


# Software industry awareness on sustainable software engineering: a Brazilian perspective

Leila Karita  [ Federal University of Bahia | [leila.karita@ufba.br](mailto:leila.karita@ufba.br) ]

Brunna Caroline Mourão  [ Federal University of Bahia | [brunna.caroline@ufba.br](mailto:brunna.caroline@ufba.br) ]

Luana Almeida Martins  [ Federal University of Bahia | [martins.luana@ufba.br](mailto:martins.luana@ufba.br) ]

Larissa Rocha Soares  [ Federal University of Bahia | [larissars@dcc.ufba.br](mailto:larissars@dcc.ufba.br) ]

Ivan Machado  [ Federal University of Bahia | [ivan.machado@ufba.br](mailto:ivan.machado@ufba.br) ]

## Abstract

Sustainable computing is a rapidly growing research topic spanning several areas of Computer Science. Particularly, it has received increasing attention in the Software Engineering field in the last years, with several studies discussing the topic from a range of perspectives. However, few studies have demonstrated the awareness of software practitioners about the underlying concepts of sustainability in the software development practice. In an earlier investigation, we performed a preliminary study on the practitioners' perception under four main perspectives: economic, social, environmental, and technical. This study extended the previous survey and reached a number of ninety-seven respondents from Brazilian companies. The extension aims to expand the results to compare and explore the previous findings in a more in-depth way. The novel results confirmed the evidence raised in the original survey that sustainability in the context of Software Engineering is a new subject for practitioners. However, professionals have shown interest in the topic, and there is a general understanding that sustainability should be treated as a quality attribute. Among the observed perspectives, we generated an initial theory that shows software practitioners know the subject around 'Green in Software', even unconsciously. This study brings evidence of how the industry understands and perceives sustainability practices in the software development process.

**Keywords:** *Sustainable Software Engineering, Survey Research, Empirical Software Engineering*

## 1 Introduction

Sustainability has been increasingly discussed in the Software Engineering (SE) field (Mourão et al., 2018). As more and more software applications are launched in the market as a means to make daily activities easier, there is an increased interest in understanding how such solutions might affect the environment.

The impact of technology on daily lives can be seen from two antagonistic perspectives, technology as a contributor to mitigate or produce the environmental impact. Whereas the technology helps organizations address environmental issues when providing many improvements (e.g., virtual meetings and improvements in logistics), it is often responsible for environmental degradation by consuming amounts of energy through engineering processes used to make products, for instance (Calero and Piattini, 2017).

Developing, maintaining, and evolving energy-efficient software solutions is rather challenging (Pinto and Castor, 2017). The software development life-cycle is not suitable for identifying the effects of the software system on sustainability (Dick et al., 2010). Therefore, sustainable thinking is still a new and challenging practice for software engineers and developers.

Environmental sustainability has been deemed as a non-functional requirement (NFR) to consider in a SE process (Calero and Bertoa, 2013; Venters et al., 2014; Becker, 2014; Penzenstadler et al., 2014b). However, this is not commonly employed yet. Authors claim that there is still a need for more discussion on "sustainable requirements" to understand how the term has been used in the SE field (Venters et al., 2017).

It is rather important to explicitly identify sustainability requirements and ensure they could be properly monitored and tested along with the software development life cycle. Indeed, the definition of *sustainable software development* is not clear in the literature yet and maybe misunderstood as a consequence. In a previous work (Karita et al., 2019), we presented a survey study conducted with twenty-five software engineers from Brazilian companies involved in projects from different domains. The study investigated the practitioners' perception of sustainable practices for software development. The study provides readers with an overview of the main discussions about software sustainability concepts. The yielded results indicate an overall lack of knowledge about the topic, in particular, related to sustainable software concepts.

In this study, we conducted an exploratory study to identify whether, and to what extent, the practitioners know about the topic. Therefore, we extended our previous work in two directions: (i) we extended the initial survey, which reached a number of ninety-seven respondents; and (ii) established an initial common understanding of the definitions of "sustainable software" based on the achieved results. From the survey, we are interested in finding out even more insights and evidence that may reveal the importance of promoting the sustainable software development field. This study also aims to leverage the state-of-the-practice in *sustainable software development*, under four main perspectives: economic, social, environmental, and technical.

