Sustainability Requirements: a Systematic Mapping Study

Amanda Alves Guimarães[1]

[1]Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brazil

aag@cin.ufpe.br

Abstract. *Sustainability is a topic gaining more notoriety in recent years. Many studies related to this topic have been published in various areas of knowledge, including the field of Requirements Engineering. This work aims to investigate the literature on sustainability requirements, seeking to provide definitions, attributes, and metrics related to sustainability requirements. In this paper, a systematic mapping was conducted and 29 papers were analyzed. The analysis revealed that there are few studies defining sustainability requirements. Additionally, there is a gap in the literature concerning characteristics and attributes related to the economic dimension of sustainability. Finally, our analysis reinforced the need for the standardization of metrics measuring the satisfaction of sustainability requirements.*

1. Introduction

Sustainability is a widely discussed topic nowadays. One of the main reasons is the concern over climate change, which can have serious consequences for humanity. According to data provided by the International Monetary Fund [5], the average temperature of the planet increased by about 1.8°C in the year 2023, compared to the average temperatures recorded between 1951 and 1980. Additionally, there are only six years left to achieve the seventeen sustainable development goals set for the UN's 2030 Agenda.

As stated by Peters et al. [7], a commonly used definition of sustainability is the one proposed by Brundtland [15], who defends that sustainability must "meet the needs of the present without compromising the ability of future generations to meet their own needs.". This definition reinforces the concern not only of meeting the needs and aspirations of the current generation but also of ensuring that future generations have the same opportunities. Additionally, the Karlskrona Manifesto [4] proposed five dimensions of sustainability: environmental, social, economic, technical, and human.

Similar to sustainability, computing is a field that is currently in high demand, increasingly impacting people and businesses. A study conducted by IBM [6] shows that 32% of companies have already actively implemented Artificial Intelligence in their businesses, and an additional 42% reported they are exploring AI. Another study [8] reports that the year 2024 will see a 6.2% growth in the information technology market compared to the previous year. Thus, we have observed in recent years a massive increase in computing in the global context.

Computing contributes to sustainable aspects of daily life through digitalization, such as the reduction of paper use, which benefits the environmental dimension [14]. However, a sustainability dimension cannot be seen as an isolated factor. Penzenstadler

[13] argues that sustainability is an encompassing concept, and one aspect of it cannot be strengthened without considering the other dimensions. As a result, sustainability must be considered a first-class quality attribute and specified as a non-functional requirement of software systems [9]. The establishment of system requirements plays a fundamental role in the application of these practices. Kotonya and Sommerville [10] define system requirements as specifications of the services that the system should provide, constraints, and basic information for developing the system.

In this paper, we aim to investigate definitions, characteristics, and metrics related to sustainability requirements in software systems. Considering this objective, we conducted a systematic mapping study with a focus on answering the following research questions:

- RQ1: How do studies define sustainability requirements in the software development field?
- RQ2: How are sustainability requirements classified in the software development field?
- RQ3: What are the main metrics for evaluating the satisfaction of sustainability requirements in software systems?

This study is structured as follows. In section 2, we present the background and related work, by presenting important concepts such as sustainability and non-functional requirements. In section 3, we describe the steps taken to conduct the systematic mapping study. Section 4 provides the results. Finally, in section 5, we present the conclusion of this study.

2. Background and Related Work

2.1. Sustainability

According to the United Nations Brundtland Commission [15], sustainable development seeks to "meet the needs and aspirations of the present without compromising the ability to meet those of the future". This definition emphasizes the balance between current needs and long-term goals, ensuring that development does not deplete resources or harm future generations. Goodland [11] defines four dimensions of sustainability: environmental, economic, social, and human, also called individual. However, in the context of software systems, it is necessary to add the technical dimension [12]. In this way, the five dimensions of sustainability are defined as follows:

- **Technical dimension:** it is concerned with the long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements [12].
- **Environmental dimension**: it seeks to improve human welfare by protecting natural capital, including water, land, air, minerals, and ecosystem services [11].
- **Social dimension:** social sustainability means maintaining social capital and preserving the societal communities in solidarity [12]. According to Goodland [11], social capital is related to “investments and services that create the basic framework for society”.

- **Economic dimension:** it refers to maintenance of capital, or keeping capital intact [11].
- **Human dimension:** it is concerned with the human capital, which is constituted by health, education, skills, knowledge, leadership, and access to services [11].

2.2. Non-functional requirements

According to Kotonya and Sommerville [10], non-functional requirements are distinct from functional requirements in that they are not related directly to the specific functions or features of a system. Instead, non-functional requirements impose constraints and limitations on the system's performance and development process. These requirements specify external constraints that the system must adhere to, impacting how the system operates rather than what it does.

Davis [16] elaborates on this by defining non-functional requirements as the overarching attributes that a system must exhibit. These attributes include portability, reliability, efficiency, human engineering, testability, understandability, and modifiability. Each of these characteristics contributes to the system's overall quality and its ability to meet user and stakeholder needs.

Calero et al. [17] further extend the concept by linking non-functional requirements to software sustainability. They propose that software sustainability can be framed as a specific type of non-functional requirement. This perspective aligns with the quality attributes outlined in the ISO/IEC 25010 standard, which includes various characteristics and sub-characteristics related to software quality. By defining sustainability in terms of these non-functional requirements, Calero et al. [17] emphasize the importance of considering long-term impacts and environmental considerations in the software development process.

2.3. Related Work

Other secondary studies addressed the topic of sustainability requirements. An important contribution in the domain of sustainability requirements is the work of Bambazek et al. [3]. The authors conducted a systematic mapping study, analyzing 55 publications to identify 29 approaches to developing sustainable software systems published since 2000. They also presented the evolution of these identified approaches over time, revealing how publications and authors have influenced each other. Besides, their research highlights several key trends and gaps in the field. Additionally, they addressed the definitions of sustainability employed, the iterative application of the approaches, and the availability of tool support for practitioners. One of the critical observations made was the increase in publications on sustainability-oriented RE approaches in recent years, with most studies adhering to a multidimensional concept of sustainability.

