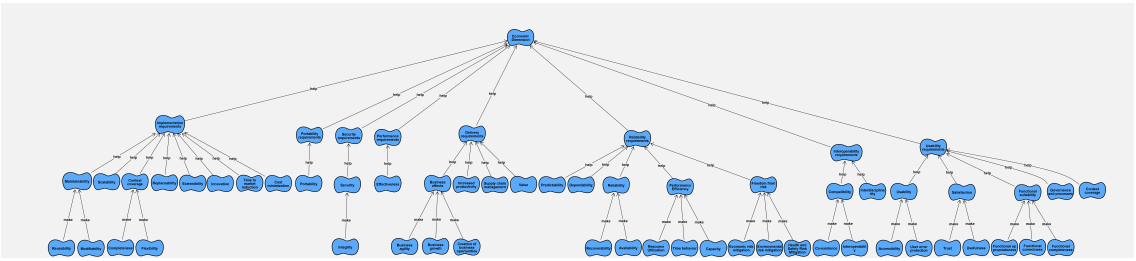


Figure 4.2: Economic dimension sub-catalog

4.1.4.3 Environmental Dimension

The Environmental dimension, associated with the management of energy consumption and waste production, preserving the balance of the ecosystems and preventing climate change [13] had the highest number of requirements which was expected considering how vast the factors that affect the environment are. Figure 4.5, Figure 4.6 and Figure 4.7 show the catalog of the environmental dimension.

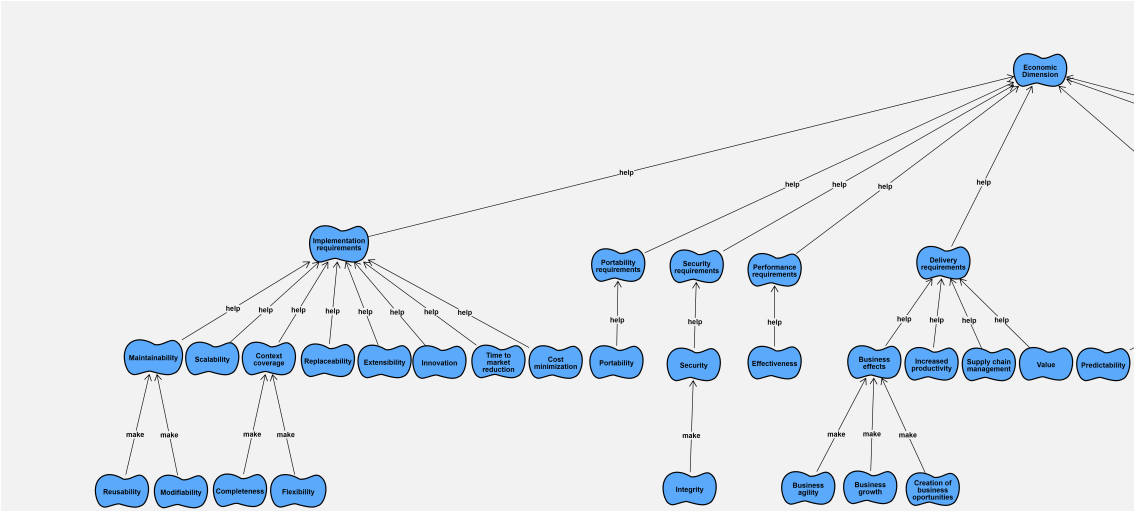


Figure 4.3: First half of the economic dimension sub-catalog zoomed in

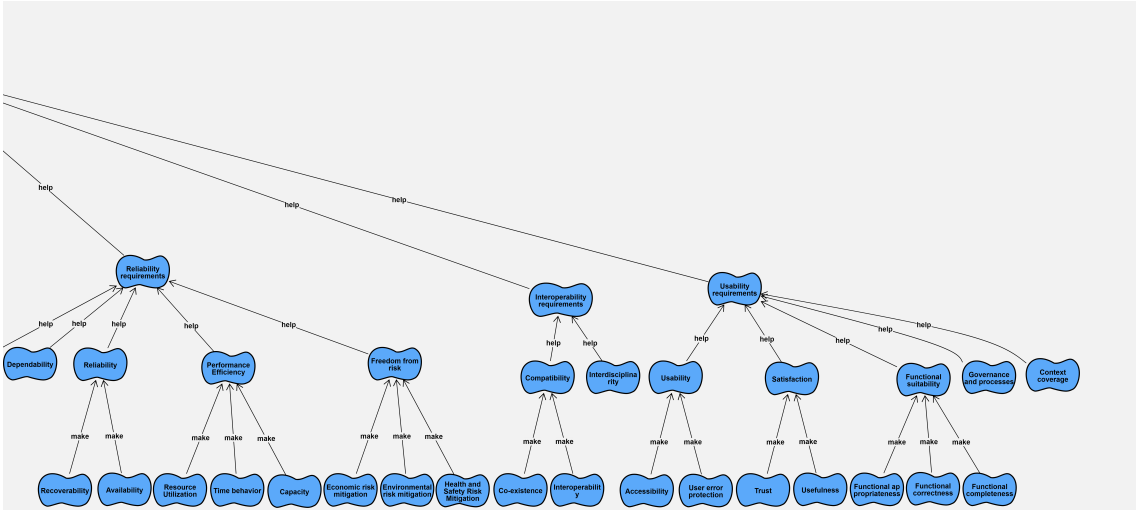


Figure 4.4: Second half of the economic dimension sub-catalog zoomed in

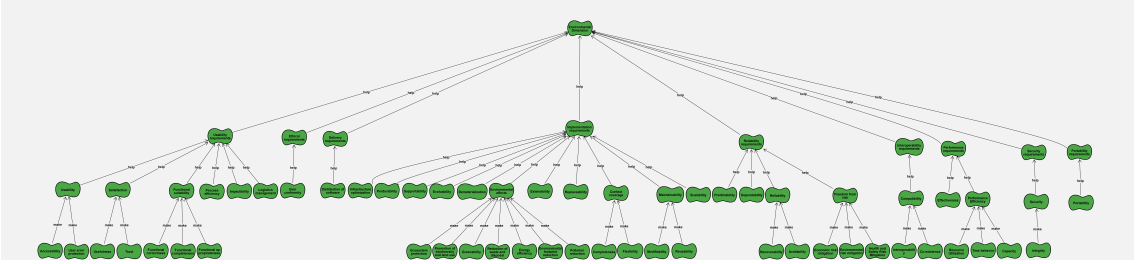


Figure 4.5: Environmental dimension sub-catalog

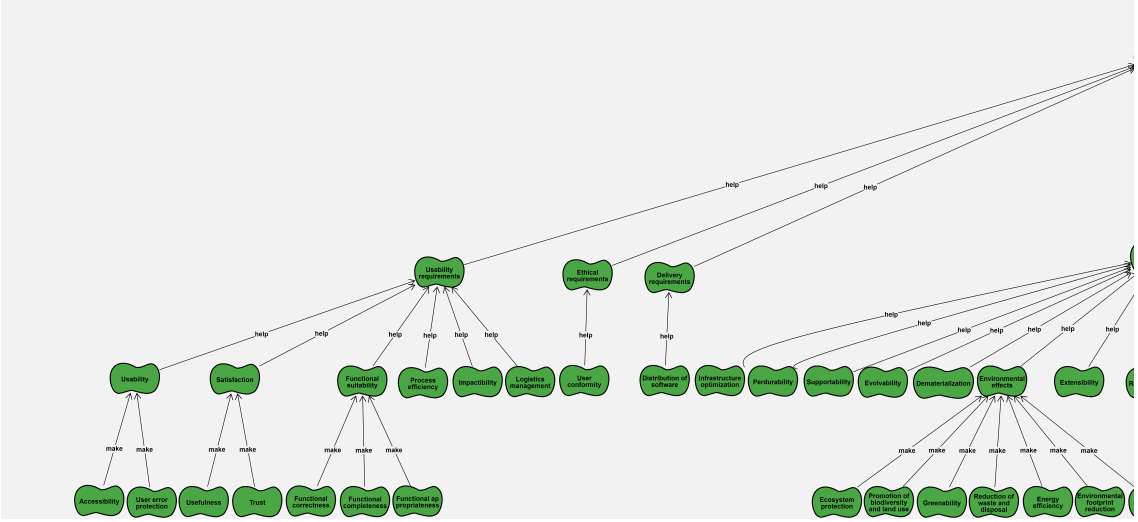


Figure 4.6: First half of the environmental dimension sub-catalog zoomed in

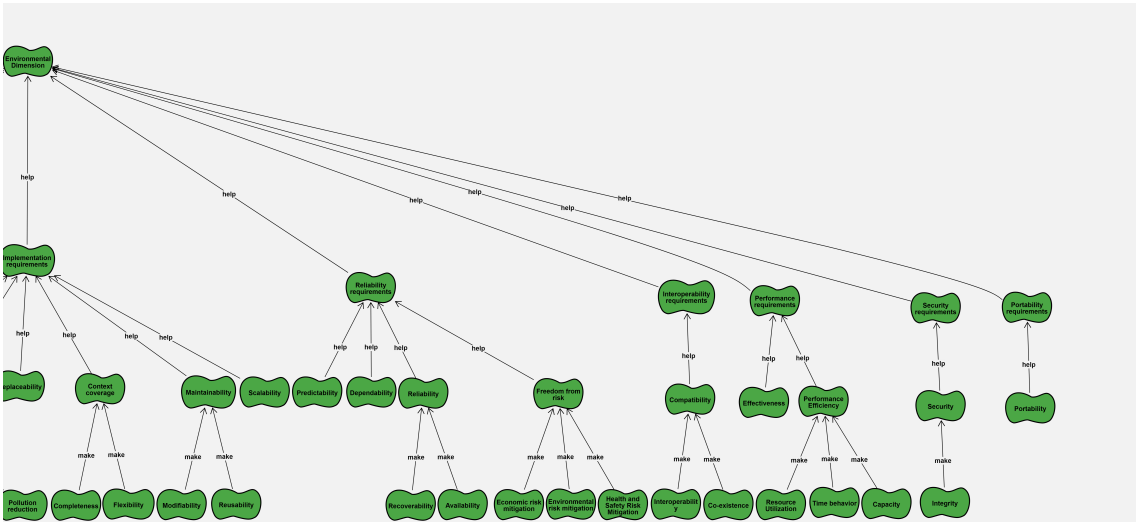


Figure 4.7: Second half of the environmental dimension sub-catalog zoomed in

4.2 The complete catalog

After applying our categorizations and joining all of the catalogs for the individual dimensions the resulting complete catalog can be seen in Appendix B.

4.3 Analysis of the catalogs

While our catalogs are usable independently, they reveal a notable degree of overlap in their requirements. Every requirement belonging to more than one dimension always belongs to the Economic and Environmental dimensions. The only dimension that may not appear in these cases is the Individual dimension, indicating a significant interrelationship between the Economic and Environmental dimensions. On the other hand, the Individual

Dimension overlaps with both the Economic and Environmental dimensions always at the same time, but with fewer requirements. Although our goal is to help achieve a fully sustainable project, it's nearly impossible to ensure that all sustainability requirements are fulfilled in practice. Project decisions often lead to positive impacts on some dimensions while negatively affecting others. Therefore, the best approach when using these catalogs is to maximize sustainability within practical limits instead of trying to have a positive effect on every dimension's requirements. We should also consider how the dimensions overlap and influence each other. These relations between different dimensions can complicate the implementation of sustainability and could require balancing conflicting objectives. The following sections are dedicated to a reflection on the multi-dimensional aspects of the catalog.

4.3.1 The individual dimension

The Individual Dimension, as evidenced by the complete catalog, shows minimal overlap with the other dimensions. It specifically lacks requirements exclusively paired with either the Economic Dimension or the Environmental Dimension. Instead, it only shares the most general requirements with both dimensions simultaneously. This this was anticipated taking into consideration the definitions provided in Chapter 2. There is little correlation between the Individual Dimension and the other two dimensions. While personal improvement, user enjoyment, and other positive effects associated with the Individual Dimension can increase user engagement and potentially enhance economic value, this does not directly translate to impacts on the Economic or Environmental Dimensions. For instance, an application designed to foster personal growth or give enjoyment might increase user engagement and add economic value. However, this increase in interaction does not necessarily contribute directly to effects in the economic or environmental dimensions.

4.3.2 The Economic and Environmental dimensions

In contrast to the Individual Dimension, the Economic and Environmental Dimensions exhibit substantial overlap. Beyond the requirements that impact all three dimensions, many requirements are specifically shared between these two dimensions. The presence of requirements common between the Economic and Environmental Dimensions means that addressing these requirements can simultaneously benefit both dimensions. For example, a system requirement that enhances a quality relevant to both dimensions results in double the progress toward sustainability. Conversely, if a requirement negatively impacts a quality shared by both dimensions, it could hinder progress in both areas, potentially regressing our sustainability goals significantly. This observation highlights an aspect emerging from applying our categorization to sustainability requirements. It highlights that by focusing on the multi-dimensional aspects of the catalog, we can more effectively optimize sustainability. Addressing these shared requirements first

can lead to more improvements. Once the multi-dimensional requirements are tackled, we can shift our attention to the requirements affecting a single dimension, ensuring a more balanced and thorough approach to achieving sustainability goals. In conclusion, many of the requirements that were found affect these two dimensions simultaneously showing a correlation between the two and displaying the wide impacts that the catalog's requirements can have on sustainability.

4.4 Summary

In this chapter we went over how we developed our sustainability catalog that addresses three of the five core dimensions of sustainability. We specified the general structure of the catalogs as well as how they were categorized in an effort to maintain a higher degree of readability and usability. We developed a total of three individual catalogs, one for each of our studied dimensions, these catalogs were then joined and transformed into the complete catalog that was then used as a basis for the plugin that was later developed. Through the development of the catalog we noticed that there was a large overlap between the economic and environmental dimensions, while the individual dimension was less connected with the other two dimensions.

5

Specification and supporting tool

As part of this project, we developed a supporting tool to enhance the usability of our sustainability catalog. The tool supports the specification of sustainability requirements and is ablew to represent the relationships between them. After evaluating various options, it was decided to create a plugin for the *piStar* tool[1]. This tool enabled us to accurately model the connections among sustainability requirements and dimensions facilitating the creation of the catalog and ensuring that our supporting tool also adhered to the iStar 2.0 standards. Using *piStar*, we could also create a visual and interactive representation of the catalog, making it easier to understand and apply the relationships between various requirements.

5.1 The piStar tool

piStar is an open-source tool developed using HTML, JavaScript, and CSS, designed for creating goal models that adhere to iStar 2.0 standards. One of the most notable features of *piStar* is its flexibility, which allows easy extension and customization through plugins. Another significant advantage of *piStar* is its ability to operate directly in web browsers, ensuring compatibility across a wide range of devices [61]. This accessibility makes it a versatile tool for developers working in different environments. By enforcing iStar 2.0 standards, *piStar* guarantees that the models created are both accurate and reliable, ensuring their practical utility in real-world applications. The interface of *piStar* allows users to place and manipulate iStar elements. These elements were used in our project according to the mapping detailed in Chapter 4, facilitating the creation of structured, standard-compliant models that effectively represent complex relationships and sustainability goals.

[1] PiStar Tool Website

5.2 Plugin functionality and implementation

Because *piStar* was programmed using HTML, JavaScript, and CSS, the tool is easy to expand and modify. The plugin developed was written in JavaScript and integrated directly into the tool's HTML code. This approach allowed us to introduce several new functionalities previously unavailable in *piStar*. By making the installation method as simple as appending an import of a JavaScript file to the end of the tool's HTML, we ensured that the plugin is not only straightforward to install but also self-contained, with all the necessary features packaged within a single entity. This choice enhanced the plugin's ease of installation and use, making it more accessible and as easily modifiable as the *piStar* tool itself. In the following subsections, we will discuss the functionalities implemented and explore how the catalog from Chapter 4 was integrated into the plugin, explaining how these contribute to the overall utility of *piStar* in supporting sustainability in software projects.

5.2.1 Catalog within the plugin

The *piStar* tool is designed to support the saving and manipulation of JSON objects. These JSON objects, when properly structured, can be interpreted by the tool and displayed as iStar 2.0 models. This functionality is native to *piStar* and it's utilized in several pre-existing example models, which are embedded as JSON elements within the tool's code. Recognizing the potential of this feature, we integrated our sustainability catalog as a JSON object within the custom plugin we developed. By incorporating the catalog as a JSON object, we ensured that all its sustainability requirements and relationships could be readily accessed and manipulated directly within *piStar*.
By aligning the catalog with *piStar's* native model-saving functionality, where models are saved as text files containing a JSON object, we significantly simplified the process of updating and modifying the catalog. This choice makes it easy for future developers to update the catalog as needed. Should further research or changes be required, anyone can quickly replace the existing catalog by copying the content from a saved *piStar* model file into the plugin's code. This method of updating ensures that the catalog can evolve alongside the ever-changing sustainability landscape without requiring redevelopment of the entire plugin. By using JSON objects, the catalog becomes more interoperable with other tools and platforms. JSON is a widely used data format, and its use opens up possibilities for integration with other systems that also support JSON, thereby extending the utility and reach of the *piStar* tool beyond its current capabilities. This interoperability can lead to broader applications, where the catalog might be used with other software development tools, enhancing its relevance and impact in developing sustainable software projects.

Embedding the catalog as a JSON object within the *piStar* tool capitalized on its existing strengths and ensured that the catalog could be easily accessed, updated, and expanded.

This approach enhances the usability and adaptability of the plugin, making it a powerful resource for developers working on sustainability-oriented projects. The decision to use JSON also opens doors for future integration with other platforms, further extending the potential applications of the catalog and the *piStar* tool.

5.2.2 Implemented functionalities

Most of the plugin's functionalities can be accessed by clicking the button within the "plugins" tab of the toolbar called "sustainability":

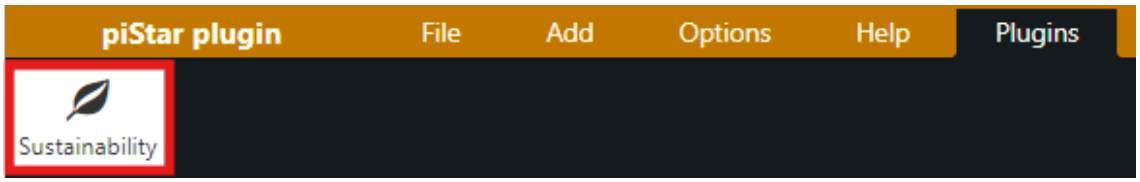


Figure 5.1: Plugin button in the toolbar

Given that the *piStar* tool allows catalogs to be saved as JSON files, we aimed to take full advantage of this feature to create a modified version of the catalog that is more tailored to user needs. The primary objective was to enable users to selectively display only the elements of the catalog that are relevant to their specific context, making the tool more efficient and the catalog easier to use. To achieve this, we implemented a range of functionalities designed to enhance the user experience. These enhancements include options for filtering and customizing the catalog's display, ensuring that users can focus on the most pertinent requirements and relationships without being overwhelmed by unnecessary information. By allowing users to interact with a more focused version of the catalog, we aimed to make the tool more accessible and easier to navigate, especially for those working on complex projects. Most of these functionalities are accessible through the plugin's menu, providing an interface that simplifies the management and customizing of the catalog, integrating seamlessly with the existing *piStar* environment. Doing so ensured that users could focus on their objectives while leveraging the full capabilities of the *piStar* tool. Next, we will explore these functionalities:

5.2.2.1 Catalog generation

The primary function of the plugin is to generate catalogs that align with the project specifications required by developers. This allows users to tailor the catalog to include only the sustainability requirements and dimensions relevant to their context, ensuring they have appropriate guidelines to work with. By offering this level of customization, the plugin enables users to streamline their development process, avoiding the potential overwhelm of searching through unnecessary or unrelated requirements. Additionally, users can continually refine and adjust their catalogs as they see fit since the catalogs

generated are the same as any other model created in piStar. When the user first accesses the plugin they will be faced with the window present in Figure 5.2.