Additionally, our study deals with the practitioners' lack of knowledge about sustainable software development to extract as much information as possible to explore the state-of-the-practice. In this sense, our earlier study allowed us to identify

that even when the practitioners do not know the formal concept of sustainable software development, they can discuss its importance through transitivity, i.e., they are able to apply general concepts of sustainability in the context of software development.

The survey results confirm the results identified in the former study (Karita et al., 2019). We found that sustainability in the context of software is a new issue for most respondents. The software professionals are not concerned with sustainability throughout the software development life-cycle and have low knowledge about the impacts of not using sustainable practices on the environment. However, they showed interest in the topic, which is a general understanding that sustainability should be treated as a quality attribute. The theory showed that software engineers practically explore the technical and environmental dimensions; that is, they may unconsciously practice ‘Green in Software’.

The remainder of this paper is organized as follows. Section 2 presents the underlying concepts of green and sustainable software. Section 3 presents the survey conducted to investigate the awareness of practitioners of sustainability. Section 4 presents the Grounded Theory conducted to generate an initial common understanding of sustainability. Section 5 provides an in-depth discussion of the yielded results. Section 6 discusses the implications for research and practice. Section 7 discusses related work. Section 8 discusses the threats to validity. Section 9 draws concluding remarks.

## 2 Green and Sustainable Software

Sustainability has been discussed in several sectors of our society. Etymologically, the word sustainable comes from the Latin *sustare*, which means “to sustain”, “to support”, and “to conserve”. The term “sustainable development” was coined in 1987 by Gro Harlem Brundtland, who published a book (Our Common Future) where she stated, “Meeting the needs of the present without compromising the ability of future generations to meet their own needs” (Imperatives, 1987). According to Calero and Piattini (2015a), when we take a closer look at the above definition, we could observe that two fundamental pillars underpin sustainability: “The capacity of something to last a long time” and “the resources used”.

When addressing sustainability in the context of SE, we find several definitions of sustainable SE in the literature (Calero and Piattini, 2015a). Tate (2005) defines sustainable SE as the development, which is able to make a balance between rapid release and long term sustainability. According to Khandelwal et al. (2017), sustainable SE consists of processes and practices that help produce sustainable software and everything related to the software product, be it development or maintenance, taking environmental aspects into account. To Erdelyi (2013), SE can be sustainable by produce sustainable software with environmental awareness and minimizing waste during the software development process.

In general, sustainable SE can be interpreted as the art of developing sustainable software through a sustainable SE process. Its goal is to enhance the SE practices aiming at the direct and indirect consumption of natural resources and en-

ergy, as well as the consequences caused by software systems throughout its life-cycle (Johann et al., 2011).

Sustainability in SE can be expressed in several dimensions. For example, the Imperatives (1987) addresses the *economic*, *social*, and *environmental* dimensions. The *economic* dimension refers to the production and consumption of resources and services. The *social* dimension refers to people and their living and labor conditions, such as education, health, leisure, social equity, livability, and other aspects. The **environmental** dimension refers to the natural resources of the planet, and the way society uses them.

The environmental dimension of sustainability is also called as *green* dimension (Moraga et al., 2017; Calero and Piattini, 2017, 2015b; García-Mireles et al., 2017; Murugesan, 2008). The term *green* has also been interpreted in two ways in the literature, either (1) *green in* software, which is related to develop a more environment-friendly software, i.e., the software is developed in a green manner and produce a green software product; and (2) *green by* software, which refers to software developed focusing on the preservation of the environment, i.e., the software is the tool supporting the sustainability goals.

Similarly, Penzenstadler et al. (2014a) also interpret sustainable software in two ways: (1) as software code that is sustainable, agnostic of purpose; and (2) as a software purpose for achieving sustainability goals. Likewise, Dick et al. (2010) emphasize that sustainable software focuses on reducing the consumption of natural resources and energy in the sense of environmental, social, and economic dimensions.