Moreira et al. [18] have also made a significant contribution to the field by developing a reusable catalog of sustainability requirements, which relates to the focus of this study. They developed a reusable sustainability requirements catalog aimed at addressing the challenges associated with sustainability in software development. Their study emphasizes the importance of changing software development practices to incorporate sustainability considerations, particularly in response to the pressing issue of climate change affecting the ICT sector. The authors collected data through a systematic mapping study, which was synthesized into feature models and represented using iStar for greater expressiveness and configurability. The catalog underwent a qualitative evaluation by 50 domain experts, with around 79% rating it as "Good" or "Very Good." Despite these positive results, some participants identified weaknesses, prompting the authors to refine the catalogue for enhanced completeness and utility.

Another work related to our study was conducted by Venters et al. [19], who explored the concept of sustainability requirements through a corpus-assisted discourse analysis study. Their research sought to understand how the term 'sustainability requirement' is used within software and requirements engineering and related fields. They suggested key focus questions that need to be addressed to establish a shared operative understanding of the term.

Our work differs from the related studies in several key aspects, including focus, scope, and comprehensiveness. Unlike Bambazek et al. [3], whose research focuses on identifying approaches for developing sustainable software systems, our study addresses sustainability requirements by presenting clear definitions, characteristics, and metrics. Additionally, while Moreira et al. [18] created a catalogue concentrating on two dimensions of sustainability (social and technical), our study expands its focus to include four dimensions (economic, social, technical, and environmental). Although their work is validated by experts, our approach provides a broader framework, but we did not perform validation. Lastly, in contrast to Venters et al. [19], who concentrate on defining and utilizing the term "sustainability requirement" within software and requirements engineering, our work not only aims to define sustainability requirements but also delves into additional aspects, including metrics and quality attributes.

3. Methodology

We employed a systematic mapping study. According to Petersen et al. [1], systematic mapping summarizes and provides an overview of a research area. Furthermore, this methodology is particularly valuable for identifying the quantity and types of research conducted, as well as the results available in the field. This approach helps in understanding the current state of research and identifying trends, gaps, and areas requiring further investigation.

In the following subsections, the detailed steps taken to execute this systematic mapping study are presented. This includes the criteria used for study selection, the databases and sources consulted, and the methods employed to analyze and interpret the data collected.

3.1. Research Questions

The main objective of this study is to provide an overview of the literature regarding sustainability requirements in the software development field. Based on that, the following research questions were formulated:

- RQ1: How do studies define sustainability requirements in the software development field?

- RQ2: How are sustainability requirements classified in the software development field?
- RQ3: What are the main metrics for evaluating the satisfaction of sustainability requirements in software systems?

The first research question aims to gather definitions for sustainability requirements from the literature sources. The second research question seeks to identify concrete practices that can be implemented to ensure the fulfillment of sustainability requirements. In this context, a catalog of sustainability requirements was developed to address this question. Finally, the third research question intends to present the main metrics used to evaluate whether a software system meets the sustainability requirements.

3.2. Search Strategy

The search strategy used in this study was a hybrid search, which included the conduction of an automatic search in digital libraries and the application of backward snowballing to the set of selected studies [2]. The following libraries were selected to conduct the automatic search, as proposed in the work of Bambazek et al. [3]:

- ACM Digital Library
- IEEE
- Scopus
- Web Of Science

The search string used was also inspired by the study by Bambazek et al. [3]:

(*green* OR *sustainable* OR *sustainability*) AND (*requirement* OR *requirements*)

The search was applied only to the titles of the publications. It was necessary to adapt the search string for each library due to differences in their respective syntaxes. Due to the high number of publications from other areas that were not relevant to this research objective, results from certain databases were restricted by field. In the results obtained from Scopus, publications were filtered by “Subject Area”, specifically selecting “Computer Science”. The same approach was applied to results obtained from Web of Science. The following filters were applied: “Computer Science Software Engineering”, “Computer Science Information Systems”, “Computer Science Interdisciplinary Applications”, “Computer Science Theory Methods”, “Computer Science Artificial Intelligence”, “Computer Science Hardware Architecture”, and “Green Sustainable Science Technology”.

3.3. Study Selection

In this stage, the inclusion (IC) and exclusion criteria (EC) were defined to filter the studies obtained through the hybrid search. To expand the results obtained with the automatic search, we conducted a backward snowballing of the primary studies identified by the automatic search. Table 1 outlines the criteria used in this study.

Table 1. Inclusion and exclusion criteria

Inclusion Criteria 1 (IC 1)	Publications addressing the topic of sustainability requirements and contributing to answering the research questions.
Inclusion Criteria 2 (IC 2)	Papers written in English.
Inclusion Criteria 3 (IC 3)	Reviewed papers published in journals, conferences, and workshops.
Exclusion Criteria 1 (EC 1)	Gray Literature.
Exclusion Criteria 2 (EC 2)	Duplicated papers.

The first search step was the conduction of an automatic search in digital libraries (ACM, IEEE, Scopus, and Web of Science) in June 2024. This search returned 211 unique titles.

The second search step involved applying inclusion and exclusion criteria to the set of studies obtained through the automatic search. After applying these criteria, 16 primary studies were selected from the initial set of 211 publications. A backward snowballing iteration was conducted on the first set of selected primary studies. In total, 385 unique titles were obtained through snowballing, expanding the pool of potentially relevant studies. After applying the inclusion and exclusion criteria again, 13 additional studies were selected. Finally, the final set of studies selected for this systematic mapping has 29 primary studies. Figure 1 illustrates the selection process.