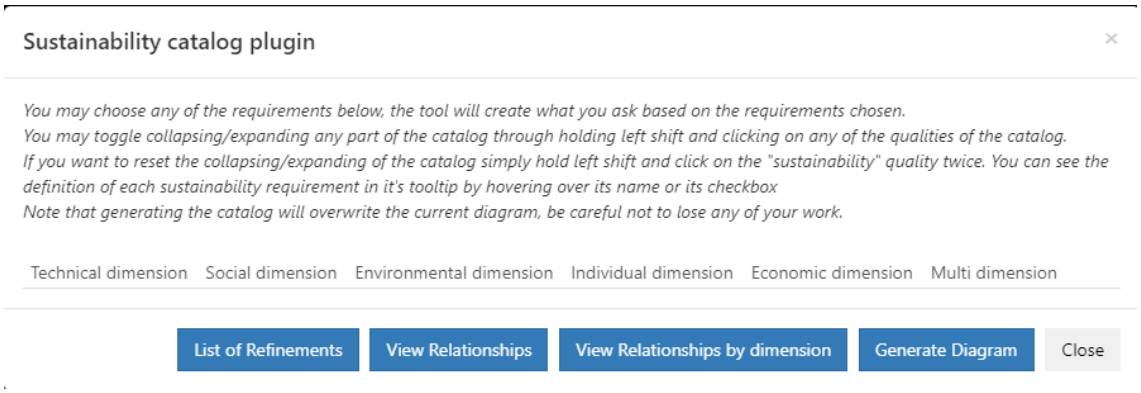


Figure 5.2: Plugin's menu

From this interface users can select any dimension's tab to explore the requirements that impact that particular dimension. The requirements displayed in the window are dynamically populated based on the content of the plugin's saved catalog. Users can then choose any number of these requirements and, by pressing the "Generate Diagram" button, create a customized model that includes only the selected requirements. For instance, if a user is working on a project focused on the environmental dimension, they can easily filter out all other requirements and generate a diagram that solely reflects the environmental criteria they need to address. As an example the selections present in Figure 5.3, Figure 5.4, and Figure 5.5 will create the model present in Figure 5.6.

Alternatively, users can press the "Generate Diagram" button without making any selections. This action loads the complete catalog, allowing users to modify the requirements themselves. It's important to note that many functionalities associated with additional buttons in the plugin depend on having a generated catalog or a loaded model that shares the same structure as the catalog. These features are designed to work with a specific model structure. Because of this, they are disabled until a catalog or model that matches the expected structure is generated or loaded. This choice was made to prevent errors and ensure the tool provides reliable outcomes, safeguarding against inconsistencies or unexpected situations.

5.2.2.2 Lists of relationships

During development, it became evident that as catalogs grow, they can become increasingly complex and difficult to understand. To address this issue, a functionality was developed that allows users to list and review all the relationships within their model. The first list implemented was a comprehensive overview of all the existing relationships, categorized by type. As previously mentioned, contribution links in the catalog can be of four categories: help, hurt, make, and break. By pressing the "View Relationship" button, users

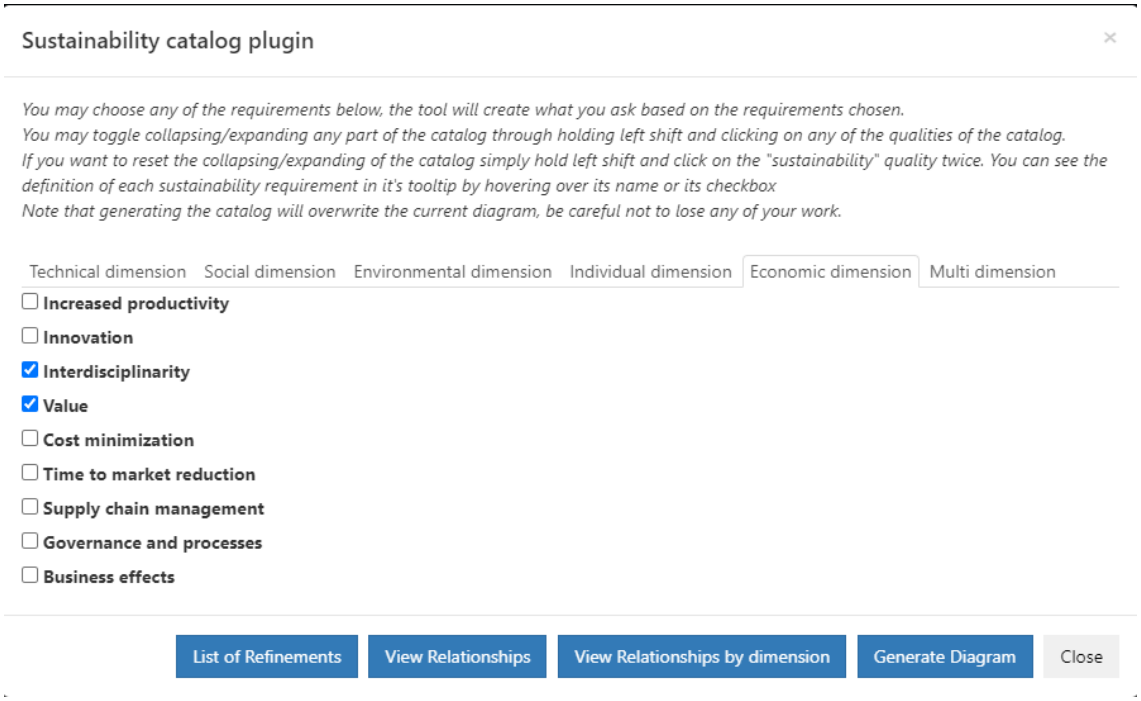


Figure 5.3: Economic dimension selection

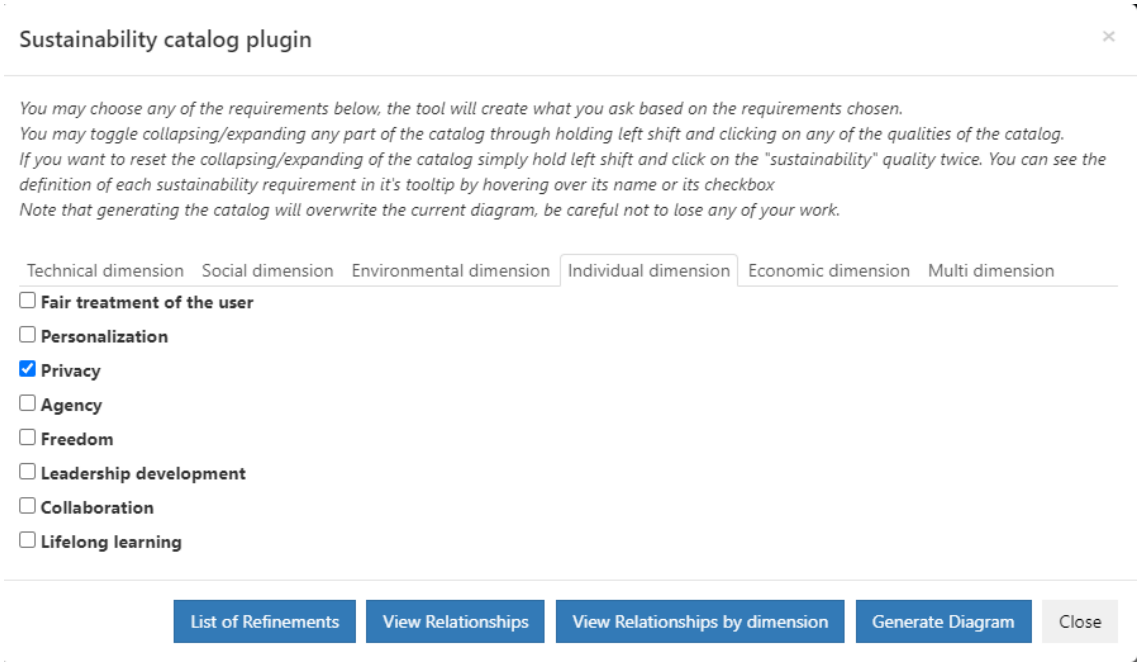


Figure 5.4: Individual dimension selection

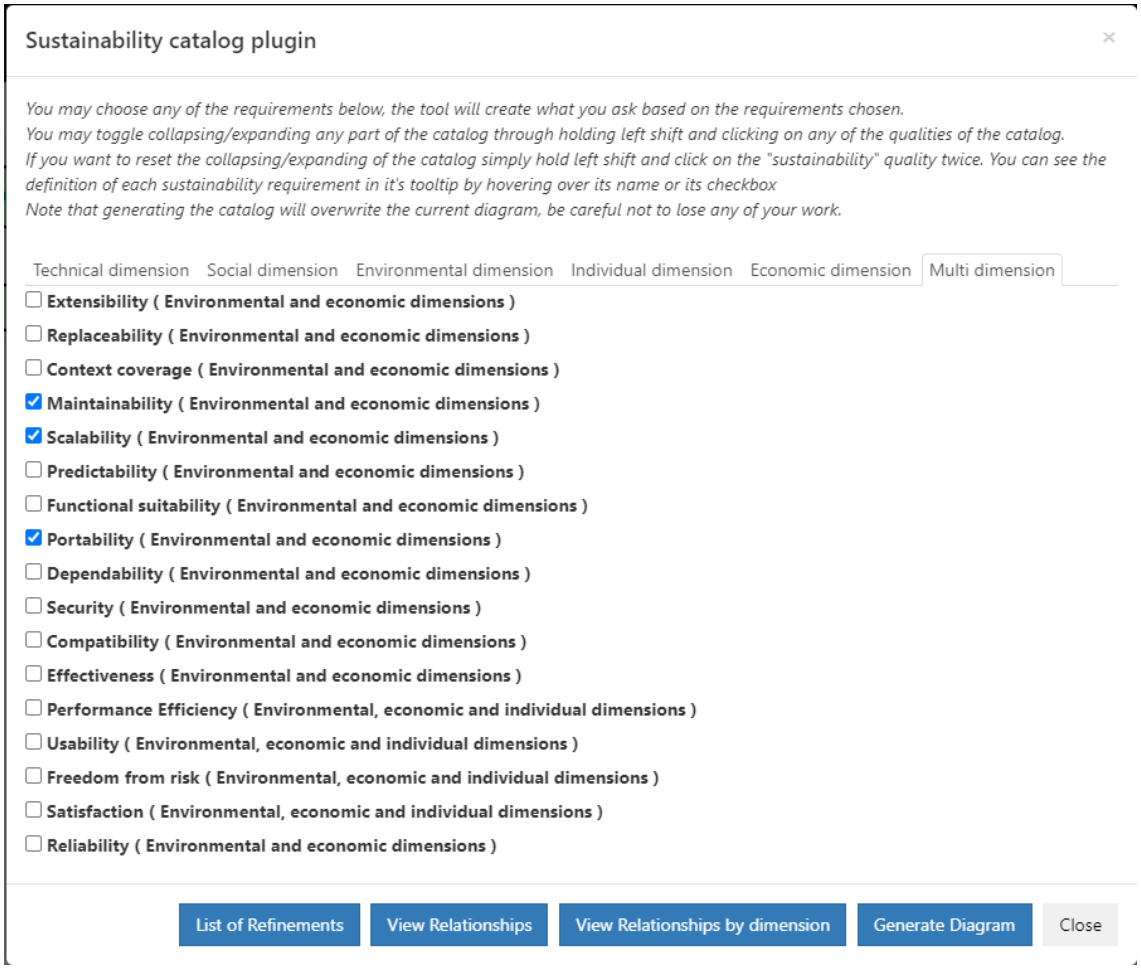


Figure 5.5: Multi-dimension selection

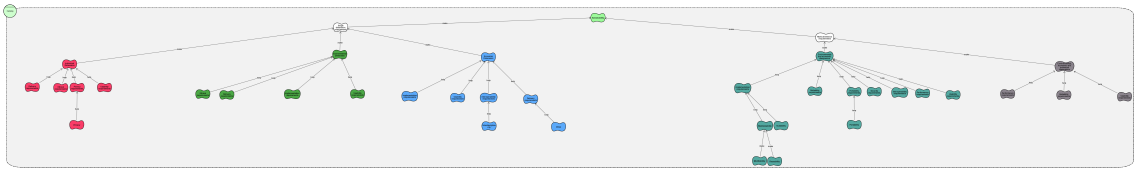


Figure 5.6: Resulting model

can open a new window that displays a detailed list of these relationships. The list begins with a small title indicating the type of relationship, such as "Help" or "Hurt." Below each title, the specific relationships are presented, with the source requirement indicated and an arrow pointing to its target requirement. To distinguish the type of relationships, a title appears to indicate that the next requirements have a relationship of a different type. An example of this list can be seen in Figure 5.7.

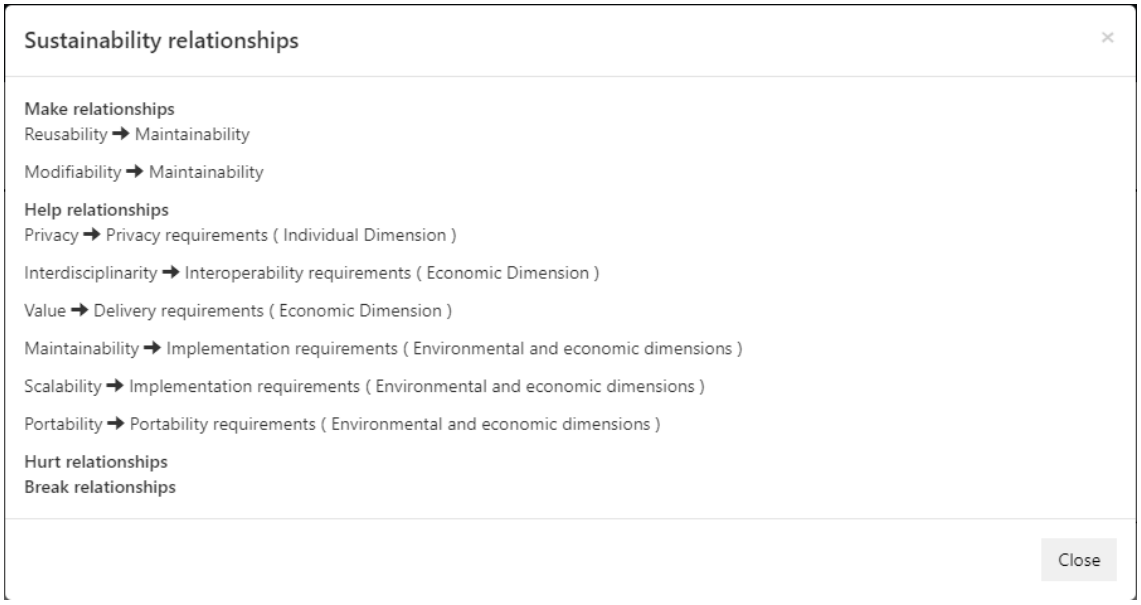


Figure 5.7: Example of a list of relationships by type

An alternative list can be accessed by pressing the "View Relationships by Dimension" button. This list provides a detailed view of how each requirement within the current catalog affects the different sustainability dimensions, including the specific type of relationship involved. This feature gives users a clearer understanding of the impact each requirement has on the various dimensions, allowing them to make more informed decisions in their project planning. Similar to the previous relationship list, this one is organized with a title identifying the dimension affected, followed by a series of requirements. Each requirement indicates the nature of the effect on that particular dimension through an arrow. In this work, since the catalog exclusively contains "help" relationships, the list only displays those while using the developed catalog, but in future catalogs with different types of relationships all types will be listed. Figure 5.8 shows an example of this type of list.

5.2.2.3 List of refinements

Another functionality added to enhance the user's ability to view the information present in their model is a list of refinements in the catalog. If a sustainability requirement can be broken down into several other, more specific requirements it is represented in the model by a "make" or "help" relationship between connected elements. In the list, each

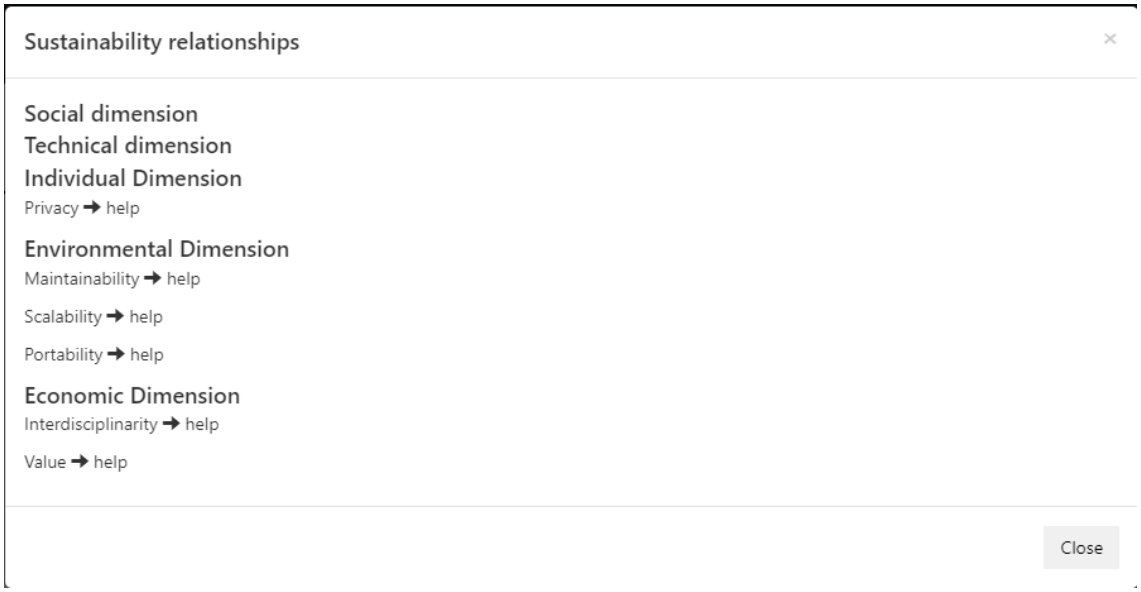


Figure 5.8: Example of a list of relationships by dimension

requirement that can be refined appears with an arrow indicating that it can be refined into the subsequent requirements. Figure 5.9 has an example of what this list can show.

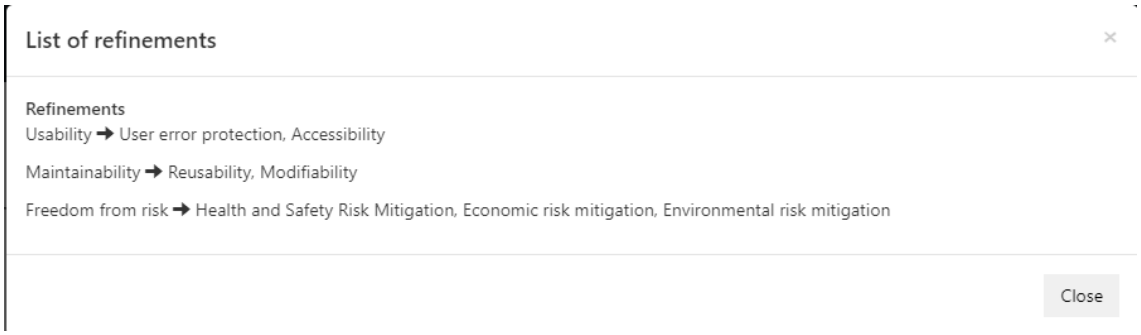


Figure 5.9: Example of a list of refinements

5.2.2.4 Definition support

Although not the primary focus of the tool, incorporating definitions for the requirements was deemed a valuable enhancement. Given that a requirement can have various interpretations depending on the user's perspective, providing clear definitions helps ensure a consistent understanding of what each requirement entails. To prevent some ambiguity that may surface during the use of the plugin, a dictionary feature was integrated into it. This dictionary is designed to be easily replaceable or modifiable through the script's code, allowing for future updates or extensions as necessary. The feature is straightforward: as the plugin assembles the list of sustainability requirements from the catalog, it references the dictionary to find the definition. If a requirement's name is found in the dictionary,

the corresponding definition appears as a tooltip when hovering over the requirement's name in the plugin's menu. If no definition is available, the tooltip will display a message: "Term definition does not exist." This approach ensures that users have immediate access to relevant definitions, enhancing their ability to make informed decisions based on clear and consistent information. Figure 5.10 shows the tooltip in action.