In addition to those three dimensions, other studies also add the *technical* and *individual* dimensions to analyze the software sustainability. The *technical* dimension addresses the long-term use of software and their evolution in a constantly changing execution environment (Moraga et al., 2017; Lago et al., 2015; Fernandez et al., 2016). The *individual* dimension concerns with how to maintain and create software in a way that takes into account developers’ satisfaction with their work over a long period (Penzenstadler and Femmer, 2013; Penzenstadler, 2014; Becker et al., 2015).

Saputri and Lee (2016) bring an inclusive view regarding those dimensions. They state that sustainability should be considered an integrated concept, taking into account each sustainability dimension. All of these dimensions could be analyzed from various perspectives. Thus, the choice of approach to be adopted should take into account the purpose and scope to be investigated.

Furthermore, others discuss more technical aspects, such as the sustainability of the software life-cycle. According to Johann et al. (2011), green and sustainable software is the enhancement of SE to deal with the consumption of natural resources and energy during the entire software development life-cycle. Moreover, Hilty et al. (2006) and Penzenstadler et al. (2014a) agree that the sustainable software can be interpreted as either a software developed to support sustainability goals or a software code being developed through a sustainable process. Both interpretations converge to a software system that contributes to more sustainable living. While a sustainable software aims to improve the sustainability of humankind on our planet, a sustainable process is a key requirement for developing sustainable software. Thus, a sus-

tainable process may consider environmental, technical, and economic impacts during the software life-cycle and involves the pursuit of sustainable development goals.

In summary, the literature presents several viewpoints, in which different researchers describe them from their perspective and area of expertise. Consequently, the same may happen in the software industry. The lack of a unanimous definition can lead to isolated contributions, and practitioners might see SE sustainability from different perspectives.

### 3 Understanding the awareness of practitioners on sustainability

This study investigates sustainability in the context of software development from an industry standpoint. In this sense, we conducted a survey to gather the opinions of professionals about sustainability without introducing participants to the topic since it could skew the answers. Thus, we aim to identify the awareness of software professionals on *sustainability in SE*.

#### 3.1 Research Questions

We defined six research questions (RQ) to understand software familiarity, application, perceived importance, practices, models, and tools regarding sustainability. These are described next.

- RQ1: Are the professionals familiar with the concepts of sustainability applied to the software development process?** This question investigates at what level the professionals are familiar with concepts related to sustainability in the context of software.
- RQ2: How important is software sustainability for practitioners?** This question investigates whether and at what level professionals consider sustainability an important factor in the software development process from a personal perspective and the software industry perception.
- RQ3: What phases of the software development life-cycle (SDLC) do sustainable practices apply to?** This question identifies in which SDLC phases the developers have adopted any sustainable SE practices in the general and specific scope.
- RQ4: What dimensions of sustainability have been explored in practice (technical, environmental, social, and economic) of software development?** This question aims to investigate which of these dimensions have been most exploited by industry.
- RQ5: What models for sustainable software development have been adopted by the software industry?** This question investigates whether and what models for sustainability in software have been adopted by professionals. The purpose is to obtain models from the industry that we did not retrieve in the previous study.
- RQ6: What tools have been used to support sustainability in the software development process?** Aligned with the preceding RQ, we aim to explore tools that

have been adopted in practice and collaborate with sustainability, based on the intuitive knowledge of the respondents.

#### 3.2 Survey Design

We designed the questionnaire to keep it as brief as possible while still enabling us to collect all relevant information. The stated questions seek to understand practitioners' motivations and knowledge regarding green practices in software development.

In our previous study, we conducted a survey study with twenty-five software professionals from Brazilian companies. This study extends the previous one in the number of respondents to seventy-two new professionals from several Brazilian software companies based in different regions of the country. Therefore, this study presents the results for a total of ninety-seven respondents.

This section encompasses the planning details, execution procedures, and reporting of desired and achieved results<sup>1</sup>. We used the methodology proposed by Kasunic (2005) and applied the research survey principles defined by Kitchenham and Pfleeger (2002). **Figure 1** shows the adapted methodological steps employed in this extended study. Steps 1 to 7 were conducted during the survey design in our previous work. These steps were reviewed and re-applied to accomplish steps 8 to 10 during this extension.