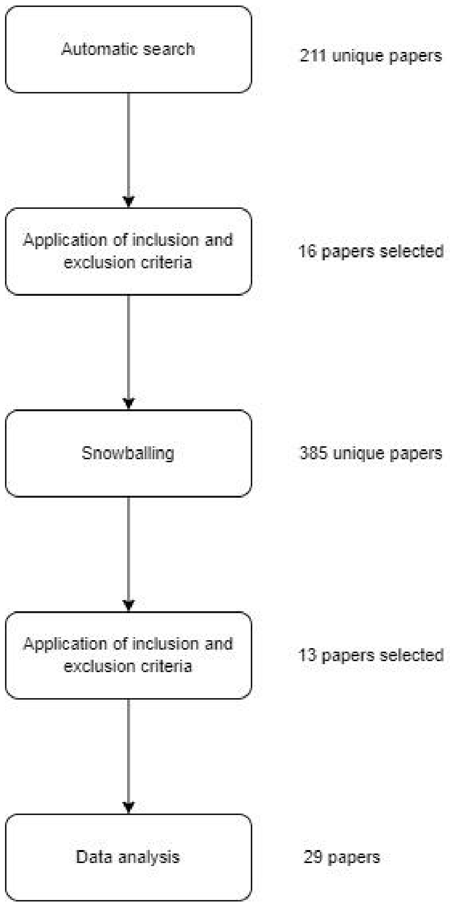

Figure 1. Study Selection Process

3.3. Data Extraction

The data related to each of the research questions were extracted from the selected studies. Additionally, study metadata and study type were also collected. These two pieces of information were used to compose an analysis of the overview of the selected publications in the following subsections. Table 2 presents the information and the description extracted from the selected studies.

4. Results

4.1. Overview of primary studies

The primary studies were published between 2003 and 2022. Figure 2 presents the distribution of the 29 studies published per year. Analyzing the publication trends reveals that 2014 was the peak year, with the highest number of publications, totaling five articles. Following closely, 2015 also saw a notable number of publications, with four articles published that year.

Table 2. Data Extraction

Information	Description

Study Metadata	Title, authors, and year of publication.
Study Type	Classification of the study as empirical or non-empirical
Definitions of sustainability requirements (RQ1)	Sustainability requirements concepts.
How sustainability requirements are classified (RQ2)	Attributes and characteristics related to each one of the sustainability dimensions: technical, environmental, social, economic, and human.
Main metrics for evaluating the satisfaction of sustainability requirements in software systems (RQ3)	The metrics and indicators that were used to evaluate the impact and effectiveness of the implemented sustainability requirements.

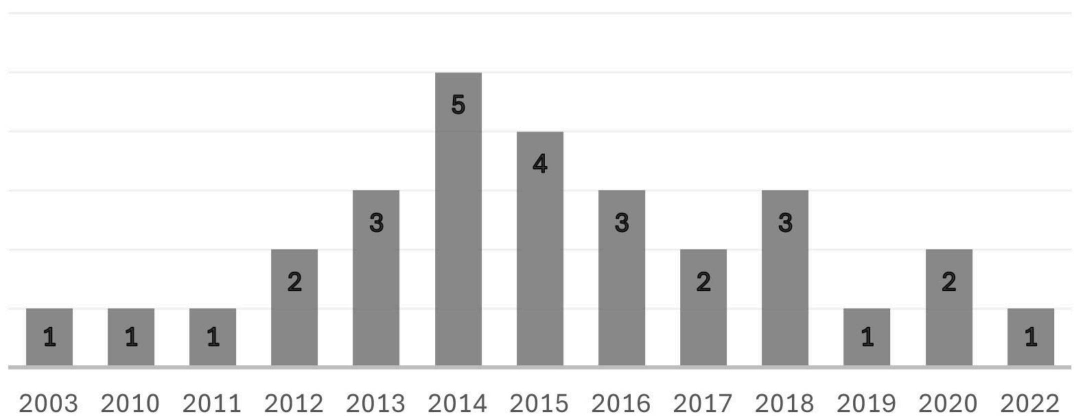

Figure 2. Selected studies published by year

In addition, the studies were categorized into empirical and non-empirical types. The analysis revealed that out of the total studies, 12 were empirical, involving data collection and analysis through experiments, surveys, or case studies. On the other hand, 17 studies were classified as non-empirical, focusing on theoretical frameworks or conceptual analyses without direct empirical validation. This analysis, presented in Table 3, helped to understand the methodological approaches employed in the selected studies and their implications for the robustness of the results.

Table 3. Classification of selected studies

Classification	Selected papers
Empirical	[S3, S6-S10, S15, S20, S21, S25, S27, S29]
Non-empirical	[S1, S2, S4, S5, S11-S14, S19-S22, S22-S24, S26, S28]

In total, 67 authors contributed to the selected studies. The most prolific contributors were Birgit Penzenstadler (9 papers), Leticia Duboc (5 papers), Ruzanna Chitchyan (5 papers), and Colin C. Venters (5 papers).

4.2. RQ1 - How do studies define sustainability requirements?

The analysis of the primary studies revealed that few studies explicitly define "sustainability requirements". Out of the reviewed literature, only 4 papers provided a clear definition of what constitutes sustainability requirements. On the other hand, during the selection of articles for this study, it was observed that many papers address broader definitions, such as "sustainability" or "sustainable software". Those papers were not included in the research, since they do not address the definition of sustainability requirements.

According to Roher and Richardson [S1], sustainability requirements "*specify the behavior of a system (such as requirements to reduce the energy consumption of a system) as well as influence user behavior (the system encourages sustainable actions)*". Huber et al. [S29] define a sustainability requirement as "*a requirement for a sustainable software system that concerns sustainability*". In turn, the authors define a sustainable software system as "*a software system that supports the transition towards sustainable development*".

Becker et al. [S5] argue that sustainability requires simultaneous consideration of environmental resources, societal and individual well-being, economic prosperity, and the long-term viability of technical infrastructure. Thus, Raturi et al. [S2] define sustainability requirements for each of the five sustainability dimensions proposed in the Karlskrona Manifesto [S26]:

- ***Environmental***: requirements relating to resource flows, including energy, which can be obtained and analyzed by Life Cycle Analysis (LCA).
- ***Human***: aspects of human sustainability are addressed by privacy, security, protection, and usability. There is a huge focus on personal health and well-being. An example would be an application that suggests taking a break after a certain period of work.
- ***Social***: a part of social sustainability requirements is considered in the form of political, organizational, or constitutional requirements such as laws, policies, etc.
- ***Economic***: it considers budgetary constraints and costs, as well as market requirements and long-term business objectives that are translated or decomposed into requirements for the system in question. Economic concerns are at the heart of most industrial enterprises.