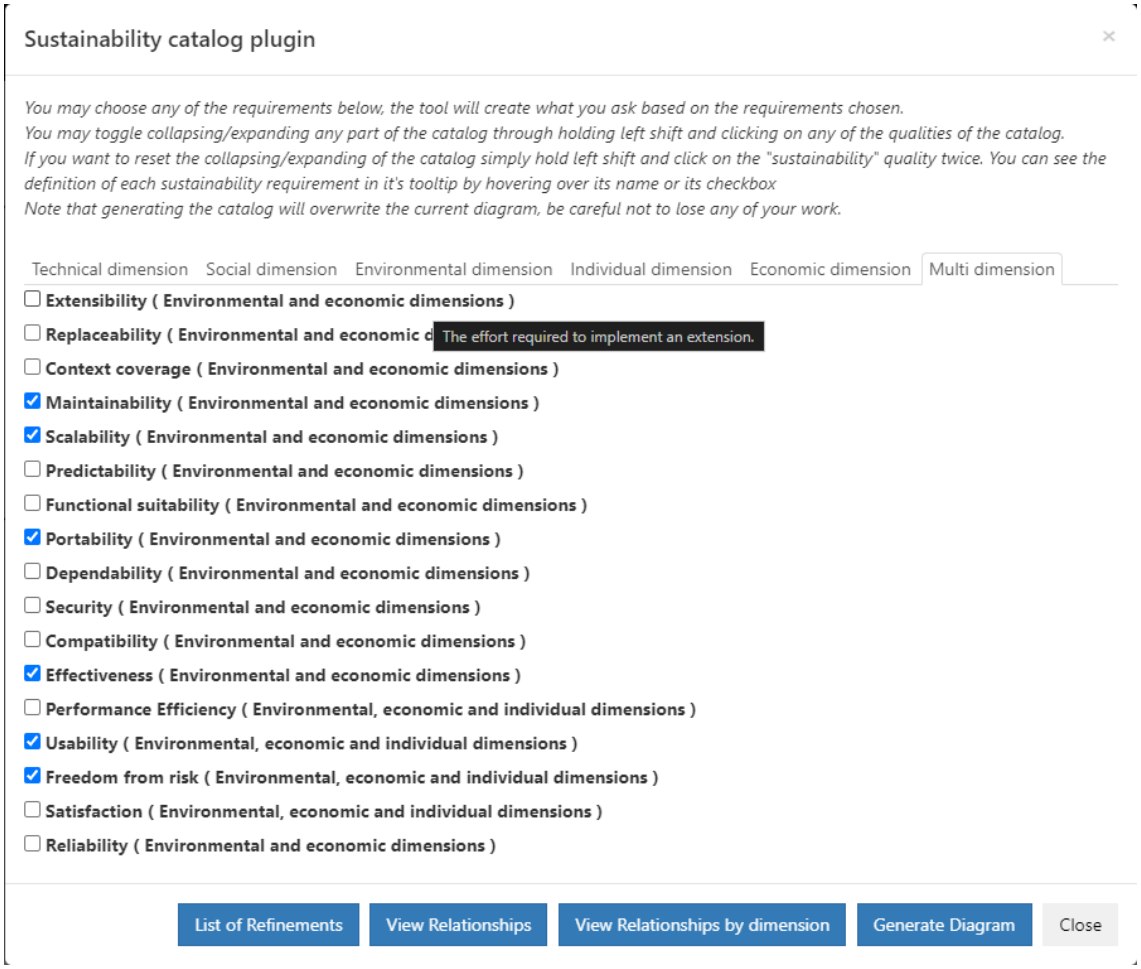


Figure 5.10: Example of a tooltip in the plugin

Table 5.1 shows the definitions that are present in the current version of the plugin. These definitions originated from the works that were used as a basis for our investigation and are listed in Chapter 3 and from the ISO/IEC 25010 standard.

Table 5.1: Requirement definitions

Term	**Definition**	**References**
Effectiveness	Complies data quality requirements both in input and output	[27]
Efficiency	Resources expended in relation to the accuracy, completeness and also less cost/time/human resources to conduct the research	[27]
Trust	Stakeholders have confidence that a product or system will behave as intended	[27]
Flexibility	The system can be used in contexts beyond those initially specified in the requirements, such as controlling different assets	[27]
Time Behaviour	Response, processing times and throughput rates of a system, when performing its functions, in real-time	[27]
User Error Protection	System protects users against making errors by being as intuitive as possible	[27]
Confidentiality	The system ensures that data is accessible only to those authorized to have access. Additionally, data should not be used for negative reporting, but only for improving efficiency	[27]
Usability	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	[60]
Risk mitigation	System mitigates the potential risk to financial status in the intended contexts of use System mitigates the potential risk to property or the environment in the intended contexts of use System mitigates the potential risk to people in the intended contexts of use	[27]
Recoverability	In the event of an interruption or a failure, the data can be recovered and the desired state of the system is re-established	[60]
Functional appropriateness/ suitability	The functions facilitate the accomplishment of specified tasks and objectives. System provides the correct results with the needed degree of precision.	[27]
Functional correctness	System provides the correct results with the needed degree of precision	[27]
Business improvement/growth	Improvements to the business with a focus on increasing the profitability and efficiency of the company, thereby achieving a greater economic sustainability of the organisation	[26]
Maintainability	System can be effectively and efficiently modified without introducing defects or degrading existing product quality	[27]
Replaceability	Product can be replaced by another specified software product for the same purpose in the same environment.	[27]
Compatibility	Capability of a product to exchange information with other products, and/or to perform its required functions while sharing the same common environment and resources	[60]

Continued on next page

Table 5.1 – continued from previous page

Term	Definition	References
Reliability	Capability of a product to perform specified functions under specified conditions for a specified period of time without interruptions and failures	[60]
Security	Capability of a product to protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors	[60]
Portability	System can effectively and efficiently be adapted for different or evolving hardware, software or usage environments.	[27]
Extensibility	The effort required to implement an extension.	[62]
Interoperability	Capability of a system to exchange information with other systems and mutually use the information that has been exchanged.	[60]
Co-existence	The product can perform its functions efficiently while sharing its environment and resources with other products	[21]
Availability	Capability of a product to be operational and accessible when required for use	[60]
Usefulness	User is satisfied with their perceived achievement of pragmatic goals.	[27]
Greenability	The degree to which a product energy and the resources are optimised, and the product can be used over a long period.	[32]
Functional suitability	Capability of a product to provide functions that meet stated and implied needs of intended users when it is used under specified conditions	[60]
Impactibility	Acceptance towards environment impacts with focused on the way of software is created, used, maintained and disposed with minimal impacts on environment acceptance towards economic impacts in developing software with lower cost of development and maintenance to survive acceptance towards social impacts of user and software functions are connectedness to each other to satisfy on using a product or system	[37]
Perdurability	Idea of producing sustainable software security products that have longevity and are adaptable and recyclable, i.e., increasing those aspects that make the software last for long time with the ability to adapt to change without losing its functionality related to its quality	[63], [64]
Freedom from risk	Degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment	[60]
Agency	The ability to act in an environment	[28]
Context coverage	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified	[60]
Effectiveness	Accuracy and completeness with which users achieve specified goals	[60]
Performance efficiency	Performance relative to the amount of resources used under stated conditions	[60]

Continued on next page

Table 5.1 – continued from previous page

Term	Definition	References
Resource utilization	Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements	[60]
Capacity	Degree to which the maximum limits of a product or system parameter meet requirements	[60]
Economic risk mitigation	Degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use	[60]
Health and safety risk mitigation	Degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use	[60]
Environmental risk mitigation	Degree to which a product or system mitigates the potential risk to property or the environment in the intended contexts of use	[60]
Satisfaction	Degree to which user needs are satisfied when a product or system is used in a specified context of use	[60]
Accessibility	Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use	[60]

5.2.2.5 Collapsing functionality

While attempting to enhance the readability of the catalog, another feature was introduced alongside the existing categorizations. This feature allows users to collapse parts of the catalog to simplify navigation and reduce visual clutter. By holding down a specific key(the shift key in the current version) and clicking on a requirement within the catalog, users can hide the detailed requirements connected to the selected item. Although the latest categorization improvements have made this feature less essential, it was kept since the ability to collapse parts of the catalog does not interfere with other functionalities. It may be useful if the catalog becomes more complex or if new requirements are added in the future.

5.3 Application example

Using our catalog and plugin in a real-world-inspired example helps assess its utility and impact in sustainable software development. In this section, we applied the plugin and catalog to an example which can be seen in Appendix C. This scenario involves a case study of an automatic toll system called ViaVerde. We focused on mapping system features to our catalog identifying goals, tasks, and qualities of the system, and connecting them with the sustainability requirements. The goal of applying our work to a real scenario is to demonstrate how it can be used to improve sustainability of real world software systems. Whether a system is in it's planning stages or already exists and wants to implement or evaluate it's sustainability the plugin works the same.

5.3.1 ViaVerde system

The system selected to test our catalog and plugin was the ViaVerde toll collection system[2], which has been operational since April 1991 on Portuguese highways. This system is crucial for facilitating urban mobility by automating toll payments. Authorized drivers can pass through designated lanes, marked with the ViaVerde logo, where they are automatically charged after passing the toll gate. To take advantage of these automatic toll gates, authorized vehicles must be equipped by their drivers with an active electronic identifier, which drivers must register and pay for. Once registered, the electronic identifier, placed on the vehicle's windshield, allows for automatic toll charges via direct debit when passing through the specialized toll lanes. The electronic identifier is read by sensors installed at the toll gates, and the information contained within is used to determine the appropriate account to debit. The toll amount depends on several factors: the distance traveled, the type of vehicle (e.g., trucks incur higher charges than cars), and the specific highway concession. As the vehicle passes through the toll gate, a green light indicates a successful transaction. If an issue arises—such as an unauthorized vehicle, an invalid electronic identifier, or a discrepancy in vehicle classification—a yellow light is displayed, an alarm sounds, and the vehicle's license plate is photographed. This process helps address potential issues, including fining unauthorized users if necessary. This example shows a practical application of our catalog to a simplified version of the ViaVerde system and plugin by demonstrating how they can help evaluate and enhance software systems.

5.3.2 Choosing sustainability requirements

During this application we started by identifying what were the sustainability requirements that made the most sense to include as part of the system. The next sections show which of the requirements were chosen, grouped by their sustainability dimension.

5.3.2.1 Requirements only belonging to the Individual Dimension

The following are the requirements that were chosen that belong solely to the individual dimension:

- **Privacy:** In any system that handles sensitive payment information, safeguarding the privacy of both the driver's and the vehicle's data is essential. We can achieve this through several measures, such as encrypting all sensitive information stored or transmitted within the system. By securing these details, the system reduces the risk of unauthorized access, which could otherwise expose users to various threats. For instance, if this information were compromised, users could be at risk of identity theft or financial fraud. Attackers accessing private data, such as personal identification or bank account details, could exploit it. Therefore, ensuring strict privacy protocols is essential to protecting users and maintaining trust in the system.

[2] ViaVerde Website

5.3.2.2 Requirements only belonging to the Environmental Dimension

These requirements belong soloely to the environmental dimension:

- **Infrastructure Optimization:** The integration of automatic toll booths, such as those used in the ViaVerde system, has a profound impact on infrastructure optimization. These automated toll booths must be installed alongside traditional toll systems, necessitating the creation of dedicated lanes for vehicles using the ViaVerde system. This presents an additional challenge for road planners and engineers, as they must carefully design roadways and toll plazas to accommodate both types of tolls without confusing drivers. Road layouts must ensure clear signage and smooth transitions between manual and automatic lanes to avoid traffic issues. Failure to optimize the infrastructure could lead to user frustration, accidents, and traffic congestion, especially during peak hours. Additionally, road space must be used efficiently to balance the needs of drivers using traditional toll booths and those with automated systems. Infrastructure must also account for future growth and technological advancements, ensuring the system can scale as more drivers adopt automatic toll payments. This requires strategies that accommodate current traffic patterns and also anticipate future increases in system usage. Effective infrastructure optimization ultimately plays a vital role in improving the overall user experience, reducing delays, enhancing road safety, and ensuring the toll collection process is as efficient and seamless as possible.

- **Supportability:** Given that the system is widely used, it is important that it is designed with supportability in mind. A highly supportable system significantly improves the resolution time for any technical issues that may arise. Whether addressing payment failures, system malfunctions, or updates, having a system focus on supportability allows faster troubleshooting and maintenance. This not only reduces downtime but also improves user satisfaction by ensuring that issues are resolved efficiently. Furthermore, future-proofing the system by making it adaptable to technological changes or third-party services ensures long-term sustainability and ease of maintenance.

- **Environmental effects:** Traditional toll booths rely on paper receipts and require the presence of workers, contributing to resource consumption and waste generation. By transitioning to automatic toll booths, the amount of paper used is significantly reduced, and waste associated with human-operated booths, such as printed receipts and on-site consumption, is eliminated presenting an opportunity for a positive environmental impact. However, the automatic booths introduce new environmental challenges. These booths rely on electronic identifiers and sensors, which may contribute to increased electronic waste, particularly when devices reach the end of their lifecycle. The added infrastructure also requires continuous electricity to power sensors, lights, and automated systems, leading to higher energy consumption

over time. While these electronic elements can result in increased energy use, the reduction in road congestion that automatic toll booths enable can help mitigate the environmental impact of vehicles. Less congestion means fewer idling cars, which directly translates to reduced fuel consumption and lower emissions.

- **Process efficiency:** Automating the toll payment process significantly enhances efficiency by streamlining the entire process. With the system in place, drivers no longer need to stop, wait in line, or manually handle payments, which not only speeds up the toll collection process but also reduces delays for drivers. The automatic system processes payments through direct debit as vehicles pass through the toll gates, making it faster and more convenient for drivers and the tolling authority.

5.3.2.3 Requirements only belonging to the Economic Dimension

The following chosen requirements only affected the economic dimension:

- **Cost minimization:** Introducing automatic toll booths significantly reduces operational costs by eliminating the need for regular toll booth workers. While there are ongoing expenses related to electricity consumption for the electronic components, the overall savings are considerable compared to the cost of employing personnel to collect tolls. In addition to labor savings, the maintenance costs for automatic toll systems are generally lower. Electronic components typically require less frequent maintenance than traditional toll booths, which involve maintaining physical structures like gates, counters, and booths where workers would be. Reducing the need for repairs and upkeep related to wear and tear from daily human interaction. The long-term benefit is a streamlined system that requires fewer resources, both in terms of manpower and physical infrastructure, leading to a more cost-efficient operation overall.

- **Increased productivity:** Automatic toll booths significantly boost productivity by processing a greater number of vehicles compared to traditional toll booths manned by workers. These automated systems operate continuously without the need for breaks, ensuring a steady flow of traffic throughout the day. This higher throughput is apparent when comparing the low congestion in lanes dedicated to automatic toll booths with the longer wait times often seen at regular booths.

- **Value:** The automatic toll collection system generates significant value for the company through a combination of upfront and recurring revenue streams. By charging users for the electronic identifier, the company secures initial revenue. This one-time payment covers the cost of manufacturing and distributing the devices. In addition to the initial fee, users are required to make monthly payments. This recurring revenue model creates a stable and predictable income stream for the company, ensuring financial stability even if toll usage varies. This approach helps balance money

flow and supports long-term system maintenance and upgrades. The convenience generated by the system not only improves user satisfaction but also encourages more frequent use of the toll roads, further increasing revenue.

5.3.2.4 Requirements affecting both the Environmental and Economic Dimensions

The next requirements are ones that have a direct effect on both the economic and environmental dimensions at the same time, but not the individual dimension:

- **Security:** Given that the system manages important payment related information, maintaining robust security measures is necessary to protect against unauthorized access and data breaches. Key measures include ensuring data integrity, which involves verifying that the information remains accurate and unaltered during transmission and storage. This helps prevent tampering or corruption of the data. Data encryption is another fundamental security practice. Encrypting sensitive information, such as payment details and personal identifiers.

- **Scalability:** Scalability is a critical consideration for any system that must adapt to changes in infrastructure, such as a toll collection system. Given that toll gates can be modified whether by being added, removed, or relocated based on evolving road planning requirements the system must be designed to accommodate these changes efficiently. A scalable architecture ensures that the system can handle varying loads and adapt to different scales of operation without performance degradation. Leveraging technologies such as container orchestration platforms like Kubernetes can significantly enhance scalability. Kubernetes automates the deployment, scaling, and management of containerized applications, enabling the system to dynamically adjust its resources based on demand. This flexibility allows the toll collection system to scale up during peak traffic periods or scale down during off-peak times, optimizing resource usage and maintaining system performance. Effective scalability also involves ensuring that the system's data management and communication protocols are designed to handle increased loads without bottlenecks. Implementing load-balancing mechanisms and distributed databases can further support this adaptability, ensuring that the system remains responsive and reliable as it grows or shrinks. By prioritizing scalability, the toll collection system can remain agile and resilient, capable of adapting to changes in infrastructure and traffic patterns while continuing to provide efficient and reliable service.

- **Replaceability:** Electronic identifiers, and the other electronic components, have a finite lifespan due to wear and tear, technological obsolescence, or technological advancements. Therefore, these identifiers must be designed with replaceability in mind to ensure the continuous availability and functionality of the system for customers. This allows for straightforward integration of new identifiers, whether they replace outdated components, upgrade to address system needs, or replace

worn down and broken electronic components. Planning for replaceability involves considering the availability of replacement parts and support services, and ensuring that replacement identifiers are readily accessible.

- **Maintainability:** Given that the toll collection system will operate continuously, it is crucial to design it with ease of maintenance in mind. Ensuring the system's maintainability involves creating a structure that allows for efficient updates, repairs, and upgrades without significant disruptions. To achieve this, the system should be easily modifiable. A modular design enables the isolation of various components, allowing for individual parts to be updated or replaced as needed without affecting the overall system. This approach not only simplifies the maintenance process but also facilitates the integration of new features or improvements. Reusability is another important consideration. Developing components and systems that can be reused across different installations reduces the need to spend money on new components for example.

- **Reliability:** Given the large number of users and the importance of the functions of the system, the toll collection system must be reliable. Reliability ensures that the system remains operational at all times and is capable of recovering swiftly from any errors. The system should be designed with availability in mind, particularly regarding the electronic identifiers. Incorporating backup components and failover mechanisms to maintain availability even if some parts of the system experience failures. Ensuring that users can use the system at any time makes it more reliable. Also, the system should have recovery processes in place, particularly regarding payment processes. This involves implementing automated error detection and correction mechanisms that can address issues as they arise without significant user impact. For example, if a payment transaction fails due to a temporary system issue, the system should be able to retry the transaction at a later date or provide the user with an alternative way to pay at a later date.

- **Portability:** In the context of our use environment, the electronic identifiers used in the toll collection system must be designed with portability in mind, given their need to be in vehicles with limited interior space. Portability ensures that the devices can easily fit in various types of vehicles without causing inconvenience to users. To achieve portability, electronic identifiers should be compact and lightweight. Their design should prioritize minimal physical dimensions and weight while maintaining functionality and durability. Additionally, the electronic identifiers should be versatile in their installation options to accommodate different vehicle designs and user preferences. This flexibility allows for easy integration into several vehicle types, from passenger cars to commercial trucks.

5.3.2.5 Requirements that affected all three dimensions

These last requirements are have an effect on all three dimensions at the same time:

- **Accessibility:** The toll collection system significantly enhances accessibility by streamlining and simplifying the toll payment process. This improvement directly impacts users' ability to pay tolls efficiently and with minimal hassle. One of the primary ways the system enhances accessibility is by automating the payment process. Automated toll booths reduce the need for manual transactions and interactions with toll booth operators, which can be particularly beneficial for drivers who might otherwise face long waits or difficulties at traditional toll booths.

- **User Error Protection:** The toll collection system enhances user error protection by streamlining and automating the toll payment process. This approach effectively minimizes the potential for mistakes that can occur in manual or semi-automated toll systems. By eliminating the need for human toll booth operators, the system removes the possibility of errors such as incorrect charges, miscommunication, or misidentification of vehicles. With automated toll booths, the system ensures that toll payments are processed consistently and accurately, reducing the risk of billing errors that can occur with manual transactions. The system also simplifies the user experience by eliminating the need for drivers to interact with toll booth operators or navigate complex payment procedures. With a straightforward, automated process, drivers can focus on the primary task of driving without the distraction or stress of managing toll payments. This reduction in cognitive load helps prevent errors and enhances overall driving safety.

5.3.2.6 Actors

In this version of the application we added three actors to the model:

- **ViaVerde:** This actor represents the ViaVerde system, it was created through the plugin's catalog generation and had it's name changed to fit the system's identity. It contains the sustainability catalog and the goals and tasks of the system.

- **Electronic identifier:** The ViaVerde system uses electronic identifiers to interact with its users, so it's important to keep them in the model to show how they interact with the system, this actor represents them.

- **Sensors:** To read the electronic identifiers the system must have some sort of sensor to read the data contained within, this actor represents those sensors.