**Target Audience.** To ensure valid results, we only selected professionals with knowledge in software development processes. The following criteria were considered:

1. Professional with experience in the SE field.
2. Professional role in the company. Practitioners should be involved in the software development process in at least one of the following roles: project manager, project leader, system analyst, requirements analyst, system architect, business analyst, developer, tester, product owner, and/or scrum master.

**Questionnaire.** We reviewed and applied the same instrument from the preliminary survey study, in which we specified six information groups. They are: respondents characterization, companies characterization, research object, company development process, faced difficulties, and sustainability as a quality attribute. We next describe each category.

- **Respondents characterization:** In this category, the goal was to investigate the respondent profile, with information about gender, name, age, level of education, and professional experience;
- **Companies characterization:** This category investigates the locality, follow-up, size, time of performance, certifications, level of environmental awareness (any aspect, not only necessarily regarding SDLC processes) of the companies, and function performed by the respondents in the company;

<sup>1</sup>The resulting data are publicly available at <http://doi.org/10.5281/zenodo.3976312>

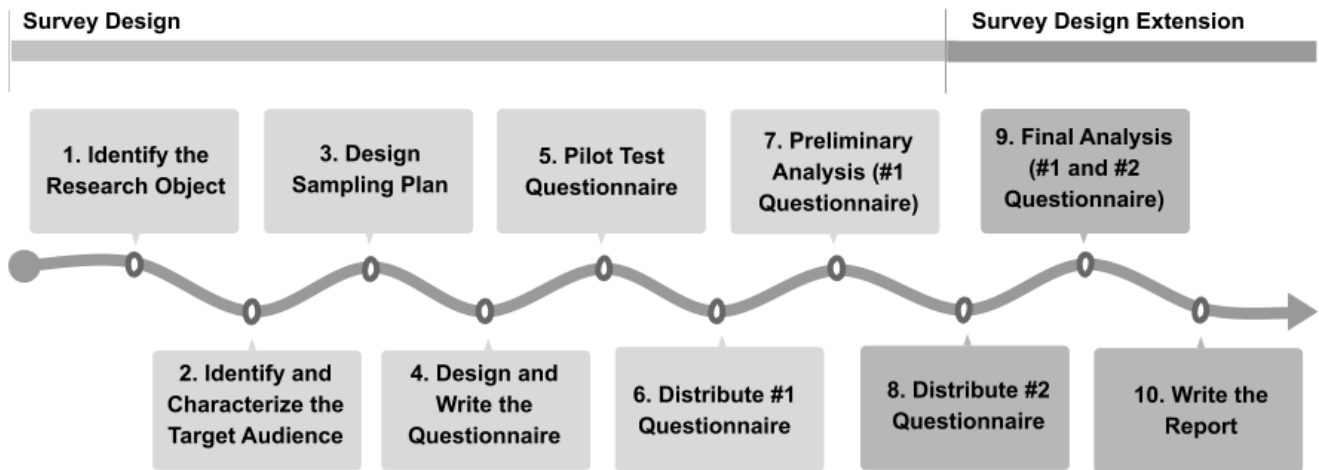


Figure 1. Survey design adapted from (Kasunic, 2005)

- **Research objective:** In this category, the goal was to investigate the respondent's knowledge regarding concepts related to software sustainability, as well as the importance of the respondent relating sustainability to the software context;
- **Company development process:** In this category, the goal was to investigate the software development process of the company the respondent work for and to identify whether and at what level sustainability practices have been applied;
- **Difficulties encountered:** This category investigates the likely benefits expected regarding the application of sustainability practices in the software development processes;
- **Sustainability as a quality attribute:** In this category, the goal was to investigate the interviewee's perception of the importance of using sustainability as a quality attribute in their projects.

We sent the replication of the reviewed questionnaire on February 1st, 2020. The extended survey instrument was hosted on Google Forms, alike the first survey. The URL to access it was sent by: (1) personalized email to our contact list and (2) *Whatsapp* contacts. We closed the survey on February 29th, 2020.

A brief introduction was made with basic information about the purpose of the study with a justification of choice and the importance of the respondent's. Participants were also informed about the privacy policies of the study.