- ***Technical***: technical sustainability requirements are related to non-obsolescence requirements, as well as the traditional characteristics of maintenance quality, support, reliability, and portability, which lead to the longevity of a system.

We observed that there is a gap in the literature where more precise definitions of sustainability requirements are needed. The lack of clear definitions for what a sustainability requirement is makes it challenging to introduce this field to those seeking to learn about it. Given the multifaceted nature of sustainability, it is important to provide more comprehensive descriptions of what sustainability requirements may encompass. Therefore, clear definitions serve as a foundational element and offer a starting point for researchers and practitioners to build knowledge and develop practical approaches.

4.3. RQ2 - How are sustainability requirements classified in the software development field?

To address this question, we aimed to identify attributes or characteristics that contribute to sustainability. The identified characteristics were refined into subcharacteristics, which were also derived from the selected primary studies. The refinement of the technical dimension characteristics was carried out according to the ISO/IEC 25010 standard. The refinement of characteristics of the other dimensions was partly based on proposals from the selected articles and partly on our suggestions. In total, 20 studies were analyzed to cover four of the five sustainability dimensions proposed in the Karlskrona Manifesto [S26]. The human dimension was not included, as the attributes covered in the social dimension are also used to describe the human dimension, as in the study conducted by Condori-Fernandez and Lago [S8]. They argued that the social dimension and the human dimension share the same social nature and what makes them different is the perspective: the social dimension encompasses larger groups, such as society and organizations; while the human dimension encompasses the individual, such as the end-user and the citizen.

Some dimensions, such as the technical and environmental ones, share some common attributes, for example, reusability. It is important to remember that the more complex the system is, the more probable it is to affect one, if not all, of the five dimensions, especially when considering that many of the dimensions are connected through common goals [S25].

4.3.1. Technical dimension

The technical dimension of sustainability involves the longevity of information, systems, and infrastructure, and their proper adaptation to evolving environmental conditions [S26]. In this way, the technical sustainability requirements are strongly linked to the non-obsolescence of software systems and include other quality attributes, such as maintainability, supportability, reliability, and portability [S2]. Maintainability is mentioned in 11 studies, as presented in Table 4. According to Venters et al. [S16], maintainability encompasses modifications and repairs. Reusability and modifiability [S24] are subcharacteristics of maintainability since the first one considers the extent to which system components can be reused in other systems and the latter addresses the ability to introduce changes quickly and cost-effectively [S20]. Reliability is related to the continuity of correct service [S16], and it is mentioned in 5 studies. It addresses availability (the readiness for correct service [S16]), which can be enhanced by recoverability, as a software system can restore its desired level of performance in the event of a failure [S8]. Albertao et al. [S20] define portability as "*the ability of the system to run under different computing environments"*. A subcharacteristic of portability is adaptability, which is mentioned in 2 studies as an important attribute for the technical dimension of sustainability. Usability is mentioned in 5 studies, and it is defined as "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use*" [S16]. Learnability is related to usability since it encompasses the accomplishment of tasks in the system without the need for assistance [S20]. Penzenstadler et al. [S4] define security as "*the degree of resistance to, or protection from, harm to information*". For this characteristic, it is important to consider integrity, defined as the absence of improper system alteration [S16]. According to Condori-Fernandez and Lago [S8], interoperability is one of the best contributors to sustainability. Interoperability refers to the ability of a software system to cooperate with other relevant software systems [S8] and is related to the compatibility attribute [S8]. Condori-Fernandez and Lago [S8] also proposed functional suitability as a quality attribute related to the technical dimension. It is divided into functional correctness [S16] and functional appropriateness.

Many authors address the topic of the technical dimension of sustainability. In total, 15 of the 29 studies discuss this theme. Maintainability was the most mentioned attribute in the primary studies. It plays a critical role in the longevity of systems and contributes to other attributes, such as reusability and modifiability [S10], for example.

Table 4 - Technical Dimension of Sustainability

Technical Dimension	Maintainability [S2] [S6] [S7] [S8] [S10] [S12] [S15] [S16] [S22] [S23] [S28]	Reusability [S8] [S24]
		Modifiability [S8] [S24]
	Reliability [S2] [S8] [S9] [S10] [S16] [S27]	Availability [S8] [S9] [S16]
		Recoverability [S8] [S15]
	Portability [S2] [S8] [S10] [S24]	Adaptability [S6] [S8]
	Usability [S6] [S7] [S9] [S10] [S27]	Learnability [S8] [S20]
	Security [S6] [S9] [S10]	Integrity [S16]
	Compatibility [S8]	Interoperability [S8] [S9] [S16]
	Functional suitability [S8]	Functional correctness [S8] [S16]
		Functional appropriateness [S8]

4.3.2. Environmental dimension

The environmental dimension involves attributes related to resource flows, including energy, and can be analyzed by Life Cycle Analysis (LCA) [S2]. LCA evaluates the environmental impact of a product's life from the extraction of raw materials to its disposal or recycling [S5]. The usage of materials and resources is an important topic covered by the environmental dimension [S6] [S8]. This point is connected with efficiency [S8], which aims to minimize electronic waste by extending hardware lifetime and minimizing energy consumption through less computer usage time [S20]. According to Condori-Fernandez and Lago [S8], maintainability is one of the best contributors to the environmental dimension of sustainability, as it is related to reusability. Albertao et al. [S20] also propose that reusability is connected to the environmental dimension of sustainability since it minimizes environmental impact through less effort in producing a system. Project's footprint is related to natural resources and environmental impact used during software development [S20], such as soil, atmospheric, and water pollution [S6]. Table 5 presents the attributes related to the environmental dimension of sustainability and the studies that discuss these attributes.