5.3.2.7 Goals affecting the Economic Dimension

After identifying the actors and the sustainability requirements we identified the key goals of the system, these were the ones that had effects on the Economic Dimension either directly or through a necessary task associated with it:

- **Profit for the company created:** Like any business, the primary goal of a company is to generate profit by providing a product or service. In the case of the ViaVerde system, the company achieves profitability through multiple revenue streams, including the sale or rental of electronic identifiers, subscription or maintenance fees, and the automatic toll charges incurred by users. Generating profit is fundamental to the company's sustainability and growth. Profit allows the company to reinvest in infrastructure, improve the system, and expand its services. It also provides the financial means to ensure consistent operations, reward stakeholders, and explore new opportunities.

- **Electronic identifier sold:** A key component of the ViaVerde business model is the sale of electronic identifiers, which are essential for enabling automatic toll payment. This goal reflects the company's focus on distributing these devices to its users. By selling these identifiers, ViaVerde not only facilitates the automation process but also generates an additional revenue stream.

5.3.2.8 Goals affecting all the Economic and Environmental dimensions

The following goals affected both the environmental and the economic dimensions at the same time:

- **Automatic toll booths:** A central goal for the business is to replace traditional toll booths with automated ones. This goal focuses on improving the toll collection process by reducing reliance on manual operations and enhancing the efficiency of the toll collecting process.

- **System maintainability ensured:** Due to the system's continuous and widespread use and the monetization model, maintaining its functionality and performance is essential.

5.3.2.9 Goals affecting all three dimensions

Lastly these goals affected all three dimensions:

- **Security of the system ensured:** Given that the system processes sensitive information, such as payment details and personal data, ensuring its security is a top priority. This goal reflects the need to protect user data from potential breaches, unauthorized access, and cyber-attacks. Implementing strong security measures,

such as data encryption and data integrity verification helps maintain the system secure.

- **Digital receipts only:** A key goal of the system is to eliminate paper receipts by transitioning fully to digital receipts. As automatic toll booths replace traditional ones, issuing paper receipts becomes obsolete, and receipts are generated electronically through direct debit transactions.

- **Tolls paid:** One of the system's primary goals is ensuring that tolls are paid efficiently and automatically. This goal reflects the system's dedication to handling the toll collection process.

- **Electronic identifier data read:** To ensure the system operates correctly, it is crucial to accurately read and interpret the data contained within the electronic identifiers. This data is essential for the functioning of the toll collection process, as it provides the information needed to identify the vehicle and process payments.

5.3.2.10 Tasks affecting the Individual Dimension

Following the identification of the system's goals we identified the tasks necessary for the system. Out of the tasks that were created only one affects the individual dimension exclusively:

- **Encrypt vehicle and driver data:** To ensure robust security for the system, it is crucial to encrypt all essential data related to vehicles and drivers. By encrypting both types of data, the system can prevent unauthorized access, mitigate the risk of data breaches, and safeguard personal and financial information.

5.3.2.11 Tasks affecting the Economic Dimension

The next tasks had an effect exclusively on the economic dimension:

- **Process direct debit:** For the system to effectively charge users, it must be capable of processing direct debit requests. The system initiates a direct debit request to the user's bank account whenever their vehicle has been detected at an automatic toll booth.

- **Charge subscription fee:** As a core component of the business model, the system must implement a subscription fee structure to generate revenue and sustain operations. This fee is for users accessing and utilizing the services provided by the system, ensuring that the business remains financially viable and capable of supporting ongoing maintenance and development and generating profit.

5.3.2.12 Tasks affecting all three dimensions

The following tasks had an effect on all three dimensions:

- **Use electronic identifier to handle toll payments:** The ViaVerde system is designed to use electronic identifiers for the complete management of toll payments, they will serve as identification of the vehicle and request direct debit payments according to what was read.
- **Verify data integrity:** Ensuring data integrity is a crucial component in maintaining the security and reliability of the system. Data integrity verification involves implementing mechanisms to confirm that the data remains accurate, consistent, and unaltered throughout its lifecycle.
- **Sell electronic identifiers:** Another element of the business model, selling electronic identifiers is crucial for generating revenue and facilitating the system's operation. These identifiers, which are essential for the automatic toll payment process, are sold to customers to enable their participation in the toll collection system. The sale of electronic identifiers provides another revenue stream for the business, helping to offset initial setup costs and electronic identifier production costs.

5.3.2.13 Tasks affecting the Environmental and Economic dimensions

The following tasks are the tasks that were created that have an impact on the Environmental and Economic dimensions at the same time:

- **Use Kubernetes architecture:** Kubernetes provides a framework for managing containerized applications, making it an ideal choice for addressing key aspects of system integrity, scalability, and maintainability. Kubernetes can automatically scale resources up or down based on demand, ensuring that performance remains optimal even during periods of high traffic. This elasticity helps manage workloads more effectively and maintains a consistent user experience. In terms of maintainability, Kubernetes supports rolling updates and rollbacks, which simplifies the process of managing application changes and addressing issues. It also provides monitoring and logging capabilities, allowing for quick resolution of any problems.

5.3.2.14 Resulting model

The following subsection contains snippets of the the resulting model from placing all with a focus on the ViaVerde components and their connections to the sustainability requirements. They can be seen in Figures 5.11 and 5.12. This example can also be found in its complete form in Appendix C.

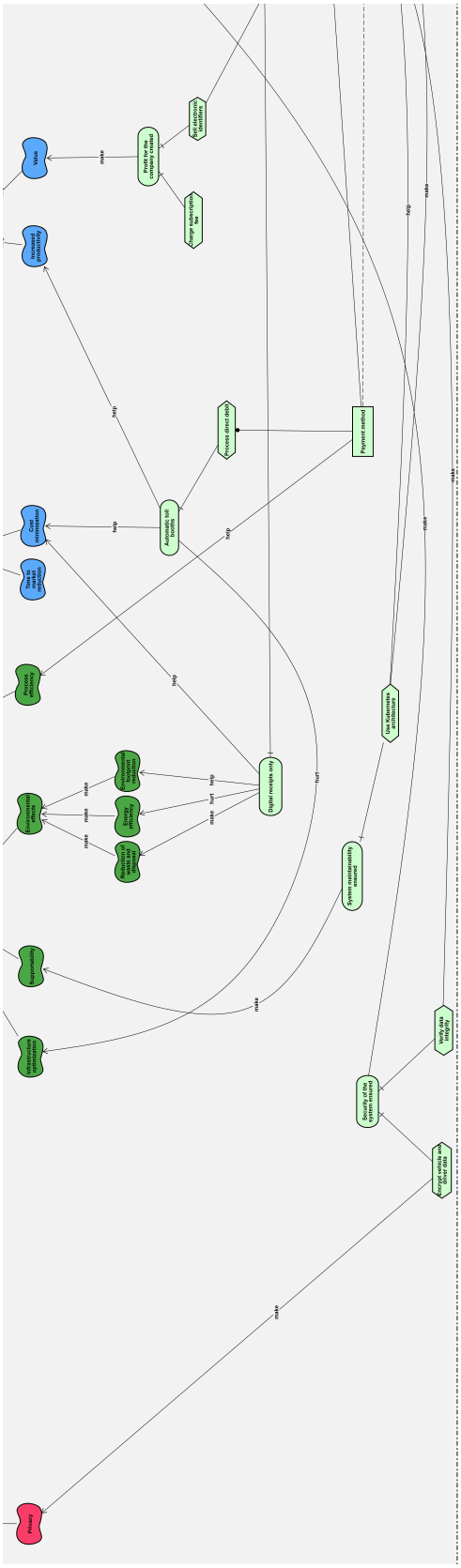


Figure 5.11: First half of the ViaVerde example snippet zoomed in

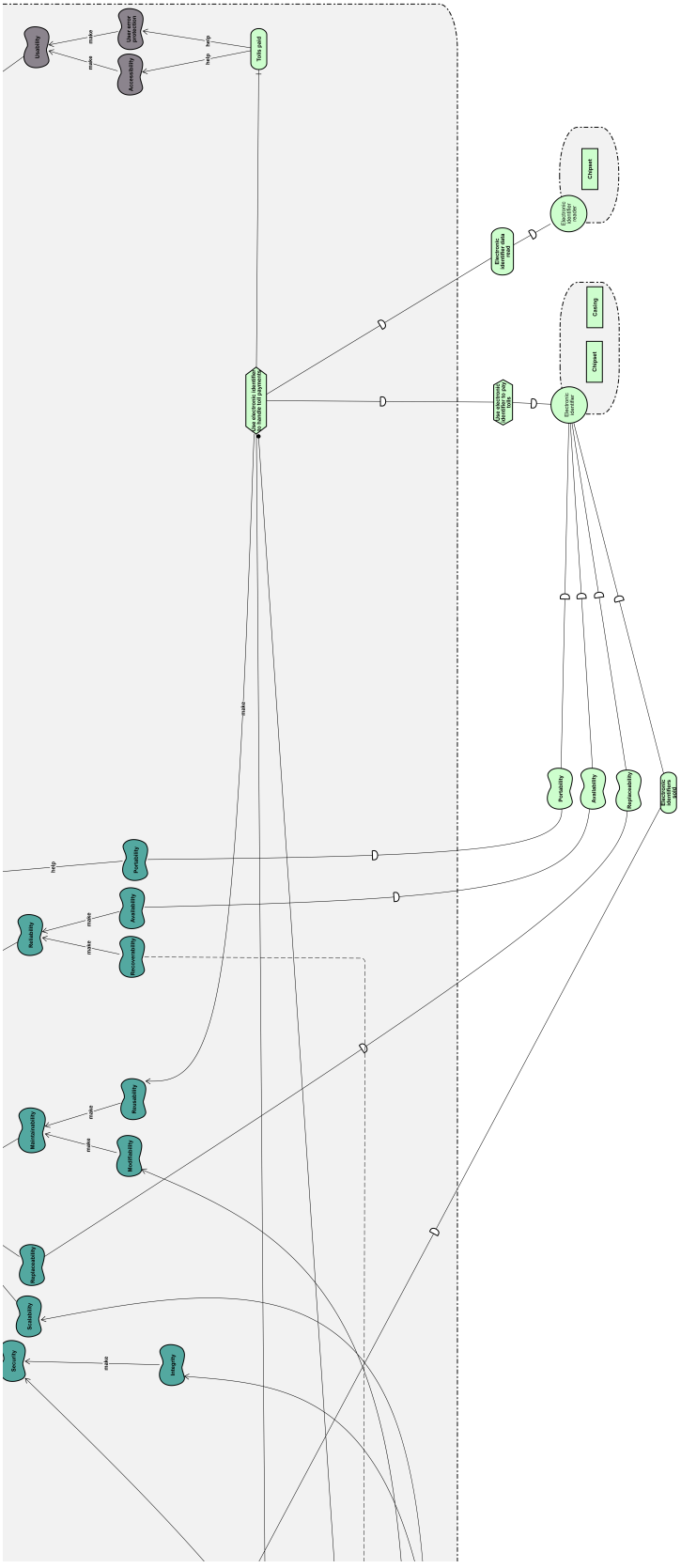


Figure 5.12: Second half of the ViaVerde example snippet zoomed in

5.4 Summary

We developed a robust sustainability plugin that addresses three of the five core dimensions of sustainability. The iStar framework and the piStar tool were good choices for developing our tool and catalog. The iStar framework effectively allowed us to manage non-functional requirements, while the open-source piStar tool offered ease of modification through the creation of a plugin, allowing us to implement new features. The new features implemented served to greatly improve piStar's functionality and added the possibility of generating models that include sustainability as one of its qualities.

6

Evaluation

In this chapter, we present an evaluation of the sustainability catalog and the supporting tool developed in our work. The evaluation process involved providing participants with a guide, which they used to analyze both the tool and the catalog. Participants were then asked to complete a questionnaire that requested feedback on various aspects of our project. This chapter outlines the structure of the guide, presents the results of the evaluation, and concludes with a discussion of the potential threats to the validity of the evaluation.

6.1 Guides

Two guides were created to help participants understand how the catalog and plugin work and to help them answer the questionnaire as accurately as possible. The first guide is a video tutorial produced using *Clipchamp*, a Windows video editing tool. The second guide is a written document, created in *Google Docs*, including text and images. Participants were allowed to use either guide, or both if preferred, as both contain the same information about our work. Both guides follow the same structure, consisting of two main sections:

- **piStar Functionality** - This first section provides a brief overview of the piStar tool, highlighting the functionalities available in the base version of the tool.
- **Plugin Functionality** - The second section explains all the features of the developed plugin and demonstrates how our work can be applied to software projects.

6.1.1 Video guide

The video guide has a duration of 6 minutes and 35 seconds[1]. The guide begins with an introduction to the piStar tool, briefly showcasing all of the default functionalities it includes. We advised participants familiar with piStar to skip to 1:38, where the introductory segment. After the introduction, the video demonstrates the functionalities

[1] Video Guide

of the developed plugin, which continues for the rest of the video. The demonstration concludes with the ViaVerde example from Chapter 5.

6.1.2 Written guide

The written guide was a *Google Docs* document spanning nine pages[2]. The document consisted of detailed instructions, supported by images where necessary. Like the video guide, it begins with a brief introduction to the piStar tool, outlining its basic functionality. Following the introduction, there was a section dedicated to explaining all of the features of the supporting tool. The guide concluded with the ViaVerde example from Chapter 5, illustrating how our work can be applied in real projects.

6.2 Questionnaire

The goal of the questionnaire was to assess whether the sustainability catalog and supporting tool were useful for developers. Specifically, we aimed to determine whether these tools helped participants understand and implement sustainability in their software projects, and to identify their general positive and negative aspects. Additionally, we gauged participants' interest in using the catalog and tool as a foundation for their future projects.

The questionnaire[3] was created using a *Google Forms* document. It was composed of several sections composed of questions, with the first set was designed to gather demographic information about the participants, while the remaining sets focused on evaluating specific facets of our work. The intent was to assess the most important aspects of the plugin and catalog as accurately as possible. To achieve this, we utilized *Google Forms* a tool that allows users to easily build questionnaires and extract information from its answers [65]. The questionnaire was divided into six sections:

- **Introduction** – The first section introduces the purpose of the questionnaire, assures participants that their responses will be confidential, and presents links to both the created guides and the online tool.
- **Personal Data** – The second section includes questions about participants' personal information, such as their age range, level of academic education, degree, and experience in software development. This section helped us understand the heterogeneity of the participants.
- **Guide Questions** – This section contains questions about the provided guides. Through these questions, we aimed to analyze their usefulness and determine which guide was more effective for participants.

[2] Written Guide
[3] Google Forms Questionnaire

- **Quality of the Catalogue** – This section includes questions regarding the quality of the created catalog, specifically addressing readability, utility, the likelihood of using it as a basis for future projects, and whether participants feel it enhances their understanding of sustainability.

- **Quality of the Tool** – This section focuses on the developed plugin, with questions aimed at assessing its ease of use, the likelihood of using it in future projects, and its effectiveness in facilitating sustainability implementation.

- **Open Feedback** – In the final section, participants were invited to provide open feedback. We specifically asked if they had additional requirements they felt should be included in the catalog, whether the categorization made sense, and what they perceived as the strengths and weaknesses of both the catalog and the plugin.

Each section contained questions categorized into three types:

- **Likert Scales** – Likert scale questions asked participants to evaluate a statement or general feeling regarding a particular characteristic of our work. The scale ranged from 1 to 5, with 1 representing a very negative response, 2 a less negative response, 3 indicating neutrality or uncertainty, 4 a positive opinion, and 5 a very positive opinion.

- **Multiple Choice** – Multiple choice questions provided participants with predetermined answers.

- **Open Answer** – The final type of question was the open answer, allowing participants to express their opinions on a topic in written form without constraints.

6.3 Participant selection

For the evaluations of our work we started by considering three groups of people:

- **Bibliography Authors** – The first group considered comprised authors from the bibliography with expertise in computer science or sustainability. These authors were selected from the studies outlined in Chapter 3. They were chosen for their ability to provide an experienced perspective during the evaluation of our model's quality.

- **Academic Students** – This group consisted of students who expressed interest in the concepts presented in our work and volunteered to provide feedback. Although they may not have as much experience as the other participants, their insights can help us understand the accessibility of the concepts and the usability of the plugin for less experienced users.

- **Practicing Software Engineers** – We contacted several developers active in the software development industry to gather their input. As current practitioners, their feedback is invaluable for understanding how our work contributes to the software development industry.

An email was sent to each individual, containing a formal introduction, justification for receiving the email, and an invitation to participate in the evaluation, along with a link to access the questionnaire. Responses were collected over a period of 12 days, resulting in a total of 13 out of 80 potential participants giving us their feedback.

6.4 Discussion of the Results

After the evaluation period, we analyzed the results to draw our conclusions. Since *Google Forms* allows us to create bar charts for Likert scale questions and pie charts for multiple-choice questions, we utilized these features, eliminating the need for additional software to generate visualizations. In this section, we will discuss the results of each question. For the Likert scale and multiple-choice questions, we will analyze the observed patterns, and for the open-ended feedback questions, we will summarize the responses and highlight key points based on the participants' input.

6.4.1 Participant data

In this section, we will discuss the heterogeneity of the participants who answered our questionnaire.

6.4.1.1 Age of the Participants

The results indicate that a majority of the participants fell within the age range of 23 to 35 years old representing 69.2% of the population, followed by those aged 36 to 45 years at 23.1%. The only other age group represented was 46 to 55, accounting for 7.7% of participants. There were no participants in the 56 to 65 years or over 65 years age ranges. While our goal was to achieve a diverse participant pool spanning various age ranges to assess how the catalog and plugin were perceived across different experience levels, most respondents were from the younger age group. As a result, the data collected during the evaluation is skewed toward their perspectives, with limited insights from more experienced participants. In Figure 6.1 we can view the data regarding participant ages.

6.4.1.2 Level of Academic Education

The level of academic education among participants showed greater variety. Most participants held a Bachelor's degree (53.8%), while a significant portion had a Doctorate/PhD (30.8%). Additionally, we had a participant who were industry practitioners with only a

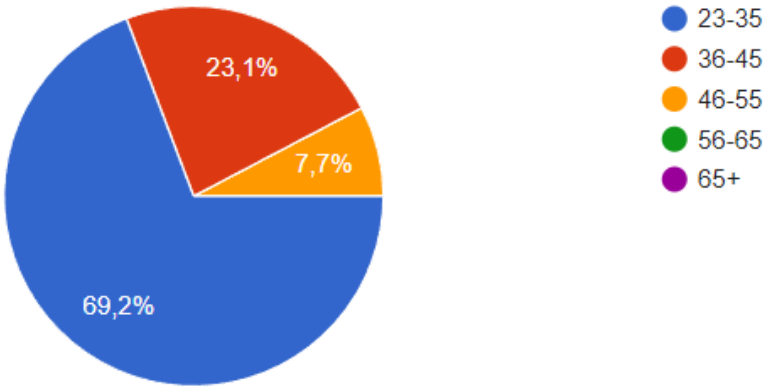


Figure 6.1: Percentage of participants by age

High School Diploma; and a participant with a Master's Degree (7.7%). While the age range of the participants had a tendency to be towards the younger side, we still managed to have diversity in the level of academic education, ensuring that we have perspectives from all kinds of developers. This data is illustrated Figure 6.2.

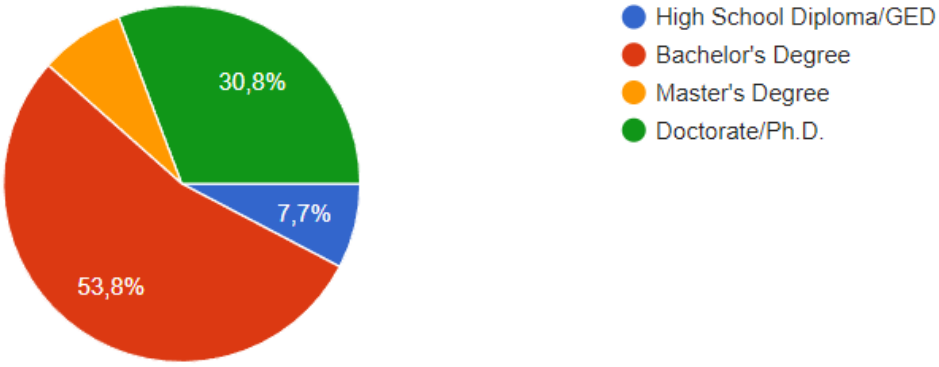


Figure 6.2: Percentage of participants by degree of education

6.4.1.3 Degrees held by the participants

Out of our participants, only 8 provided information regarding the area of their degrees. The question asked participants to specify their degrees as long as they were related to software development or sustainability. The lower response rate suggests several possibilities. Some participants may not have completed their degrees, which would explain their lack of response, especially among those in the academic group. Additionally, some participants might not have held a degree relevant to the question and thus left it blank. There is also the possibility that participants felt uncomfortable answering this question or simply forgot to respond. Among the responses we did receive, all were related to computer science or sustainability. The most common answer was "Computer Science" (50%),followed by the remaining answers having all the same number of responses."Informática

(FCT-UNL)" (12.5%) is essentially the Computer Science degree from FCT-UNL. There was a participant with a PhD in artificial intelligence and machine learning, also within the computer science field. We had participants that hold a degree in sustainability of software (12.5%) and a participant that has a Master in software engineering (12.5%). Our objective was to have feedback from participants that specialize in computer science and participants who specialize in sustainability, having both present in our responses helped validade our work from both perspectives. Figure 6.3 demonstrates our data.