The environmental dimension is also addressed by many studies. Just as in the technical dimension, maintainability also plays an important role in the environmental dimension, as it is linked to reusability. However, most studies addressing this dimension focus on aspects related to efficiency, especially energy consumption.

Table 5 - Environmental Dimension of Sustainability

<table>
<tr><td rowspan="4">Environmental dimension</td><td>Maintainability [S8]</td><td>Reusability [S8] [S20] [S24]</td></tr>
<tr><td rowspan="2">Efficiency [S8]</td><td>Resource utilization [S6] [S8]</td></tr>
<tr><td>Energy consumption [S2] [S20] [S24]</td></tr>
<tr><td>Project's Footprint [S20] [S24]</td><td>Soil, atmospheric, and water pollution [S6]</td></tr>
</table>

4.3.3. Social dimension

Social sustainability requirements encompass political, organizational, or constitutional demands, such as in-laws, policies, etc [S2]. According to Al Hinai and Chitchyan [S13], the notion of social sustainability does not directly pertain to software functionality. In Requirements Engineering, it has been categorized as a soft goal [S13]. In this way, Condori-Fernandez and Lago [S8] proposed confidentiality, authenticity, trust, and freedom from risk as the main quality attributes related to the social dimension of sustainability. Confidentiality and authenticity are related to security and trust is related to satisfaction [S8]. Freedom from risk is concerned with health risk and safety risk mitigation [S8]. Al Hinai and Chitchyan [S13] proposed the topic of equality, which can be refined into access to services and fairness. Access to services is concerned with providing adequate services to all types of stakeholders to enable them equal opportunities to complete a given goal [S13]. Fairness is concerned with a fair selection of stakeholder goals to be implemented in the system [S13]. Johann and Maalej [S14] addressed the topic of reliability in their study as they questioned how software systems will react to collapsing infrastructures. They argued that important software systems should be designed in a way that they are still available even in an unreliable infrastructure.

In total, 3 studies mention security as one of the main attributes of the social dimension of sustainability, as this attribute addresses specific aspects of protecting stakeholder information. Following security, the next most frequently cited attributes are safety and equality, both mentioned in two studies. Reliability was mentioned in only one study, despite being an important attribute in cases of infrastructure collapses.

Table 6 - Social Dimension of Sustainability

Social dimension	Security [S4] [S8] [S14]	Confidentiality [S8]
		Authenticity [S8]
	Satisfaction [S8]	Trust [S6] [S8]
	Safety [S14] [S16]	Freedom from risk [S8]
	Equality [S6] [S13]	Access to services [S13] [S14]
		Fairness [S13]
	Reliability [S14]	Availability [S14]
		Recoverability [S14]

4.3.4. Economic dimension

The economic dimension addresses budget constraints and costs, as well as market requirements and long-term business objectives that get translated or broken down into requirements for the system under consideration [S2]. It is reflected by the degree to which life cycle costs are minimized, economic efficiency is improved, and capital and product value are protected [S15]. According to Condori-Fernandez and Lago [S8], effectiveness is the main characteristic of the economic dimension, followed by reliability and satisfaction. Usefulness is related to satisfaction, while recoverability and availability are related to reliability [S8]. Duboc et al. [S6] proposed value and customer relationship management as important attributes for the economic dimension as well.

It is noticeable that few studies address characteristics related to the economic dimension. In total, two studies proposed characteristics for this theme [S6] [S8]. The study by Albertao et al. [S20] discusses various quality characteristics and their contributions to the technical, social, and economic dimensions. However, in their study, it was not possible to determine which attributes are the most relevant for this dimension. Considering that the economic dimension is at the core of most industrial ventures [S2] and that it is a key factor in the prioritization of requirements [S5], the scarcity of research on quality attributes for this dimension highlights a significant gap in the literature.

Table 7 - Economic Dimension of Sustainability

<table>
<tr><td rowspan="6">Economic dimension</td><td>Satisfaction [S8]</td><td>Usefulness [S8]</td></tr>
<tr><td rowspan="2">Reliability [S8]</td><td>Recoverability [S8]</td></tr>
<tr><td>Availability [S8]</td></tr>
<tr><td colspan="2">Effectiveness [S8]</td></tr>
<tr><td colspan="2">Value [S6]</td></tr>
<tr><td colspan="2">Customer relationship management [S6]</td></tr>
</table>

4.4. RQ3 - What are the main metrics for evaluating the satisfaction of sustainability requirements in software systems?

To address this question, we analyzed 9 studies that propose software sustainability metrics. It is important to note that it is not just one metric that will determine whether software is sustainable or not. Sustainability is an encompassing concept, with multiple dimensions, and one aspect of it cannot be strengthened without considering the other dimensions [S12]. Therefore, evaluating the satisfaction of sustainability requirements demands a comprehensive approach that integrates various metrics. No single metric can provide a complete assessment, as each dimension of sustainability impacts and interacts with the others.

Ojameruaye et al. [S3] presents the following metrics for four of the five sustainability dimensions:

- Human: measured by the number of people-hours involved in development and implementation.
- Economic: measured by the total development and implementation costs.
- Technical: measured by the capability of the feature to maintain a specified level of performance when used under specified conditions (includes maturity, fault tolerance, and recoverability).
- Environmental: measured through energy usage.

Besides proposing energy usage, total costs, and total working hours as metrics, Oyedeji et al. [S11] present other metrics to measure the satisfaction of sustainability dimensions:

- Rework metric: consists of the total number of functions modified per commit related to adding a new feature/function. It is used in the technical dimension.
- Backlog Management Index (BMI): relates to both the rate of defect arrivals and the rate at which fixes for reported problems become available. It is used in the technical, economic, and environmental dimensions.
- Defect Density: it is the value of the total known defects divided by the size of the software product. It is used as a metric in the economic, environmental, social, and human dimensions.
- Gateway metric: the number of successful task completions. It is used as a metric in the social and individual spheres.