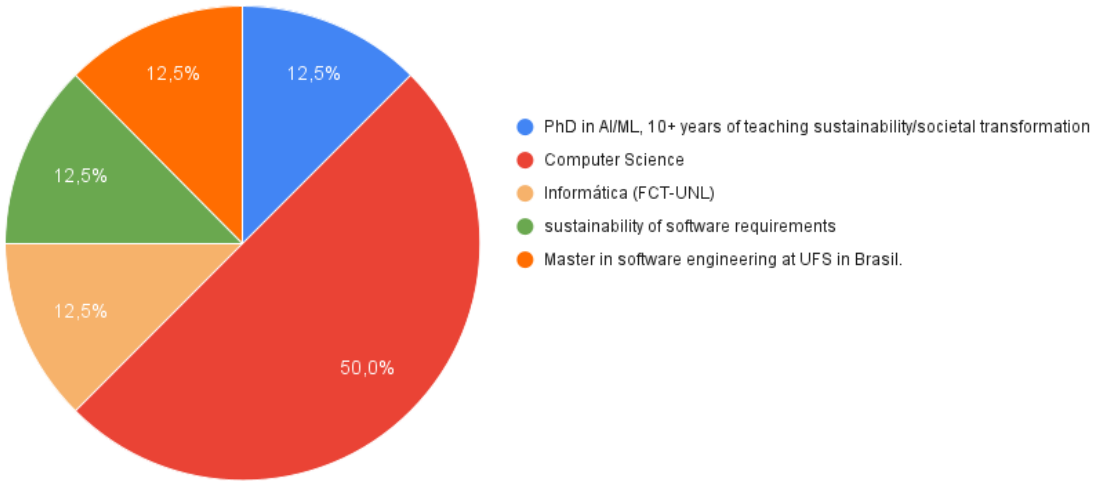


Figure 6.3: Percentage of participants by degree type

6.4.1.4 Experience of the participants

The final question regarding participants' personal data asked about their experience developing software systems, measured in years. The responses revealed a diverse range of experience levels. Most participants reported having 3 to 5 years of experience (53.8%), followed by those with 1 to 2 years of experience (15.4%), as well as he 6 to 10 years range(15.4%). The more than 10 years and less than 1 year categories received equal representation (7.7%). Despite the lack of participants in the older age ranges we managed to get responses from a large variety of experience ranges. Figure 6.4 presents this data visually.

6.4.2 Quality of the catalog

The purpose of the questions in this section was to analyze participants' opinions on the catalog. These questions were crucial, as the catalog serves as the foundation of our work. Participants' perspectives on whether the catalog facilitated an understanding of sustainability in software development, and its overall utility, were fundamental in assessing the value of the catalog we created.

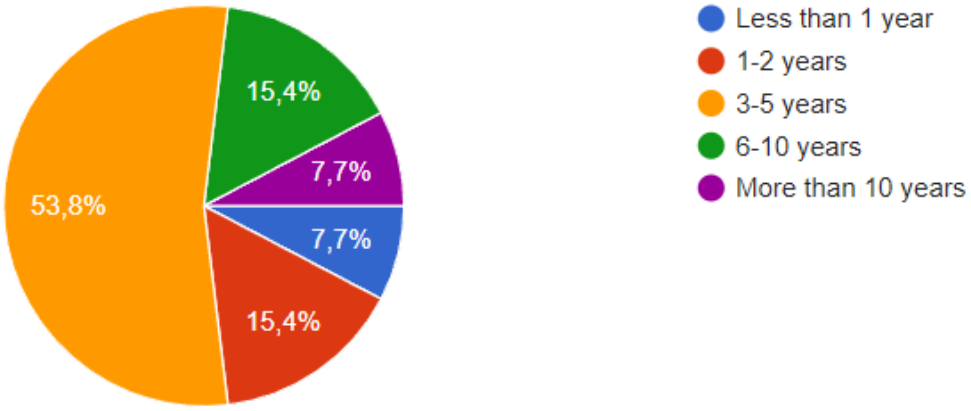


Figure 6.4: Percentage of participants by years of experience

6.4.2.1 Catalog readability

The first question in this section assessed whether participants found the catalog easy to read. This question was particularly important, as one of the main challenges during the early stages of constructing the catalog was ensuring its readability. Participants were asked to rate the catalog's readability on a scale from 1 to 5, where 1 represents "Very weak" readability and 5 represents "Very good" readability. As shown in Figure 6.5 the majority of participants felt the catalog was easy to read. Specifically, 53.8% rated its readability as 4 out of 5, which we interpret as "Good." The second most common rating was 3 out of 5 (30.8%), indicating satisfactory readability. The two least popular ratings: 2 out of 5 and 5 out of 5—each received responses from 7.7% of participants with no participants rating the catalog a 1. These results suggest that overall, participants consider the catalog to have good readability, indicating that we successfully achieved our goal of enhancing its readability during development.

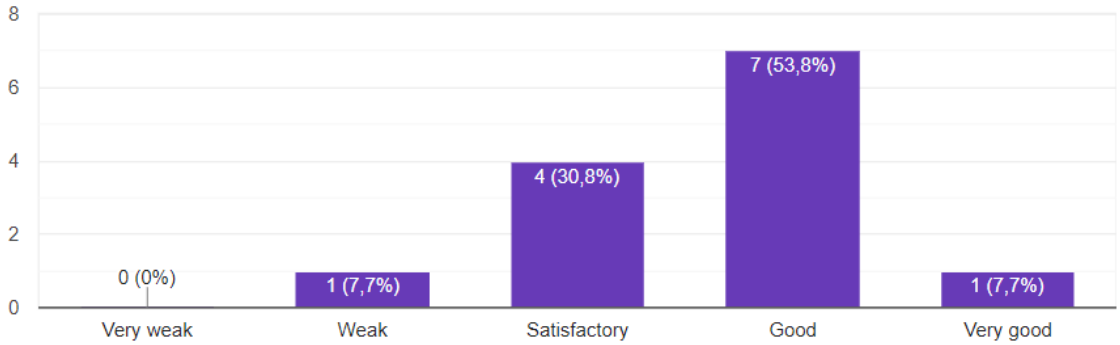


Figure 6.5: Participant opinions on the catalog's readability

6.4.2.2 Catalog utility

The next question asked participants to evaluate how useful they found the catalog for understanding potentially relevant sustainability requirements in software projects. Similar to the previous question, participants rated its usefulness on a scale from 1 (Not useful) to 5 (Very useful). The results, as illustrated in Figure 6.6, indicate that the majority of participants consider the catalog to be useful, with 69.2% rating it 4 out of 5. The second most common rating was 5 out of 5, indicating that 15.4% of participants found it very useful. The only other ratings with responses were the scores of 2 out of 5, indicating weak utility, and 3 out of 5 indicating mediocre utility, both chosen by 7.7% of participants; the remaining options received no selections. These responses suggest that the catalog, in its current form, serves as an effective tool for understanding relevant sustainability requirements in software projects.

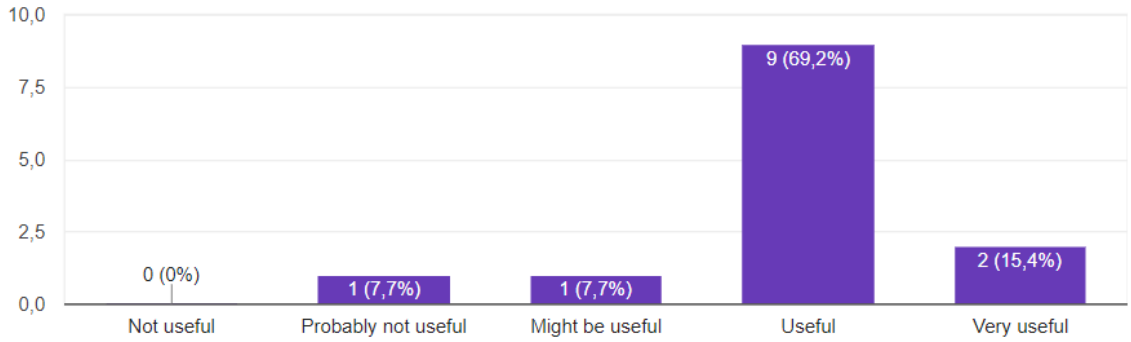


Figure 6.6: Participant opinions on the catalog's utility

6.4.2.3 Clarity of the concepts

Another point of interest for our work was how clear the concepts in the catalog were to the participants. Given that our catalog addresses concepts related to sustainability that software engineers may not be familiar with, it was important to assess how effectively the catalog presents these ideas. Participants rated the statement "The concepts in the catalog are clear" on a scale from 1 (Strongly disagree) to 5 (Strongly agree). A majority of participants (53.8%) indicated that they agreed. The second most common response was neutral, selected by 23.1% of participants. Additionally, 15.4% of participants strongly agreed, while 7.7% strongly disagreed. Although the overall feedback was positive, the increase of neutral responses suggests that clarity may need to be improved. Figure 6.7 presents the results for this question.

6.4.2.4 Interest of using the catalog in the future

This question aimed to gauge how interested participants would be in using our sustainability catalog as a foundation for implementing sustainability in their future projects.

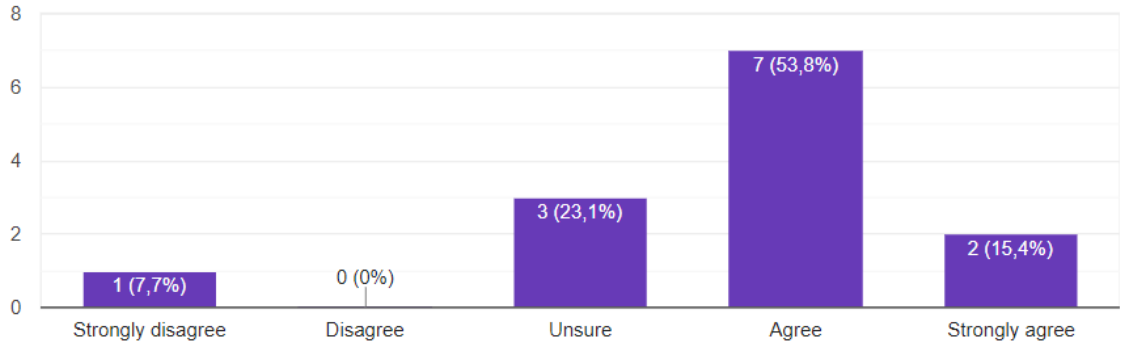


Figure 6.7: Participant opinions on the catalog's clarity

They rated their likelihood on a scale from 1 (not likely) to 5 (very likely). The most common response was a neutral rating of 3, chosen by 46.2% of participants. Ratings indicating likely (4) and very likely (5) were each selected by 23.1% of participants. Notably, no one rated their likelihood as 2 (unlikely), but one participant representing 7.7% of the participants chose 1, indicating they most likely would not use the catalog at all. These responses suggest that while the catalog has commendable qualities, its application in real projects may be challenging. The higher number of neutral responses, compared to previous questions, could reflect the unpredictable nature of software development, which could prevent participants from committing to using the catalog, as decisions in projects are often influenced by various factors and not made by a single individual. Despite this, it is encouraging that a total of 46.2% of participants indicated they were likely or very likely to use the catalog in the future. Figure 6.8 presents the data for this question.

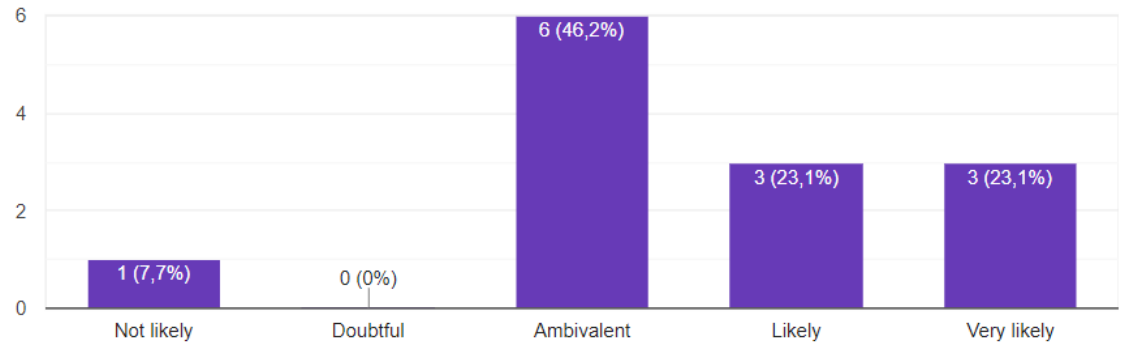


Figure 6.8: Participant interest in using the catalog

6.4.2.5 Understanding of sustainability

The final question in this section asked participants whether they felt the catalog provided them with a better understanding of sustainability. As with the other questions, participants rated their responses on a scale from 1 (strongly disagree) to 5 (strongly agree). The results showed that the majority (61.5%) agreed(4) that the catalog enhances their

understanding of sustainability, meanwhile 30.8% were neutral on it (3). The remaining 7.7% indicated strong disagreement (1). These results suggest that the primary objective of the catalog, to improve users' understanding of sustainability in software development, was mostly achieved. However, the presence of negative and neutral feedback indicates that there is still room for improvement. Notably, none of the participants selected the highest rating (5), suggesting that while the catalog is generally perceived as beneficial, further enhancements could increase its effectiveness. Figure 6.9 illustrates the results of this question.

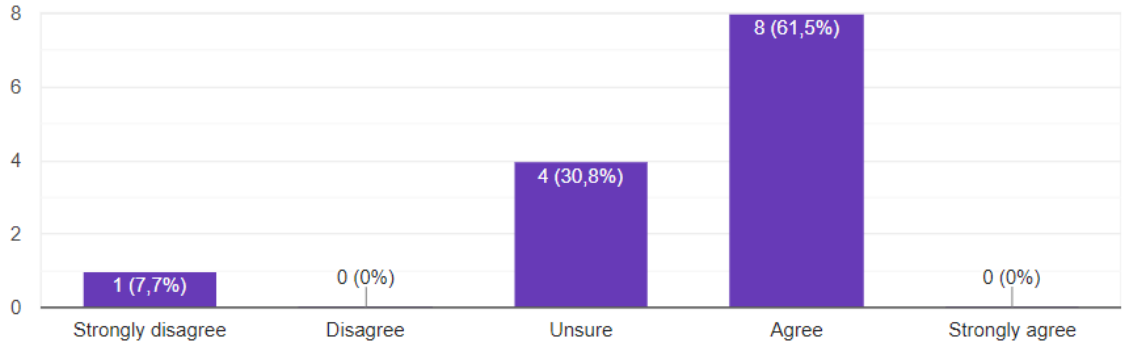


Figure 6.9: Participant opinions on the catalog's contribution towards understanding sustainability

6.4.3 Quality of the developed tool

The purpose of the questions in this section was to analyze participants' opinions on the supporting tool developed during our project. These questions specifically focused on evaluating whether the tool is a viable option for assisting developers in implementing sustainability in their software projects.

6.4.3.1 Ease of use

The first question addressed the ease of use of the tool. Participants were asked to rate their agreement with the statement that the plugin was easy to use, using a scale from 1 to 5, where 1 represented strong disagreement and 5 indicated strong agreement. The majority of responses were positive, with 61.5% rating the tool a 4. Scores of 5 and 3 were tied for second place, receiving 15.4% of the responses, while 7.7% rated it a 1. No participants rating the tool a 2. These results indicate that participants generally found the developed tool user-friendly, likely due to its strong foundation in the original piStar. Ease of use is essential for any tool, suggesting that our supporting tool is well constructed and will attract users and maintain their attention more easily, enhancing its overall value. Figure 6.10 displays the results obtained.

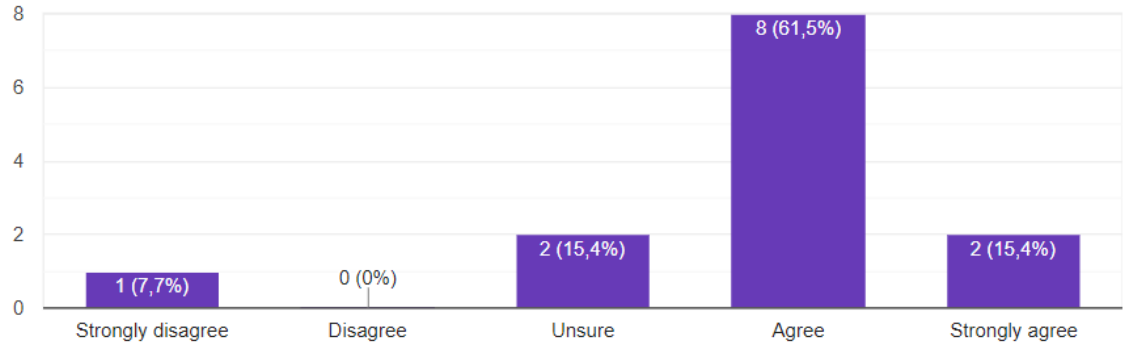


Figure 6.10: Participant opinions on the tool's ease of use

6.4.3.2 Likelihood of future usage

Similar to the questions about the catalog, we asked participants how likely they would be to use the developed tool. The results closely aligned with those for the catalog. The most common response was a score of 3, with 46.2% of participants indicating uncertainty about whether they would use the tool. A notable 30.8% selected a score of 4, suggesting they were likely to use it. Meanwhile, 15.4% gave a 5, the highest possible rating, and 7.7% gave it a 1. No participants selected the response of 2. When comparing these results to the similar question about the catalog, it becomes clear that while the tool enhances the catalog's usability, participants are not able to commit to using it. As mentioned earlier, decisions about adopting new tools are often made collectively, rather than by individuals alone. Therefore, even though a substantial portion of participants expressed a likelihood or strong likelihood of using the tool, this does not necessarily mean they would be able to use it as part of a development team. Despite this, the results are encouraging: the positive responses shown, even among those uncertain, suggests the tool has the potential to influence the adoption of sustainability in future projects. Figure 6.11 presents these results visually.

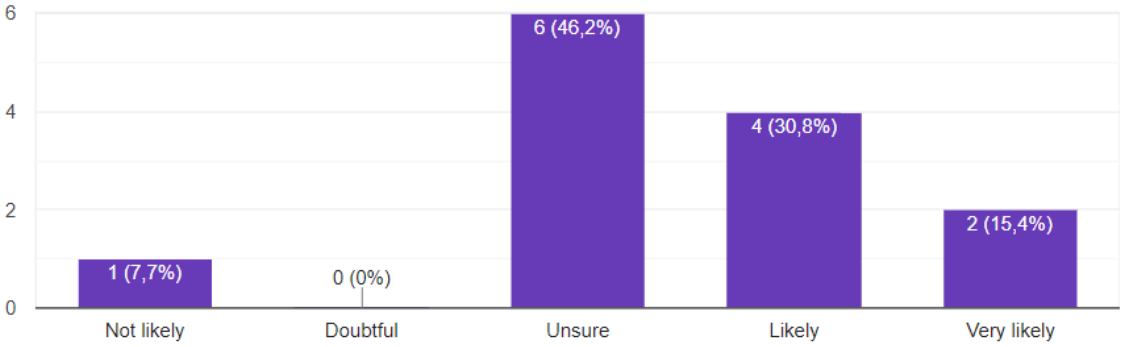


Figure 6.11: Participant interest in using the tool in the future

6.4.3.3 Impact on the implementation of sustainability

The final question regarding the tool asked whether it made implementing sustainability in software projects easier. The responses reflected those from the corresponding question about the catalog. A significant majority of participants (76.9%) indicated agreement, selecting a score of 4, which suggests they believe the plugin enhances the ease of integrating sustainability into their work. However, some participants did choose a score of 1, 3, and 5, (7.7%) expressing complete disagreement, uncertainty and complete agreement respectively. These results, illustrated in Figure 6.12, indicate that many participants perceive the plugin as a valuable resource for promoting sustainability in software projects. This perception by our participants adds further value to our work.

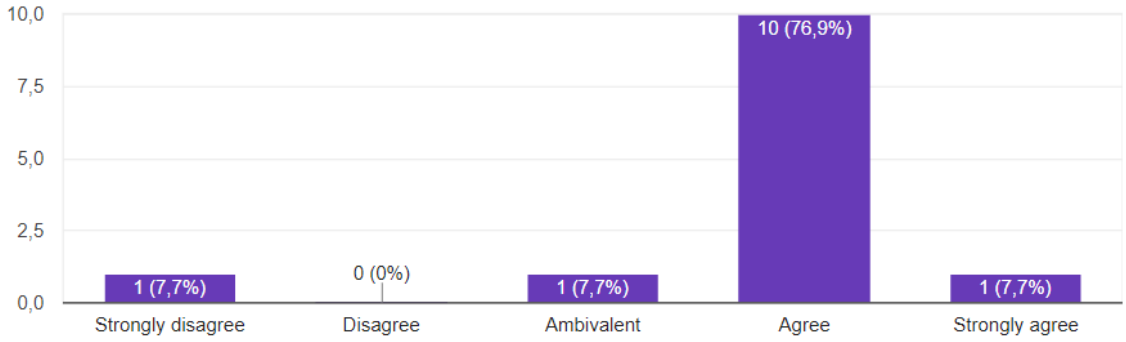


Figure 6.12: Participant opinions on the tool's ability to ease the implementation of sustainability

6.4.4 Guide evaluation

We included several questions to evaluate the guides provided for the execution of the evaluation. The questions were divided into those related to the text guide and those about the video guide. Since participants were not required to use both guides, the number of responses regarding each guide varied.

6.4.4.1 Textual guide clarity

The first question asked the participants if they thought the textual guide was clear in its instructions. The overall results were positive, with 16.7% of participants responding neutrally, 50% indicating that the instructions were clear, 25% rating them as very clear and 8.3% claiming the instructions were unclear. Since not all participants responded to this question we must contextualize within the complete participant population. The data translates to aproximately 15% of the total participants giving a neutral response, 46% indicating that the instructions were clear, 23% rating them as very clear and 7.6% indicating that the instructions were unclear. This distribution suggests that the guide was generally effective in communicating the concepts behind our work, which is essential for ensuring users can give useful feedback. Figure 6.13 contains the data on this question.

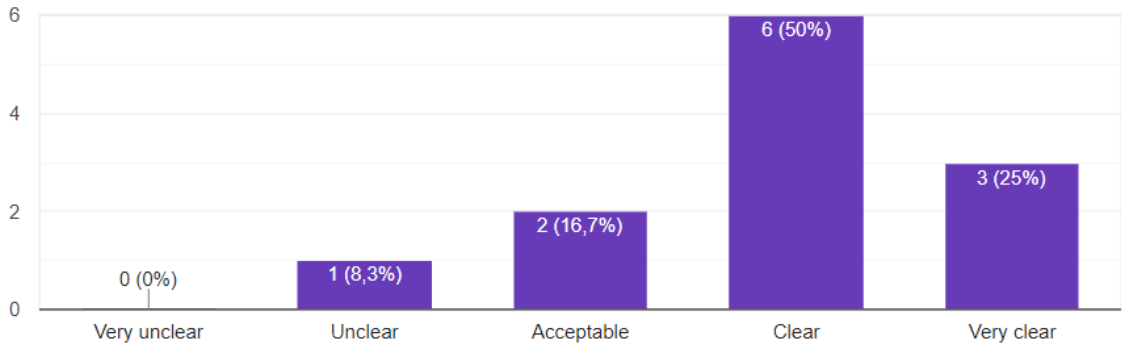


Figure 6.13: Participant opinions on the text guide's clarity

6.4.4.2 Video guide clarity

The next question asked participants about the clarity of the video guide's instructions. Unlike the text guide, not all participants responded to this question; only aproximately 61% (8 participants) chose to provide feedback regarding the video guide, indicating a preference for the textual format. Among those who did respond, the opinions were very positive, with 62.5% rating the video clarity as a 5 (very clear) and the other 37.5% as a 4 (clear). However, we should contextualize these results: only 61% of participants answered, the percentage of users rating the video as a 5 translates to aproximately 38.1% of the total population, and the user rating of 4 accounts for approximately 22.8% of the total population. Overall, it appears that the participants who engaged with the video guide found it to be very effective in conveying its content. Figure 6.14 displays the data for this question.

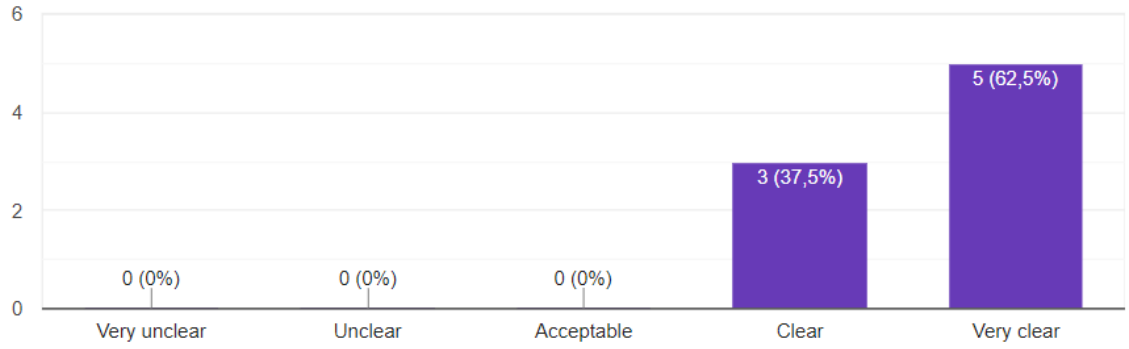


Figure 6.14: Participant opinions on the video guide's clarity

6.4.4.3 Quality of the text guide

Next, we aimed to obtain information on the overall quality of the text guide. This question was created to gather data regarding the general quality of the guide, which we could later compare with the written feedback from a follow-up question. The results were mostly positive: the majority of participants rated the guide a 4 (66.7%), while the remaining

ratings included a 3 (25%) and a 5 (8.3%). Contextualizing this data to the complete population since only 12 of the 13 participants answered: aproximately 61.3% gave the guide a 4, 23% rated it a 3 and 7.6% rated it a 5. This distribution suggests that while many found the text guide to be of high quality, there remains a small percentage of participants who felt it could be improved.Figure 6.15 displays the results for this question.

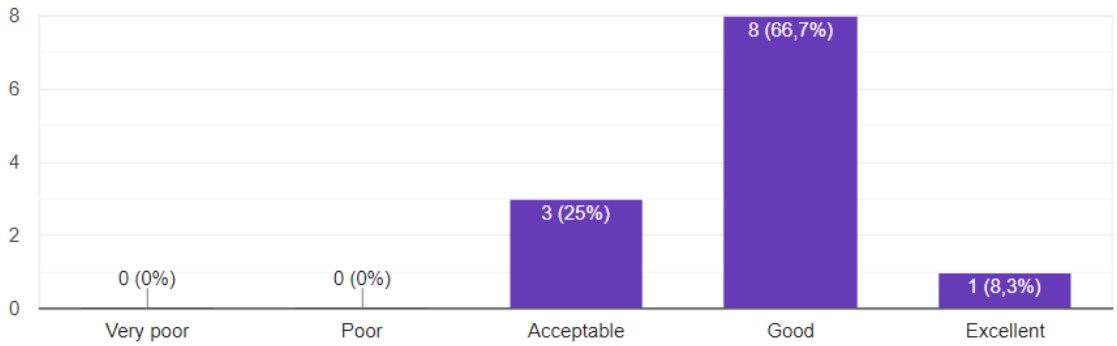


Figure 6.15: Participant opinions on the text guide's quality

6.4.4.4 Quality of the video guide

Following the same pattern, we asked about the overall quality of the video guide, aiming to compare these results with responses from a subsequent follow-up question. Consistent with previous findings regarding the clarity of the guide, only 61% of participants chose to respond to this question. The majority rated the video guide with a quality score of 4 (75%), while the remaining participants rated it a 5 (25%), as illustrated in Figure 6.16. Considering that only 61% of the participants answered this question we must add context to it: in reality, this translates to 45.7% of the total population having positive feedback and 15.2% of the total population having very positive feedback. These results indicate that the video guide was generally well-received, with most participants finding it to be of high quality and useful for understanding our work.

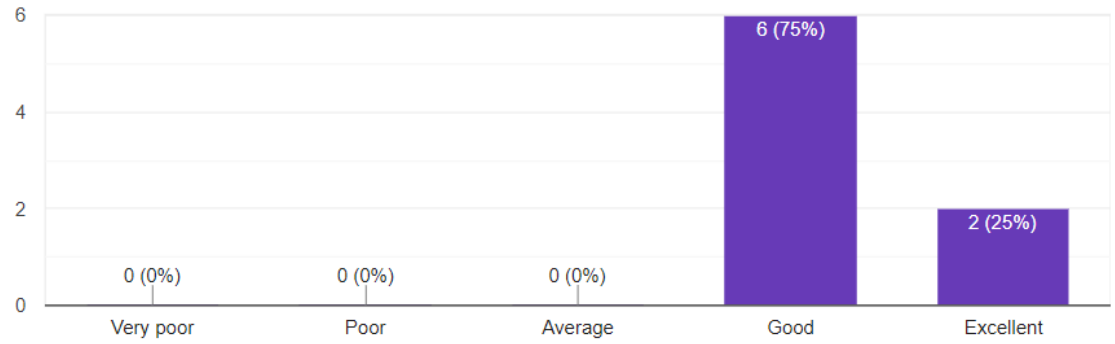


Figure 6.16: Participant opinions on the video guide's quality

6.4.4.5 Textual guide improvement suggestions

We asked participants to provide suggestions for improving our text guide, and we received five recommendations:

- "In my opinion, the figure on page 8 should include a brief description. Each figure should have a corresponding caption."
- "I suggest putting in both the explanation about the dimension concepts for the beginners users. Inserting the link to the modeling of the page 8. The figure is small or putting this in example in the tool. Considering that the guide is intended for tool usage, it is fine, but I believe it needs to go beyond just the tool itself. The catalog tutorial could also include the definition of each requirement, along with an example for its dimension. This would be extremely important for beginner professionals."
- "Some parts of the figure in the textual guide are unnecessary and can be cropped."
- "I missed examples in the guide, for example, of how I could use the requirements within a real system. The video made it clearer, but I missed it in the tutorial. I also wanted to better understand how the sustainability dimensions are combined. I missed an explanation about how to combine the dimensions in the catalog. The piStar image, as it was arranged, makes it difficult to read. If it were shown from top to bottom, I think it would be more interesting to view, but it was also too small. I missed the explanation of the example in the image as well."

The first suggestion focused on improving the figure present on page 8. Participants found the example difficult to read and noted the absence of a caption, which hindered understanding. One solution for this problem could be to use the model as an example within the plugin, similar to how piStar provides its examples in the help tab for easy loading. The second suggestion was to include definitions for each requirement, along with examples to assist beginners in comprehending the requirements more effectively. The third suggestion was to remove irrelevant parts of the example present on page 8. The last suggestion claimed that they believe there should be more examples in the guide on using the requirements in a real system and an explanation of how to combine the sustainability dimensions. The piStar image was difficult to read—too small and better suited to a top-down view. The example in the image also lacked explanation. Overall, these improvement suggestions are valid and would be beneficial to implement in future versions of our guide or tool. Enhancing the figure on page 8 would ensure clarity and improve the user experience. Likewise, incorporating the model into the plugin as an accessible example would not only be consistent with the established piStar design but also address the issue of presenting complex examples within the text document. Additionally, the proposal to include definitions and examples for each requirement is particularly valuable. This could significantly enhance the learning experience for beginners, making

the material more approachable. Moreover, this feature could also be integrated into the tool itself, using its existing dictionary containing the definitions of the requirements. By displaying definitions alongside examples directly within the plugin, users would have a comprehensive resource easily available reducing the potential for confusion. Finally one of the suggestions also believes there should be a section in the guide for explaining how the sustainability dimensions are combined, and more examples. While we received only four suggestions, they highlight key areas for improvement. Implementing these changes could lead to a more user-friendly experience, ultimately enhancing the effectiveness of our guides in conveying the sustainability concepts present in our work.

6.4.4.6 Video guide improvement suggestions

We also asked participants for improvement suggestions regarding our video guide, and we received four suggestions one of which was a participant saying that they had no recommendation, so we discarded it. These were the three remaining recommendations:

- "With video guides I always find it useful to separate the video into "Chapters" covering different topics, using a short title card to introduce each of them. That way when I want to look at some concept presented in the video again it is easier to locate."
- "Pass the information more slowly, mainly the last modeling to show the final model implemented. I choose the video guide with the best, because it is more comfortable to watch the video than read the guide. But both are important, mainly thinking in accessibility."
- "Just enhance the video resolution."

These suggestions indicate that while the content of the video is generally well-received, there are opportunities to enhance its accessibility. Dividing the video into chapters would not only improve user navigation but also enable users to revisit specific sections without needing to rewatch the entire video. The feedback regarding the pace of the final example also presents a valid point, the example was perhaps a bit too fast in its demonstration. The example was shown this way because videos can be slowed down or sped up. To prevent the video from becoming too long this section had its speed increased. By slowing down this segment, we can ensure that all viewers, especially those who may be less familiar with the concepts, have more time to follow along and understand the example without having to pause. Giving the users the option to speed up the video instead of slowing it down may be a more intuitive option even if the effect is similar. The suggestion on the video resolution could also be investigated, while during production there were no clear signs of lack of resolution on the video improving its resolution would ensure that the video is more easily watchable. Overall, the presence of only these three suggestions, along with the generally positive feedback, suggests that the video

guide is well-constructed and effectively delivers the necessary instructions. However, implementing these improvements could further enhance the user experience, making the video a more effective learning tool.

6.4.4.7 Preference between the video and written guide

Finally, we asked the participants, in the case of having used both guides, to choose which one they felt was the best. Here, we observed an inconsistency: in the previous questions regarding the video, we had at most 8 answers about the characteristics of the video, indicating that the maximum number of participants who used the video guide was 8. However, in this question, we obtained 9 answers. This discrepancy suggests that one of the responses may be invalid, either because the participant answered without using one of the guides, or they may have forgotten or opted not to answer the earlier questions about the video. Although this affects the certainty of our data, the impact is minimal since the vote was unanimous in favor of the video being considered the better guide. Removing one answer would not have altered the outcome. Having the participants choosing exclusively the video guide over the text one indicates that the video guide was much more effective or engaging that the text guide. The results of this question are shown in Figure 6.17.



Figure 6.17: Participant opinions on the video guide's quality

6.4.5 Open feedback

This final section consists of open-ended questions, where participants were asked to share their opinions on specific parts of the catalog.

6.4.5.1 Misplaced sustainability requirements

The first three questions in the open feedback section asked participants whether they believed any of the requirements were wrongly categorized within the dimensions, and where they would place them instead. One participant who expressed discomfort in answering due to the lack of attached definitions for all requirements. As previously

mentioned, the catalog's primary objective was not to include definitions at the outset, which limited our ability to find comprehensive descriptions within the research. A second participant claimed that collaboration is a requirement that should be in the social dimension instead of belonging to the individual dimension. This mirrors what was said in Chapter 3 when we discussed that some works believe that the social and individual dimension are very similar or even the same. Our research led us to place collaboration in the individual dimension but it's worth investigating in a future iteration of the catalog with all dimensions included. Finally in the environmental dimension's feedback one participant claimed that there is no distinct boundary between the categories. The catalog was built in such a way that the categories were clearly defined by levels, considering the clear structure of the catalog this could indicate a flaw in the guides. Alternatively it may be pointing out redundancy in the categorization as this is something that is pointed out in the feedback for another question. Both possibilites are worth investigating in future work.

6.4.5.2 Missing requirements in the environmental dimension

Like the first three questions, the next three questions asked participants if they believed any requirements were missing from each of the dimensions. This time, however, we received differing answers for each question. Regarding the environmental dimension, a single suggestion was made:

- "Environmental risk mitigation; Maintainability/reusability from ISO/IEC 25010:2011 Quality model."

Upon review, this suggestion didn't make sense, as these requirements are already present. They fall within the multi-dimension section of the catalog. This raised the possibility that the suggestion indicated an issue with readability or clarity. The multi-dimension section isn't separate from the individual dimension sections; it's simply a way to present overlapping information in a less cluttered format. This could indicate that the guide didn't adequately explain how the multi-dimension and single-dimension categories are interrelated. In future iterations, it would be useful to revise either the guide or the plugin's built-in instructions to clarify how these sections work, ensuring that this isn't simply a misunderstanding. Additionally, the possibility exists that the participant couldn't easily locate the relevant requirements, highlighting another potential area for improvement: the addition of a dedicated search functionality. Although the "locate in page" feature in most web browsers can already perform this function, an integrated search tool would make this feature more accessible, especially for users who might not be as familiar with browser-based applications.

6.4.5.3 Missing requirements in the economic dimension

Following the same pattern, we asked about potentially missing requirements in the economic dimension. This time, we received two responses:

- "Economic risk mitigation; Maintainability from ISO/IEC 25010:2011Quality model."
- "These comments are to environmental and economics dimensions: When identify the interoperability requirements it supports beyond the compatibility. I suggest looking at: - Resource Utilization Efficiency (e.g., Economic: Optimizing resource use reduces operational and infrastructure costs. Environmental: Lower resource consumption reduces energy use and hardware, contributing to sustainable operations with a smaller environmental footprint.) - Scalability (e.g., Economic: Scalability adjusts resource allocation based on actual needs, preventing costs associated with over-provisioning. Environmental: Efficient use of computational resources reduces environmental impact by limiting energy use over the system's lifecycle) - Adaptability and Reusability (e.g., Economic: Modularity and reusability of components lower development and maintenance costs. Environmental: Reduces the need for new resources, lowering energy consumption and reducing digital waste). These requirements support the achievement of the interoperability."

The first response presents the same issue as in previous questions: the suggested requirements (economic risk mitigation and maintainability) already exist in the catalog. This suggests the same clarity problem we identified earlier. The participants may not have fully understood the structure or contents of the multi-dimension section, which contains these elements, or they simply had difficulty finding them. This reiterates the need to improve the guide or the tool's accessibility features. The second response, however, introduces a different perspective by linking requirements like resource utilization efficiency, scalability, and adaptability to interoperability, arguing that interoperability impacts more than just compatibility. The participant suggests that these concepts support achieving interoperability and points out the distinct benefits for both the economic and environmental dimensions. This feedback highlights a potential gap in our catalog, which focused primarily on the established connections between requirements found in Chapter 3. While our catalog was built with possible future expansions in mind, this response confirms that further exploration of sustainability requirements aspects would enhance the catalog.

6.4.5.4 Missing requirements in the individual dimension

The last question about missing requirements focused on the individual dimension, asking participants to suggest any requirements they felt were missing. The following responses were received:

- "Health and safety risk mitigation and Satisfaction from ISO/IEC 25010:2011Quality model."
- "Ethical transparency is important to thinking in transparency"
- "I am currently studying the individual dimension and in my doctoral research, we are also looking at the issue of human values, such as motivation and justice. However, the work is still in its infancy."

The first response, as seen in previous questions, points to requirements already present in the catalog. This suggests the same misunderstanding or difficulty in locating certain requirements. Since "Health and safety risk mitigation" and "Satisfaction" are part of the catalog, it emphasizes the need to improve how these concepts are presented, possibly by enhancing the guide or providing better explanations within the tool. The second response propsed a requirement missing from the catalog: ethical transparency. The participant suggests that transparency, particularly in ethical matters, could be an important addition to the individual dimension. However, they did not elaborate on how or why this would be relevant and were overly sussinct, leaving some ambiguity about its placement or significance within the context of the catalog. Given the limited explanation provided, one interpretation could be that the participant is referring to the idea of transparency related to ethical software development. This aligns with the "Ethical requirements" category defined by Sommerville. Ethical transparency could involve ensuring that software systems operate in a clear, understandable way, allowing users to see how decisions are made, data is used, or impacts are determined. This could be especially relevant in systems where trust and accountability are critical. Lastly, one participant referenced their work on studying the individual dimension of sustainability. Since this research is still in progress, it emphasizes how sustainability continues to evolve. As a result, any catalog developed will inevitably need to be updated to reflect new insights from ongoing studies. Future work could aim to integrate these findings to ensure the catalog remains both updated and accurate.