The study published by Seacord et al. [S17] proposed a software sustainability metric called Weighted Modification Requests Days (WMRD). WMRD measures the historic capacity of a sustainment team to satisfy each classification of modification request from a user perspective. In other words, WMRD measures the aggregate time the users of the system wait for a number of changes to be enacted, according to the formula in Figure 3. The formula uses estimated change effort, which can be collected for closed change requests and contrasted with effort estimates. The wait time is calculated as the difference between the date the measure is taken (the snapshot date) and the date the modification request was submitted (the submission date). It is important to note that, being a study published in 2003, they do not rely on the idea of sustainability dimensions defined in the Karlskrona Manifesto. They define a sustainability measure as a combined measure of the sustainment organization, the sustainment team, the customers, and the operational domain in which the software operates.

$$\sum_{i=1}^{\#of\ open\ modification\ requests} estimated\_change\_effort_i * (snapshot_{date} - submission_{date_i})$$

Figure 3. WMRD formula proposed by Seacord et al. [S17]

The study of Mussbacher et al. [S18] proposes that sustainability can be measured through time. They introduce an approach that is based on human time as the unit of measure for sustainable technological development. By framing sustainability in terms of time, their approach emphasizes the importance of allowing individuals to have sufficient time for leisure and personal development. The focus of their work is on the social dimension, intending to promote quality of life for stakeholders.

According to Albertao et al. [S20], sustainability performance is measured and analyzed against a set of quality attributes, which when improved will bring economic, social, and environmental benefits. They propose the following metrics related to sustainability requirements [S20}:

- Instability: it measures the potential impact of changes in a given package. The formula for this metric is described in Figure 4. It ranges from 0 to 1, where 0 means that the package is maximally stable, and 1 means that the package is maximally unstable.
- Abstractness: it measures how much a package can withstand change. The formula for this metric is presented in Figure 5. It ranges from 0 to 1, where 0 means the package is completely concrete and 1 means completely abstract.
- Distance From Main Sequence: it measures how far a package is from the idealized balance between Instability and Abstractness. The formula for this metric is presented in Figure 6. It ranges from 0 to 1, where 0 means the package has the ideal balance and 1 means the package requires redesign and refactoring.
- Estimated System Lifetime: estimated number of years that the minimum hardware required by the system reached the market.
- Support Rate: it is calculated by the number of user questions that required assistance divided by the number of minutes the system was used in a given session.
- Relative Response Time: it measures performance with a focus on user productivity. It is calculated by the number of tasks with an unacceptable response time divided by the total number of tasks tested.

- Defect Density: the number of known defects divided by the number of lines of code.
- Testing Efficiency: the number of defects found divided by the number of days of testing.
- Testing Effectiveness: the number of defects found and removed divided by the number of defects found.
- Learnability: the number of minutes to accomplish the first critical task without assistance divided by the number of minutes the system was used by the user.
- Effectiveness: the number of tasks completed without assistance divided by the total number of tasks.
- Error Rate: the number of tasks that were completed but deviated from the normal course of action divided by the total number of tasks.
- Estimation Quality Rate: calculated by dividing the number of iterations where the difference was +/- 20% by the number of total iterations in the project.

$$I = \frac{Ce}{Ca + Ce}$$

Where:

Afferent Couplings (Ca): The number of classes outside a package that depend upon classes within the package.
Efferent Couplings (Ce): The number of classes inside a package that depend upon classes outside the package.

Figure 4. Instability formula proposed by Albertao et al. [S20]

$$A = \frac{Na}{Nc}$$

Where:

Na: The number of abstract classes in a given package
Nc: The number of classes inside a package that depend upon classes outside the package.

Figure 5. Abstractness formula proposed by Albertao et al. [S20]

$$D = |A + I - 1|$$

Figure 6. Distance From Main Sequence formula proposed by Albertao et al. [S20]

Capra et al. [S21] argue that if performance is measured from a user's perspective (in terms of response time) then improving software performance might increase CPU usage. It suggests that, for software applications, energy efficiency and performance represent separate and possibly conflicting issues.

The study of Beghoura et al. [S10] delved into the metrics related to energy consumption in software systems. They proposed metrics related to the hardware components responsible for energy consumption when used by the software:

- Computation metrics: measurement counters related to the set of processing units that are dedicated to performing the computation operations. It corresponds to the quantification of the energy consumed by the software when performing operations on the computing resources.
- Data storage metrics: measurement counter related to the set of storage components used by the software to perform data storage operations on the local storage components. The objective is to measure the amount of energy consumed when software is performing the operations of reading and writing on the storage devices.
- Communication metrics: measurement counters related to the set of communication components to count the number of operations or the amount of data that have been sent or received over the network interfaces.

It is important to highlight that it is not only the factors mentioned above that influence the energy consumption patterns in IT devices. Other external factors, such as hardware configuration and battery aging, can also have a significant impact [S10].

Johann et al. [S19] also present metrics related to energy consumption, but they are primarily focused on measuring the energy consumption directly associated with the software. They defined a generic metric of energy efficiency as the amount of useful work done divided by the used energy. Besides, they calculated the amount of energy consumed for executing tasks such as sorting algorithms. In this way, they argue that energy consumption of specific parts of the software can be measured with a white box method and programmers can find resource-intensive code.

It is important to mention that Penzenstadler et al. [S4] reinforce the need for metrics standards for the different dimensions of sustainability. They suggest defining a set of metrics by relying on existing metrics available, such as ISO 14000 Family for environmental sustainability and ISO 26000 for social sustainability.

The selected studies revealed a range of metrics that help evaluate the satisfaction of sustainability requirements. Most of the metrics presented are related to the technical or environmental dimension, considering quality attributes and energy consumption, respectively. However, as noted by Penzenstadler et al. [S4], further studies are needed to propose metric standards. Thus, adopting standardized metrics can help organizations set clear sustainability goals, monitor progress, and demonstrate their commitment to sustainable development.