6.4.5.5 Strong points of the catalog

In this section, participants were asked to identify the strengths of the sustainability catalog. The three responses focused on key positive aspects:

- "I think the strongest point is the ammount of requirements available."
- "It helps organize, reuse, and identify sustainability requirements"
- "Good overview"
- "It's quite thorough with the requirements it focuses on, and the structure is quite readable."

- "Easy to organize sustainability requirements"
- "The social and technical dimensions exhibit an absence of numerous categories."
- "Its existence is a guide for sustainability researchers."

One participant highlighted the amount of sustainability requirements as a strength, suggesting that the catalog offers extensive coverage, which can be helpful for users looking to comprehensively address sustainability in software projects. A few participants emphasized how the catalog helps organizing, reusing, and identifying sustainability requirements. This suggests that the catalog provides a clear structure, making it easier for developers to navigate and apply these principles in their projects. Another participant appreciated the catalog's ability to give a broad overview of sustainability requirements. The feedback confirms that the catalog achieves its goal of offering a well-organized, comprehensive resource for developers aiming to integrate sustainability into software development. One comment noted the absence of categories for the social and technical dimensions, which was presented as a strong point but could be more accurately described as a limitation. These dimensions were not included in the current catalog because they fell outside the scope of our thesis' theme. However, we have developed support to incorporate these dimensions in future work. The strong points presented coincide with our objectives: to help users identify and reuse sustainability requirements, effectively adding value to our work.

6.4.5.6 Strong points of the plugin

In this section, participants were asked to identify the strong points of the plugin, and the feedback highlighted several positive aspects:

- "It uses familiar geometric / non geometric forms that other plugins use and it seems to follow the KISS principle"
- "From the video, it looks very snappy to use (when generating a diagram for example)"
- "Easy to use."
- "The generated diagram"
- "The usability"
- "Better visualization of dimensions"

One participant noted that the plugin uses familiar geometric and non-geometric forms that are common in other tools, suggesting that users would not need to learn new design paradigms, making the transition to this tool easier. They also mentioned that the plugin follows the KISS (Keep It Simple, Stupid) principle, which is a hallmark of

good software design, indicating that the plugin is easy to understand and not overly complex. Another participant mentioned that the plugin appears "snappy" in terms of responsiveness, especially when generating diagrams. This highlights that the plugin performs well and efficiently, which is critical to reduce frustration during use. We obtained an answer emphasizing that the plugin is easy to use, reaffirming that the tool successfully meets one of its primary goals: ensuring a smooth and accessible user experience. An answer pointed out the generated diagram as a strong point, which suggests satisfaction with the visual output and the quality of the diagrams produced by the plugin. These responses reflect the success of the plugin's design in several key areas: ease of use, simplicity, and performance. The positive feedback on the plugin's responsiveness and the quality of its diagrams indicates that users appreciate both the functional efficiency and the results.

6.4.5.7 Weak points of the catalog

In this section, participants were asked to identify the weak points of the catalog, and their responses highlighted three key weakpoints:

- "Requirements could be sorted in some non-arbitrary way, that way if more requirements get added there could be an easy method to finding one. (if they were, lets say, sorted alphabetically)"
- "I would like to better understand the purpose of the level that presents requirements (quality) according to Sommerville's classification, as outlined in the guide. For example, it seems redundant to have both 'usability requirements' and 'usability' as separate qualities, even though they are at different levels. Some sustainability requirements definitions are missing is other "weak" point."
- "Don't allow to zoom out"
- "The separation of single-dimension and multi-dimension requirements, and repetition of certain concepts between them, wasn't very clear to me."
- "Additional catalogues are required. Is there an option to make modifications and incorporate more catalogues?"
- "Need more examples and details"

One participant suggested that the requirements could be sorted in a more systematic and non-arbitrary way, such as alphabetically. This point is particularly useful because, as the catalog expands, users may find it increasingly difficult to locate specific requirements. Implementing a clear, logical sorting method would greatly enhance usability, especially as more requirements are added. This could be considered in future iterations of the catalog to improve navigability and user experience. Another participant questioned

the utility of having requirements and qualities separated according to Sommerville's classification, particularly when there appears to be redundancy (e.g., having both "usability requirements" and "usability" as distinct levels). It may be necessary to either clarify this distinction or rethink the organization to avoid confusion. Additionally, the participant mentioned the absence of definitions for certain sustainability requirements, which was acknowledged as a limitation in earlier sections due to the lack of definitions in the research material. This is another area for future research and improvement. The next point concerns the lack of zoom functionality in the piStar tool, which affected the overall usability of the catalog. This issue lies with the piStar tool itself, users can resort to using the browser's zoom feature, which can result in awkward visuals. This is a legitimate criticism and could be a feature request for the piStar developers. Extending the catalog to address this limitation would significantly improve user interaction with larger models. These suggestions point to meaningful improvements that could greatly improve the catalog in future iterations. Two criticisms raised in previous responses were that the distinction between single-dimension and multi-dimension requirements was unclear, and that more examples and details were needed. Both issues can be addressed by enhancing the guidance materials. Specifically, the guides can provide a clearer explanation of the difference between single-dimension and multi-dimension requirements, and include more examples. The examples could be in the guides themselves or within the plugin. Finally, one user mentioned the need for more catalogs in the future. This concern has already been addressed, as the catalog in our plugin is designed to be easily updated. Future work can incorporate additional catalogs into the tool as needed.

6.4.5.8 Weak points of the developed tool

The final question of the questionnaire asked participants what they thought the weak points of our piStar plugin were, and we received two answers:

- "There is an assumption similar to the SusAF, that prompting developers themselves to ask questions about hypothetical effects would have effects on design decisions taken. Those deciding on design might not be willing to accept that a system should not be built, or even that an existing system might have to be scrapped, without a social context that allows for questions of a much deeper character to be allowed. Such a context is much more important, and promoting it by inviting people who experience that which calls for change."
- "Scalability issues, i.e. it will be nice to have an expand and collapse functionality for the dimensions, and qualities. Suggestions to improve usability: for each dimension tab in the window, add two buttons: one to select all requirements and the other to deselect all."
- "There should be a way to go from the relationship/refinement lists, back to the menu, instead of having to close it and open it again."

- "If symbols indicate the contribution, such as (++) and (+) located in the quality shapes corner."
- "Need more details"

The first criticism presents an interesting discussion. The participant claims that merely prompting developers to consider hypothetical effects may not influence their design decisions. They argue that decision-makers might resist abandoning or scrapping systems without a social context that encourages change on a deeper level. While this perspective is valid, it overlooks the fundamental purpose of creating our tool and plugin. Our goal was to provide developers, regardless of their stage in the development process, with a starting point to consider and potentially implement sustainability in their software projects. We did not expect that developers would abandon projects simply because they lacked sustainability features; instead, the tool is designed to allow users to pick and implement only the requirements they feel are feasible. Change, especially in societal contexts, often starts small. By encouraging individual developers to contemplate sustainability, we hope to encourage these considerations to be integrated into software development practices. The second criticism regarding usability is helpful. The participant's suggestion for buttons to select and deselect all requirements, along with expand/collapse functionality, speaks directly to enhancing user experience. Although the collapse and expand features exist in the plugin, it's possible that it was not explained well enough or the instructions were not obvious enough. To address this, we could improve the user guides or change the instructions regarding this functionality to become more obvious, these usability improvements align with our objective of making the tool as accessible as possible. A usability issue that was found was that there should be an easier way to navigate the lists present in the plugin, this could be corrected in a future iteration. One possibility would be to have the navigation buttons present in all windows related to the plugin, instead of just the main one. Another usability improvement was to indicate the contributions through symbols. Our work was somewhat limited to what piStar was able to do. Due to the diagram validation process that it has in place it would be very difficult to change the symbols of the contribution, and due to this difficulty some changes were not possible. A future iteration of the plugin could attempt to implement its own version of the model validation process to override piStar's. By doing so it would enable changes such as what the participant suggested. Finally one user reiterated that the plugin should have more details. PiStar has the ability to add descriptions to most of its elements, in a future version of the plugin this description field could be used to introduce more details into the plugin and catalog.

6.5 Threats to validity

With the presentation of our evaluation concluded, we will now discuss the threats to validity. We adhered to the guidelines outlined in [66] to evaluate four key aspects: internal

validity, external validity, construct validity, and conclusion validity.

6.5.1 Internal validity

To ensure our internal validity as much as possible, we aimed to make the guides for our work as accessible as possible. This was the primary reason for providing two guides during the evaluation process. Should a participant find any aspect of one guide difficult to understand or overwhelming, they could switch to the other guide for a different explanation. Most concepts are elaborated in the guides, the help section of the piStar tool, or the brief explanations embedded in the plugin, ensuring that participants have the necessary information to make informed evaluations. While the feedback regarding the guides was largely positive, the open-ended questions revealed that some users misunderstood either the intention of the tool or certain concepts within our work. This indicates a potential threat to our evaluation. Another measure we took to enhance internal validity was to keep the guides and questionnaire succinct. We aimed to limit their length to prevent participants from experiencing fatigue during the evaluation. Most questions had the format of multiple-choice or Likert scales, and open-ended questions were made optional. The questionnaire was designed to take up to 20 minutes, assuming users would copy parts of the examples from the guides into piStar. Considering the responses provided in the open-ended questions, we believe that this potential threat to internal validity was mitigated.

6.5.2 External validity

Despite a considerable number of invitations sent out, the participant count was low, which poses a risk to our external validity as it limits the ability to generalize based on our findings. However, the participants represented a diverse range of backgrounds: some were authors from our bibliography with expertise in software engineering or sustainability, while others were current practitioners in the software engineering industry. Additionally, some participants were academics, a few of whom had industry experience, providing insights from varying levels of expertise. The questionnaire and accompanying materials were built in English to include participants who might not be fluent in Portuguese, such as the authors cited in our bibliography. While this approach aimed to include a wider audience, it introduced potential challenges; some participants may have struggled with English, as evidenced by the brevity of some responses. However, this could also reflect their level of familiarity with sustainability concepts, rather than language proficiency alone. Despite these challenges, we believe that using English helped our external validity. We also aimed for a diverse range of experiences in software development among our participants, there was a noticeable skew toward younger age groups, indicating that our sample may not fully represent all levels of experience. Since our work is encompasses both requirements engineering and sustainability, we invited potential participants engaged in sustainability research and participants that are specialized in the area of computer

science. We obtained at least one answer from a specialist in each area mitigating the threat of not having specialized feedback.

6.5.3 Construct validity

A potential threat to our evaluation is related to the construction of the questions themselves. If the questions are poorly phrased or fail to accurately capture the intended concepts, this could be a threat to the validity of our findings. Additionally, if a question does not align with the goals of our research, it could further threaten construct validity. To mitigate these risks, we focused on crafting questions that are straightforward, and easy to understand. We organized the questions into specific sections, ensuring that each question's intent was clear and directly related to the objectives of our work. We believe that the threats related to the construct validity have been mitigated.

6.5.4 Conclusion validity

Our evaluation indicated that the tool and catalog developed during our work were effective in helping participants understand how to better implement sustainability in their systems. To ensure the validity of our conclusions, we designed a focused set of questions that directly addressed the primary objectives of our research. We separated the questions into distinct categories for the catalog and the plugin, which helped eliminate ambiguity regarding what was being evaluated. By employing an appropriate number of well-constructed questions, we minimized the risk of drawing incorrect conclusions based on misleading data. However, it is important to acknowledge that the very limited number of participants presents a potential threat to our conclusion validity, as we are inferring results from a small sample size.

6.6 Summary

In this chapter, we presented the results of the evaluation of our catalog of sustainability requirements, which encompasses environmental, economic, and individual dimensions. We began by outlining the guides and the questionnaire developed for the evaluation. Our objective was to assess whether our tool achieved its goals of usefulness and accessibility for users with varied experience levels in computer science and sustainability. After explaining the structure of both the guide and the questionnaire, we discussed the largely positive evaluation results. Most participants expressed interest in using the tool and catalog again in the future. Additionally, we received constructive feedback that provided valuable insights for potential improvements to both the catalog and the tool. Although the number of responses was limited, the feedback we received offered meaningful suggestions for enhancing our work in the future.

7

Conclusion and Future work

As the global population continues to grow and resources become more scarce, sustainability has become a critical priority across all scientific fields. Failing to adopt sustainable practices risks depleting our resources and causing irreversible damage to the only planet we can inhabit. In the realm of computer science and software development, there is an ongoing struggle to effectively implement sustainability within the industry. As software increases its effects on various aspects of our lives, addressing this issue becomes even more urgent. Throughout this dissertation, we investigated sustainability in software engineering and explored existing tools and catalogs currently available to tackle the challenges associated with implementing sustainability. Our findings revealed significant gaps in the available resources, encouraging us to propose a catalog of reusable sustainability requirements that belong to the environmental, economic, and individual dimensions, along with a supporting tool. In this final chapter we will present the contributions of the work developed in our dissertation. We will go over what we aimed to accomplish and what we believe to have been our achievements. We will finish by giving suggestions for future work.

7.1 Contributions

Throughout the course of our research, we made significant strides toward achieving our objectives. Our work addressed challenges in the field of sustainable software, offering solutions that enhance the applicability of sustainability in software. The contributions of our work were:

- A systematic mapping study that identified various sustainability requirements across the three dimensions, as well as the relationships between these requirements. The results highlighted two significant issues: a scarcity of easy-to-use catalogs and tools for eliciting sustainable software requirements, and a lack of resources that address the interconnections between sustainability dimensions and their associated requirements.

- A catalog that covers the environmental, economic and individual dimensions of sustainability based on the requirements found in our systematic mapping study.
- A supporting tool that uses our catalog to help developers implement sustainability in their software projects.
- An evaluation of our work.

These contributions have strengthened the overall impact of our findings and provided a solid basis for continued exploration in future research.

7.2 Future work

In this dissertation, we successfully developed a catalog and tool that serve as a solid foundation for future research and enhancements. Some functionalities were intentionally implemented with this goal in mind, such as the dictionary, which currently lacks definitions for all requirements due to the absence of clear definitions in the selected works. Based on our implementation and the feedback received during the evaluation, we propose the following improvements and opportunities for future work:

- **Completing the Catalog**: The catalog currently covers only three dimensions of sustainability and does not list any negative relationships between requirements. While this approach aligns with our systematic mapping study and dissertation theme, future research should aim to expand the catalog by incorporating additional dimensions and reviewing the relationships to ensure their accuracy.
- **Completing the Dictionary**: The dictionary could benefit from additional definitions. Our current limitations were due to the definitions available in the studies and sustainability standards. A comprehensive dictionary would enhance the tool's accessibility, as confirmed by participant suggestions during the evaluation.
- **Evaluate the work with more participants**: The number of participants was considerably low compared to the number of invites sent, resulting in a limited response pool. Future evaluations should involve more participants to gather more diverse feedback and suggestions regarding the catalog and tool.
- **Improving User Guides**: The guides available to users could be enhanced. As noted in the evaluation, the text guide could include definitions to clarify the requirements, and the examples could be made more readable. Alternatively, as suggested in Chapter 6, incorporating examples directly into the plugin, similar to how piStar loads examples in its help tab, could be beneficial. Additionally, improving the video guide by adding chapters and slowing down the last example would enhance usability.

- **Expanding Catalog Requirements**: While most participants expressed satisfaction with the existing requirements, some questioned the absence of additional requirements. Given the broad scope of sustainability, it is natural to expect more inclusivity, and future work should aim to increase the number of available requirements.
- **Integrating ongoing and future research into the catalog**: As mentioned by one of our evaluation participants, sustainability still has ongoing research. This means that the catalog that was developed by this work will inevitably become outdated and innacturate. Future work could update the catalog to integrate new relevant research results into the catalog keeping it up to date and accurate.
- **Reviewing requirement placements**: While not numerous, we obtained feedback claiming that some of the requirements should have been placed in other dimensions and that there was no distinct boundary between the categories in the environmental dimension. A new version of the catalog with the assistance sustainability experts could address the criticism and create a more accurate version of the catalog.
- **Enhancing Usability Features**: Implementing sorting functions, a search functionality, and zoom in/out features—as suggested in Chapter 6—could significantly improve the accessibility of the catalog.
- **Quality of Life Improvements in the Plugin**: Finally, the plugin could incorporate quality of life enhancements, such as the suggested "select all" and "deselect all" buttons from our evaluation.

Bibliography

[1] João M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] Ana Moreira, João Araújo, Catarina Gralha, Miguel Goulão, Isabel Sofia Brito, and Diogo Albuquerque. "A social and technical sustainability requirements catalogue". In: *Data & Knowledge Engineering* 143 (2023), p. 102107 (cit. on pp. 1, 3, 9, 19, 21, 28).

[3] Sara S Mahmoud and Imtiaz Ahmad. "A green model for sustainable software engineering". In: *International Journal of Software Engineering and Its Applications* 7.4 (2013), pp. 55–74 (cit. on pp. 2, 19, 21).

[4] A van Lamsweerde. *Requirements engineering: from system goals to UML models to software specifications*. John Wiley & Sons, Ltd, 2009 (cit. on pp. 5, 6).

[5] Gerald Kotonya and Ian Sommerville. *Requirements engineering: processes and techniques*. Wiley Publishing, 1998 (cit. on pp. 5, 6).

[6] Saurabh Tiwari, Santosh Singh Rathore, and Atul Gupta. "Selecting requirement elicitation techniques for software projects". In: *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*. IEEE. 2012, pp. 1–10 (cit. on p. 7).

[7] Fabiano Dalpiaz, Xavier Franch, and Jennifer Horkoff. "istar 2.0 language guide". In: *arXiv preprint arXiv:1605.07767* (2016) (cit. on pp. 8, 32, 33).

[8] Gro Harlem Brundtland. "Our common future—Call for action". In: *Environmental conservation* 14.4 (1987), pp. 291–294 (cit. on p. 9).

[9] Ans Kolk. "Sustainability reporting". In: *VBA journal* 21.3 (2005), pp. 34–42 (cit. on p. 9).

[10] Hans Gersbach, Noemi Hummel, and Ralph Winkler. *Sustainable climate treaties*. Tech. rep. 2017 (cit. on p. 9).

[11] *The United Nations Sustainable Development Goals*. https://sdgs.un.org/goals. Last visited: September 2024 (cit. on p. 9).

[12] Birgit Penzenstadler and Henning Femmer. "A generic model for sustainability with process-and product-specific instances". In: *Proceedings of the 2013 workshop on Green in/by software engineering*. 2013, pp. 3–8 (cit. on p. 9).

[13] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M Easterbrook, Birgit Penzenstadler, Norbet Seyff, and Colin C Venters. "Requirements: The key to sustainability". In: *IEEE Software* 33.1 (2015), pp. 56–65 (cit. on pp. 9, 37, 38).

[14] Colin C Venters, Norbert Seyff, Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Dan McIntyre, and Birgit Penzenstadler. "Characterising sustainability requirements: A new species red herring or just an odd fish?" In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*. IEEE. 2017, pp. 3–12 (cit. on p. 10).

[15] Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff, and Colin C Venters. "Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow". In: *Requirements Engineering* 25 (2020), pp. 469–492 (cit. on pp. 10, 19, 21, 29).

[16] Barbara Kitchenham and Stuart M. Charters. *Guidelines for performing systematic literature reviews in software engineering*. 2007 (cit. on p. 11).

[17] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. "Systematic mapping studies in software engineering". In: *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*. 2008, pp. 1–10 (cit. on pp. 11, 12).

[18] Mark Petticrew and Helen Roberts. *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008 (cit. on p. 11).

[19] Barbara Kitchenham. "Procedures for performing systematic reviews". In: *Keele, UK, Keele University* 33.2004 (2004), pp. 1–26 (cit. on p. 17).

[20] Computing Research and Education Association of Australasia. *CORE Rankings Portal*. https://www.core.edu.au/conference-portal. Last visited: January 2024 (cit. on p. 17).

[21] Nelly Condori-Fernandez and Patricia Lago. "Characterizing the contribution of quality requirements to software sustainability". In: *Journal of systems and software* 137 (2018), pp. 289–305 (cit. on pp. 19, 21, 22, 24, 27, 29, 30, 53).

[22] Leticia Duboc, Stefanie Betz, Birgit Penzenstadler, Sedef Akinli Kocak, Ruzanna Chitchyan, Ola Leifler, Jari Porras, Norbert Seyff, and Colin C Venters. "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)*. IEEE. 2019, pp. 6–16 (cit. on pp. 19, 21, 22, 24, 26).

[23] Nelly Condori-Fernandez and Patricia Lago. "Towards a software sustainability-quality model: Insights from a multi-case study". In: *2019 13th international conference on research challenges in information science (rcis)*. IEEE. 2019, pp. 1–11 (cit. on pp. 19, 21, 24).

[24] Sedef Akinli Koçak, Gülfem Isiklar Alptekin, and Ayse Basar Bener. "Integrating Environmental Sustainability in Software Product Quality." In: *RE4SuSy@ RE*. 2015, pp. 17–24 (cit. on pp. 19, 21, 28).

[25] Shola Oyedeji, Ahmed Seffah, and Birgit Penzenstadler. "Sustainability quantification in requirements informing design". In: *6th Int. Work. Requir. Eng. Sustain. Syst* 1 (2017) (cit. on pp. 19, 21).

[26] Ruzanna Chitchyan, Iris Groher, and Joost Noppen. "Uncovering sustainability concerns in software product lines". In: *Journal of Software: Evolution and Process* 29.2 (2017), e1853 (cit. on pp. 19, 21, 22, 24, 52).

[27] Patricia Lago, Roberto Verdecchia, Nelly Condori-Fernandez, Eko Rahmadian, Janina Sturm, Thijmen van Nijnanten, Rex Bosma, Christophe Debuysscher, and Paulo Ricardo. "Designing for sustainability: lessons learned from four industrial projects". In: *Advances and New Trends in Environmental Informatics: Digital Twins for Sustainability*. Springer. 2021, pp. 3–18 (cit. on pp. 19, 21, 22, 24, 27, 29, 52, 53).

[28] Yen Dieu Pham, Abir Bouraffa, and Walid Maalej. "Shapere: Towards a multi-dimensional representation for requirements of sustainable software". In: *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE. 2020, pp. 358–363 (cit. on pp. 19, 21, 22, 24, 26, 53).

[29] Gabriel Alberto García-Mireles, Mª Ángeles Moraga, Félix García, Coral Calero, and Mario Piattini. "Interactions between environmental sustainability goals and software product quality: A mapping study". In: *Information and Software Technology* 95 (2018), pp. 108–129 (cit. on pp. 19, 28).

[30] Shola Oyedeji, Ahmed Seffah, and Birgit Penzenstadler. "A catalogue supporting software sustainability design". In: *Sustainability* 10.7 (2018), p. 2296 (cit. on pp. 19, 21, 22, 24).

[31] Amin Khalifeh, Peter Farrell, Mohammad Alrousan, Shaima Alwardat, and Masar Faisal. "Incorporating sustainability into software projects: a conceptual framework". In: *International Journal of Managing Projects in Business* 13.6 (2020), pp. 1339–1361 (cit. on pp. 19, 21, 22, 24).

[32] Ahmed D Alharthi, Maria Spichkova, and Margaret Hamilton. "Sustainability requirements for eLearning systems: a systematic literature review and analysis". In: *Requirements Engineering* 24 (2019), pp. 523–543 (cit. on pp. 19, 24, 27, 29, 53).

[33] Peter Bambazek, Iris Groher, and Norbert Seyff. "Requirements engineering for sustainable software systems: a systematic mapping study". In: *Requirements Engineering* (2023), pp. 1–25 (cit. on pp. 19, 21).

[34] Stefan Naumann, Eva Kern, Markus Dick, and Timo Johann. "Sustainable software engineering: Process and quality models, life cycle, and social aspects". In: *ICT Innovations for Sustainability*. Springer. 2015, pp. 191–205 (cit. on pp. 19, 21).

[35] Ahmad F Subahi. "BERT-Based Approach for Greening Software Requirements Engineering Through Non-Functional Requirements". In: *IEEE Access* (2023) (cit. on pp. 20, 21, 24).

[36] Kamal Uddin Sarker, Aziz Bin Deraman, Raza Hasan, and Ali Abbas. "SQ-framework for improving sustainability and quality into software product and process". In: *International Journal of Advanced Computer Science and Applications* 11.9 (2020) (cit. on pp. 20, 21).

[37] Ruzita Ahmad, Fauziah Baharom, and Azham Hussain. "GOSSEC: Goal Oriented Software Sustainability Evaluation Criteria". In: *Journal of University of Babylon for Pure and Applied Sciences* 27.1 (2019), pp. 387–407 (cit. on pp. 20–22, 24, 53).

[38] Sofia Ouhbi, José Luis Fernández-Alemán, Ambrosio Toval, Jose Rivera Pozo, and Ali Idri. "Sustainability requirements for connected health applications". In: *Journal of Software: Evolution and Process* 30.7 (2018), e1922 (cit. on pp. 20, 21).

[39] Birgit Penzenstadler, Ankita Raturi, Debra Richardson, Coral Calero, Henning Femmer, and Xavier Franch. "Systematic mapping study on software engineering for sustainability (SE4S)". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 2014, pp. 1–14 (cit. on p. 20).

[40] Barbara Paech, Ana Moreira, João Araujo, and Peter Kaiser. "Towards a Systematic Process for the Elicitation of Sustainability Requirements." In: *RE4SuSy@ RE*. 2019, p. 6 (cit. on pp. 20–22, 26).

[41] Shola Oyedeji and Birgit Penzenstadler. "Karlskrona Manifesto: Software Requirement Engineering Good Practices." In: *RE4SuSy@ RE*. 2018, pp. 15–23 (cit. on p. 20).

[42] Bendra Ojameruaye, Rami Bahsoon, and Leticia Duboc. "Sustainability debt: A portfolio-based approach for evaluating sustainability requirements in architectures". In: *Proceedings of the 38th International Conference on Software Engineering Companion*. 2016, pp. 543–552 (cit. on pp. 20–22, 24, 26).

[43] Isabel Sofia Brito, José M Conejero, Ana Moreira, and Joao Araújo. "A concern-oriented sustainability approach". In: *2018 12th international conference on research challenges in information science (RCIS)*. IEEE. 2018, pp. 1–12 (cit. on pp. 20, 22, 24).

[44] Shola Oyedeji, Ahmed Seffah, and Birgit Penzenstadler. "Classifying the measures of software sustainability". In: *Proceedings of the 4th International Workshop on Measurement and Metrics for Green and Sustainable Software Systems co-located with 12th International Symposium on Empirical Software Engineering and Measurement (ESEM 2018)*. 2018 (cit. on pp. 20, 21, 24–26).

[45] Timo Kehrer and Birgit Penzenstadler. "An Exploration of Sustainability Thinking in Research Software Engineering." In: *RE4SuSy@ RE*. 2018, pp. 34–43 (cit. on pp. 20, 22, 26).

[46] Coral Calero, Mª Ángeles Moraga, and Félix García. "Software, Sustainability, and UN Sustainable Development Goals". In: *IT professional* 24.1 (2022), pp. 41–48 (cit. on p. 20).

[47] Eva Kern, Markus Dick, Stefan Naumann, and Andreas Filler. "Labelling sustainable software products and websites: ideas, approaches, and challenges". In: *EnviroInfo and ICT for Sustainability 2015* (2015), pp. 82–91 (cit. on p. 20).

[48] Alka Agrawal, Mamdouh Alenezi, Rajeev Kumar, and Raees Ahmad Khan. "A unified fuzzy-based symmetrical multi-criteria decision-making method for evaluating sustainable-security of web applications". In: *Symmetry* 12.3 (2020), p. 448 (cit. on pp. 20, 21).

[49] Najia Saher, Fauziah Baharom, and Rohaida Romli. "Identification of sustainability characteristics and subcharacteristics as non-functional requirement for requirement change management in agile". In: *International journal of Scientific & Technology Research* 9.3 (2020), pp. 5727–5733 (cit. on pp. 20–22, 24).

[50] Ahmed D Alharthi and Maria Spichkova. "Green architecture for sustainable elearning systems". In: *Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings*. 2017, pp. 199–204 (cit. on pp. 20, 22, 26).

[51] Sulaiman Aljarallah and Russell Lock. "A comparison of software quality characteristics and software sustainability characteristics". In: *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*. 2019, pp. 1–11 (cit. on p. 20).

[52] Shallu Madan, Sukhdip Singh Sangwan, and Rajesh Bhatia. "Identification, selection and prioritization of sustainable and green software factors using fuzzy set theory". In: *International Journal of Computational Intelligence & IoT* 2.1 (2019) (cit. on pp. 20, 24, 25).

[53] Azza Alajlan and Malak Baslyman. "Toward a Comprehensive Understanding and Evaluation of the Sustainability of E-Health Solutions". In: *Applied Sciences* 13.9 (2023), p. 5811 (cit. on pp. 20, 22).

[54] Mariam Bachiri, Ali Idri, Leanne M Redman, Jose Luis Fernandez-Aleman, and Ambrosio Toval. "A requirements catalog of mobile personal health records for prenatal care". In: *Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part IV 19*. Springer. 2019, pp. 483–495 (cit. on p. 20).

[55] Vitor Maia, Taisa G Gonçalves, and Ana Regina C da Rocha. "Quality Characteristics of Mobile Applications: A Survey in Brazilian Context". In: *Proceedings of the XVIII Brazilian Symposium on Software Quality*. 2019, pp. 109–118 (cit. on p. 20).

[56] José A García-Berná, Sofia Ouhbi, and José L Fernández-Alemán. "Assessing Software Sustainability of Connected Health Applications". In: *World Conference on Information Systems and Technologies*. Springer. 2022, pp. 498–509 (cit. on p. 21).

[57] Joao Ferreira, Gustavo Callou, Albert Josua, and Paulo Maciel. "Estimating the environmental impact of data centers". In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2018, pp. 1–4 (cit. on p. 26).

[58] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C Venters. "Sustainability design and software: The karlskrona manifesto". In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 2. IEEE. 2015, pp. 467–476 (cit. on p. 29).

[59] Ian Sommerville. "Software engineering (ed.)" In: *America: Pearson Education Inc* (2011) (cit. on pp. 34, 35).

[60] 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*. ISO/IEC, 2011 (cit. on pp. 34, 36, 52–54).

[61] Joao Pimentel and Jaelson Castro. "pistar tool–a pluggable online tool for goal modeling". In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE. 2018, pp. 498–499 (cit. on p. 43).

[62] Robert B Grady. *Practical software metrics for project management and process improvement*. Prentice-Hall, Inc., 1992 (cit. on p. 53).

[63] Ibrahim Abunadi and Mamdouh Alenezi. "An empirical investigation of security vulnerabilities within web applications." In: *J. Univers. Comput. Sci.* 22.4 (2016), pp. 537–551 (cit. on p. 53).

[64] William Stallings and Lawrie Brown. *Computer security: principles and practice*. Pearson, 2015 (cit. on p. 53).

[65] *Google Forms: Free Online Surveys for Personal Use.* https://www.google.com/forms/about/. Last visited: September 2024 (cit. on p. 68).

[66] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Vol. 236. Springer, 2012 (cit. on p. 90).

A

Data extraction form

Data extraction	
Paper information	
Paper title:	
Paper URL:	
Date published:	
Authors:	
Goal of the study:	
Quality assessment	
Number of citations:	
Digital library:	
Core Rank:	
Data extraction:	
Tools developed or mentioned:	
Quality attributes discussed:	
Refinements listed:	
Sustainability requirements mentioned:	
Listed interactions between quality attributes and sustainability:	

Figure A.1: Data extraction form of the systematic mapping study

B

Full sustainability catalog

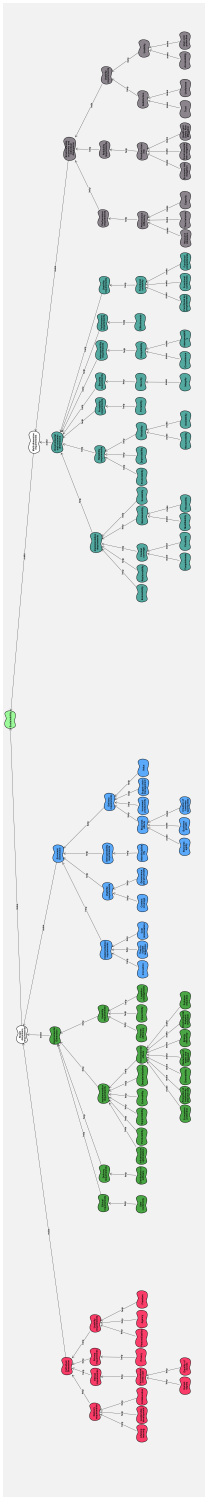


Figure B.1: Complete sustainability catalog

C

ViaVerde example

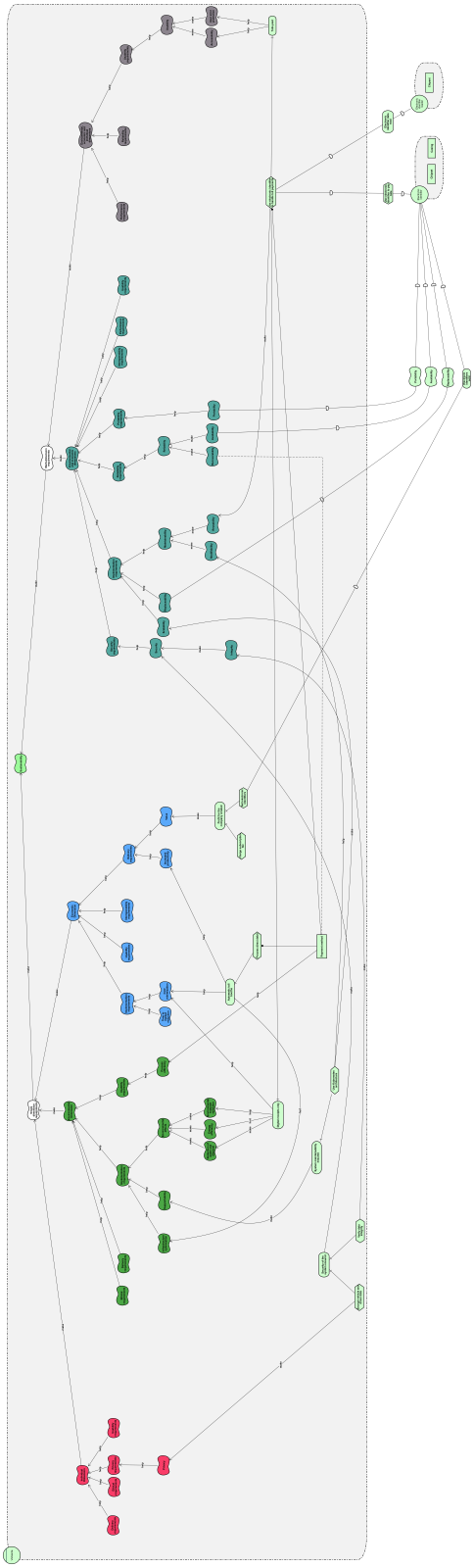


Figure C.1: ViaVerde example

D

Evaluation Questionnaire

A Sustainability Software Requirements Approach: The Economic, Environmental and Individual Dimensions Questionnaire

The purpose of this study is to evaluate a plugin for PiStar and a catalog of sustainability requirements. This work was developed in the context of my MSc dissertation in Computer Science and Informatics at the NOVA School of Technology.

The catalog is modelled in iStar 2.0, and the plugin is javascript running on the original Pistar tool, it currently covers the environmental, economic and individual dimensions.

The catalog and plugin were designed to be easily updated in the future, allowing for improvements based on changes to the software sustainability landscape and user feedback.

This evaluation form is composed of 26 questions and should take 10 to 20 minutes to answer.

The answers are anonymous and any personal information will be kept confidential. Any answers that are used directly in the dissertation will be anonymised.

Should you be interested in this dissertation, feel free to contact me.

We are thankful your feedback, as it will help improve both the usability and the completeness of our catalog and tool.

You can access the tool at:

https://talmoliveira.github.io/pistar_sustainability_plugin/#

We have a textual guide at:

https://docs.google.com/document/d/1uc3GzHU2dRxK9IYcSAuf0q103EqDIrabYnkVVFUBkNQ/

We also have a video guide at (if you already know how to use pistar you can skip to 1:38 in the video):

https://drive.google.com/file/d/19-P_KE26T2Vw1K6dkNJUKBqlKIQqeSWZ/view?usp=sharing

Should you have any doubts or want to contact me you can do so by sending an email to: talm.oliveira@campus.fct.unl.pt

Finally, if you are not confortable sharing your answers with us you are free to discard your answer at any time.

Thank you for your time and I hope you enjoy using this tool.

Sincerely Tiago Oliveira.

* Indica uma pergunta obrigatória

Personal data

This section contains questions about the personal data of the participants: age, academic training, degree, and experience (in years) of software development. These questions will help for us understand the heterogeneity of the participants.

1. Please select your age range in years. *

Marcar apenas uma oval.

- ◯ 23-35
- ◯ 36-45
- ◯ 46-55
- ◯ 56-65
- ◯ 65+

2. What is your highest level of academic education? *

Marcar apenas uma oval.

- ◯ High School Diploma/GED
- ◯ Bachelor's Degree
- ◯ Master's Degree
- ◯ Doctorate/Ph.D.
- ◯ Outra: _____

3. If you have a degree related to software development or sustainability, please specify it:

_____

4. How many years of experience do you have in software development? *

Marcar apenas uma oval.

- Less than 1 year
- 1-2 years
- 3-5 years
- 6-10 years
- More than 10 years

Quality of the Catalog

This section contains questions about the catalog.
To see the full catalog, please access the tool at https://talmoliveira.github.io/pistar_sustainability_plugin/#, and in the sustainability plugin's menu press "Generate diagram" without having any requirements selected

5. How would you rate the readability of the catalog? *

Marcar apenas uma oval.

	1	2	3	4	5	
Very	◯	◯	◯	◯	◯	Very good

6. How would you rate the utility of the catalog for understanding potentially relevant sustainability requirements in software projects? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very useful

7. The concepts in the catalog are clear. *

Marcar apenas uma oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

8. What is the likelihood that you would use the catalog as a basis for implementing sustainability in future software projects? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very likely

9. The catalog provides a better understanding of sustainability. *

Marcar apenas uma oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

Quality of the tool

These are regarding the tool (plugin)

You may access the tool at: https://talmoliveira.github.io/pistar_sustainability_plugin/#

10. The plugin is easy to use. *

Marcar apenas uma oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

11. What is the likelihood that you would use the plugin as a tool to help implement sustainability in future software projects? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very likely

12. The plugin makes it easier to implement sustainability *

Marcar apenas uma oval.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

Guide questions

This section contains a set of questions about the guide. The answers to these questions allow us to know if both the explanations and illustrations are readable and understandable.
Text guide:

https://docs.google.com/document/d/1uc3GzHU2dRxK9IYcSAuf0q103EqDIrabYnkVVFUBkNQ/

Video guide:

https://drive.google.com/file/d/19-P_KE26T2Vw1K6dkNJUKBqlKIQqeSWZ/view?usp=sharing

13. How clear were the explanations provided in the textual guide? (Ignore if you did not use it)

Marcar apenas uma oval.

	1	2	3	4	5	
Very	◯	◯	◯	◯	◯	Very clear

14. How clear were the explanations provided in the video guide? (Ignore if you did not use it)

Marcar apenas uma oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very clear

15. How would you rate the overall quality of the text guide? (Ignore if you did not use it)

Marcar apenas uma oval.

	1	2	3	4	5	
Very	◯	◯	◯	◯	◯	Excellent

16. How would you rate the overall quality of the video guide? (Ignore if you did not use it)

Marcar apenas uma oval.

1	2	3	4	5
◯	◯	◯	◯	◯

17. Do you have any suggestions for improving the explanations or illustrations in the written guide? (Ignore if you did not use it)

18. Do you have any suggestions for improving the explanations or illustrations in the video guide? (Ignore if you did not use it)

19. If you used both guides, which one would you say is the best?

Marcar apenas uma oval.

- Video guide
- Written guide

Open feedback

These final questions allow you to share your thoughts about our work more openly. You may access the tool at: https://talmoliveira.github.io/pistar_sustainability_plugin/#. As mentioned before to see the full catalog you may open the plugin's menu and press "Generate diagram" without having any requirements selected

20. If there are any qualities that you feel are wrongly categorized in the individual dimension please list them here and tell us where you would place them.

21. If there are any qualities that you feel are wrongly categorized in the environmental dimension please list them here and tell us where you would place them.

22. If there are any qualities that you feel are wrongly categorized in the economic dimension please list them here and tell us where you would place them.

23. If there any qualities related to the environmental sustainability dimension that you feel are important and missing from the catalog please list them here.

24. If there any qualities related to the economic sustainability dimension that you feel are important and missing from the catalog please list them here.

25. If there any qualities related to the individual sustainability dimension that you feel are important and missing from the catalog please list them here.

26. What do you think are the strong points of the catalog?

27. What do you think are the strong points of the plugin?

28. What do you think are the weak points of the catalog?

29. What do you think are the weak points of the plugin?

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