5. Conclusion, limitations, and future work

In this paper, 29 primary studies related to sustainability requirements in the field of software engineering were analyzed. The research addressed questions concerning definitions of sustainability requirements, classification of these requirements, and metrics related to sustainability in software systems. Our analysis revealed a significant gap in studies presenting clear definitions of sustainability requirements. On the other hand, many studies do discuss broad definitions of sustainability or sustainable software. We observe that more research should focus on defining sustainability requirements, as this is a foundational step for individuals, such as researchers, students, and professionals seeking to understand the area.

Another aspect to consider is the lack of studies addressing the characteristics and metrics related to the economic dimension of sustainability. Only two studies were found that discuss this topic [S6] [S8], along with their relevant characteristics for the context of software engineering. This lack of research highlights a notable gap in the field, as the economic dimension plays a crucial role in evaluating the overall sustainability of software systems. Incorporating economic factors into sustainability assessments is essential for understanding the cost-effectiveness and long-term financial implications of sustainable software practices. Future research should aim to explore this dimension more thoroughly, investigating how economic considerations can be integrated into sustainability frameworks and metrics.

Our study faces some threats to validity. Despite our use of four distinct search engines (ACM, IEEE, Scopus, and Web of Science) there remains a risk that important studies may not have been selected. While we did not impose a time restriction on the studies, our search concluded in June 2024, and significant studies published after this period may not have been included. To mitigate the potential problems of the automatic search gap and enhance the comprehensiveness of our search, we incorporated an iterative backward snowballing approach. Nevertheless, the possibility of missing relevant research cannot be eliminated, which may impact the completeness of our study.

The validity of the data collected is connected to the risk of potential data extraction bias. Data extraction was performed by a single author and reviewed by the advisor and co-advisor. During this process, we concentrated on addressing the Research Questions (RQs) to ensure the collected data was aligned with the predefined questions. We utilized a Google Spreadsheet to record the data from the selected studies, which also served to centralize the information and extracted data.

To ensure research validity, we followed the steps proposed by Petersen et al. [1]. However, some risk was associated with the catalog developed for our Research Question 2, which involved creating a catalog of quality attributes related to the sustainability dimensions. Although the advisor and co-advisor, who possess expertise in requirements engineering, provided feedback and suggested adjustments, the proposed catalog of sustainability quality attributes was not validated by additional professionals in the field. Consequently, the catalog's effectiveness in real-world scenarios remains uncertain.

As future work, we recommend the definition of standards for metrics related to the satisfaction of sustainability requirements in software systems, considering all five dimensions. Developing these standards will provide software engineers with a central reference point for evaluating and ensuring that sustainability criteria are satisfactorily met. Clear and standardized metrics are essential for assessing the impact of sustainability practices and for making informed decisions throughout the software development lifecycle. Future work should focus on creating and validating these metrics standards. This effort will involve collaboration among academia, industry, and standardization organizations to ensure the metrics address real-world needs and challenges.

Appendix - List of studies selected for the systematic mapping

[S1] ROHER, Kristin; RICHARDSON, Debra. Sustainability requirement patterns. In: 2013 3rd international workshop on requirements patterns (RePa). IEEE, 2013. p. 8-11.

[S2] RATURI, Ankita et al. Developing a sustainability non-functional requirements framework. In: Proceedings of the 3rd International Workshop on Green and Sustainable Software. 2014. p. 1-8.

[S3] OJAMERUAYE, Bendra; BAHSOON, Rami; DUBOC, Leticia. Sustainability debt: A portfolio-based approach for evaluating sustainability requirements in architectures. In: Proceedings of the 38th International Conference on Software Engineering Companion. 2016. p. 543-552.

[S4] PENZENSTADLER, Birgit et al. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. IEEE software, v. 31, n. 3, p. 40-47, 2014.

[S5] BECKER, Christoph et al. Requirements: The key to sustainability. IEEE Software, v. 33, n. 1, p. 56-65, 2015.

[S6] DUBOC, Leticia et al. 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. p. 6-16.

[S7] SAPUTRI, Theresia Ratih Dewi; LEE, Seok‐Won. Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition. Journal of Software: Evolution and Process, v. 32, n. 8, p. e2254, 2020.

[S8] CONDORI-FERNANDEZ, Nelly; LAGO, Patricia. Characterizing the contribution of quality requirements to software sustainability. Journal of systems and software, v. 137, p. 289-305, 2018.

[S9] CONDORI FERNANDEZ, Nelly; LAGO, Patricia. The influence of green strategies design onto quality requirements prioritization. In: Requirements Engineering: Foundation for Software Quality: 24th International Working Conference, REFSQ 2018, Utrecht, The Netherlands, March 19-22, 2018, Proceedings 24. Springer International Publishing, 2018. p. 189-205.

[S10] BEGHOURA, Mohamed Amine; BOUBETRA, Abdelhak; BOUKERRAM, Abdallah. Green software requirements and measurement: random decision forests-based software energy consumption profiling. Requirements Engineering, v. 22, p. 27-40, 2017.

[S11] OYEDEJI, Shola; SEFFAH, Ahmed; PENZENSTADLER, Birgit. Sustainability quantification in requirements informing design. 6th Int. Work. Requir. Eng. Sustain. Syst, v. 1, 2017.

[S12] PENZENSTADLER, Birgit. From requirements engineering to green requirements engineering. Green in software engineering, p. 157-186, 2015.

[S13] AL HINAI, Maryam; CHITCHYAN, Ruzanna. Engineering requirements for social sustainability. In: ICT for Sustainability 2016. Atlantis Press, 2016. p. 79-88.

[S14] JOHANN, Timo; MAALEJ, Walid. Position paper: The social dimension of sustainability in requirements engineering. In: Proceedings of the 2nd International Workshop on Requirements Engineering for Sustainable Systems. 2013.

[S15] PHAM, Yen Dieu; BOURAFFA, Abir; MAALEJ, Walid. Shapere: Towards a multi-dimensional representation for requirements of sustainable software. In: 2020 IEEE 28th International Requirements Engineering Conference (RE). IEEE, 2020. p. 358-363.

[S16] VENTERS, Colin et al. The blind men and the elephant: Towards an empirical evaluation framework for software sustainability. Journal of open research software, v. 2, n. 1, 2014.

[S17] SEACORD, Robert C. et al. Measuring software sustainability. In: International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings. IEEE, 2003. p. 450-459.

[S18] MUSSBACHER, Gunter; NUTTALL, Douglas. Goal modeling for sustainability: The case of time. In: 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE). IEEE, 2014. p. 7-16.

[S19] JOHANN, Timo et al. How to measure energy-efficiency of software: Metrics and measurement results. In: 2012 First International Workshop on Green and Sustainable Software (GREENS). IEEE, 2012. p. 51-54.

[S20] ALBERTAO, Felipe et al. Measuring the sustainability performance of software projects. In: 2010 IEEE 7th International Conference on E-Business Engineering. IEEE, 2010. p. 369-373.

[S21] CAPRA, Eugenio; FRANCALANCI, Chiara; SLAUGHTER, Sandra A. Measuring application software energy efficiency. IT Professional, v. 14, n. 2, p. 54-61, 2012.

[S22] BRITO, Isabel Sofia et al. A concern-oriented sustainability approach. In: 2018 12th international conference on research challenges in information science (RCIS). IEEE, 2018. p. 1-12.

[S23] PENZENSTADLER, Birgit. Towards a definition of sustainability in and for software engineering. In: Proceedings of the 28th annual ACM symposium on applied computing. 2013. p. 1183-1185.

[S24] NAUMANN, Stefan et al. The GREENSOFT Model: A reference model for green and sustainable software and its engineering. Sustainable Computing: Informatics and Systems, v. 1, n. 4, p. 294-304, 2011.

[S25] MENDES, Salvador; ARAUJO, João; MOREIRA, Ana. On the Current Practices for Specifying Sustainability Requirements. In: International Conference on Research Challenges in Information Science. Cham: Springer International Publishing, 2022. p. 426-441.

[S26] BECKER, Christoph et al. Sustainability design and software: The karlskrona manifesto. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015. p. 467-476.

[S27] KOÇAK, Sedef Akinli; ALPTEKIN, Gülfem Isiklar; BENER, Ayse. Evaluation of Software Product Quality Attributes and Environmental Attributes using ANP Decision Framework. In: RE4SuSy@ RE. 2014. p. 37-44.

[S28] PENZENSTADLER, Birgit et al. Software engineering for sustainability: Find the leverage points!. IEEE Software, v. 35, n. 4, p. 22-33, 2018.

[S29] HUBER, Martina Z. et al. Uncovering sustainability requirements: An exploratory case study in canteens. In: CEUR Workshop Proceedings. CEUR, 2015. p. 35-44.

References

[1] PETERSEN, Kai et al. Systematic mapping studies in software engineering. In: 12th international conference on evaluation and assessment in software engineering (EASE). BCS Learning & Development, 2008.

[2] MOURÃO, Erica et al. Investigating the use of a hybrid search strategy for systematic reviews. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, 2017. p. 193-198.

[3] BAMBAZEK, Peter; GROHER, Iris; SEYFF, Norbert. Requirements engineering for sustainable software systems: a systematic mapping study. Requirements Engineering, v. 28, n. 3, p. 481-505, 2023.

[4] BECKER, Christoph et al. Sustainability design and software: The karlskrona manifesto. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015. p. 467-476.

[5] International Monetary Fund. 2022.Climate Change Indicators Dashboard. Annual Surface Temperature Change, https://climatedata.imf.org/pages/access-data. Accessed on May 1, 2024.

[6] IBM. Global AI Adoption Index 2022. Available at: https://www.ibm.com/downloads/cas/GVAGA3JP. Accessed on: August 02, 2024

[7] PETERS, Anne-Kathrin et al. Sustainability in computing education: A systematic literature review. ACM Transactions on Computing Education, v. 24, n. 1, p. 1-53, 2024.

[8] Information Technology Market Analysis, Size And Trends Global Forecast To 2022-2030. Available at: <https://www.thebusinessresearchcompany.com/report/information-technology-global-market-report>. Accessed on May 1, 2024.

[9] PENZENSTADLER, Birgit et al. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. IEEE software, v. 31, n. 3, p. 40-47, 2014.

[10] KOTONYA, Gerald; SOMMERVILLE, Ian. Requirements engineering: processes and techniques. Wiley Publishing, 1998.

[11] GOODLAND, Robert et al. Sustainability: human, social, economic and environmental. Encyclopedia of global environmental change, v. 5, p. 481-491, 2002.

[12] PENZENSTADLER, Birgit; FEMMER, Henning. A generic model for sustainability with process-and product-specific instances. In: Proceedings of the 2013 workshop on Green in/by software engineering. 2013. p. 3-8.

[13] PENZENSTADLER, Birgit. From requirements engineering to green requirements engineering. Green in software engineering, p. 157-186, 2015.

[14] DE MORAES, Gustavo Hermínio Salati Marcondes; CAPPELLOZZA, Alexandre; DE SOUZA MEIRELLES, Fernando. Será o fim do papel? Os avanços tecnológicos e seus possíveis impactos no consumo de papel. Revista Eletrônica de Negócios Internacionais (Internext), v. 6, n. 2, p. 48-65, 2011.

[15] IMPERATIVES, Strategic. Report of the World Commission on Environment and Development: Our common future. Accessed Feb, v. 10, n. 42,427, 1987.

[16] DAVIS, Alan M. Software requirements: objects, functions, and states. Prentice-Hall, Inc., 1993.

[17] CALERO, Coral; MORAGA, M.; BERTOA, Manuel F. Towards a software product sustainability model. arXiv preprint arXiv:1309.1640, 2013.

[18] MOREIRA, Ana et al. A social and technical sustainability requirements catalogue. Data & Knowledge Engineering, v. 143, p. 102107, 2023.

[19] VENTERS, Colin C. et al. 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. p. 3-12.