**Data Analysis.** In this section, we present the data analysis process employed in this study. This research is an exploratory and qualitative study. To achieve the objectives, we adopted the following assumptions about the instrument:

1. For closed questions that could combine multiple responses, the sum of percentages could reach a value above 100%.
2. For closed questions that followed the same pattern of responses, we applied a five-point Likert Scale, from Irrelevant (1) to Very important (5).
3. For the open question about the concept of sustainability in the software development process, we applied a

coding strategy. Two of the authors extracted the general themes of the answers. Using these themes, the authors had discussion sessions to develop a single coding scheme. The results were collected and translated into an appropriate graphic image to facilitate understanding.

4. For the other open questions, we include excerpts from the qualitative answers to clarify the results. Each of the excerpts is followed by a number that represents a unique identifier for the respondent who expressed the opinion. For example, [#1] indicates the respondent's answer number 1.

### 3.3 Survey Results

In this section, we report the results of our extended survey study.

#### 3.3.1 Respondents' Demographics

This section describes the demographics of the respondents. We investigated their age, and experience time to draw the profile of the observed sample. Overall, regarding their gender, 63% of the respondents were men, 36% were women, and 1% were others. **Figure 2** shows the respondents age. 1% had between 15 and 19 years, 12% had between 20 and 24 years, 22% had between 25 and 29 years, 27% had between 30 and 34 years, 21% had between 35 to 39 years, 8% had between 40 to 44 years, 6% had between 45 to 49 years, and 3% had more than 50 years. More than 50% of the respondents are mainly concentrated in the 25 to 39-year-old range.

In terms of their professional experience in software development, **Figure 3** shows that 28% had up to 3 years of experience, 18% had between 4 and 6 years, 14% had between 7 and 10 years, and 40% of the respondents had more than 10 years of experience in the industry.

Finally, the respondents informed the roles they play in the companies: 22% work as a software developer, 20% work as a system analyst, 10% work as a requirements analyst, 9% work as a project leader, 7% work as a business analyst and software tester, 5% work as a software architect, 4% work as project management, scrum master and product owner, 3% work as a consultant, 2% as portfolio management, and 1%

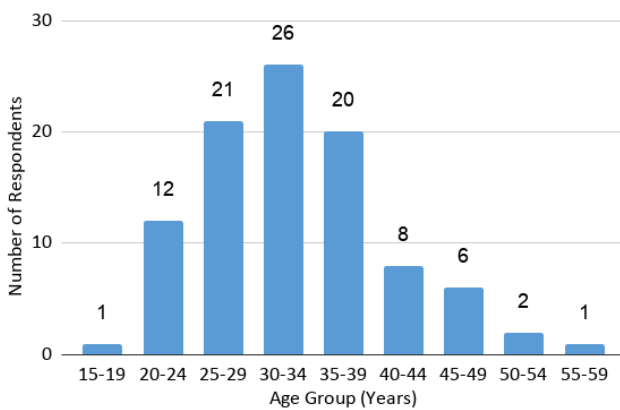


Figure 2. Distribution of respondents by age

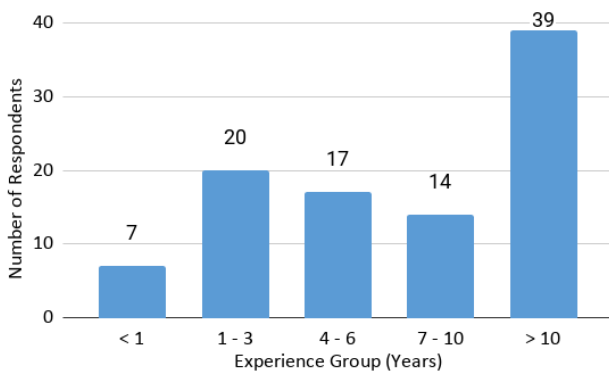


Figure 3. Distribution of respondents by experience time

work as a designer, data analyst, and DBA. In this question, the respondents could answer more than one option.

### 3.3.2 Companies' Demographics

This section describes the demographics of the companies analyzed by respective practitioners in terms of segment and size.

The respondents work in companies of different segments. Figure 4 shows that 51% of the respondents work in software factories, 21% in government companies, and 8% in Startups and Consulting Company, each one. The others add up to 12% working in other segments. About company size, 87% of the respondents reported that the size of the company is "large", that is, it has more than 99 employees.

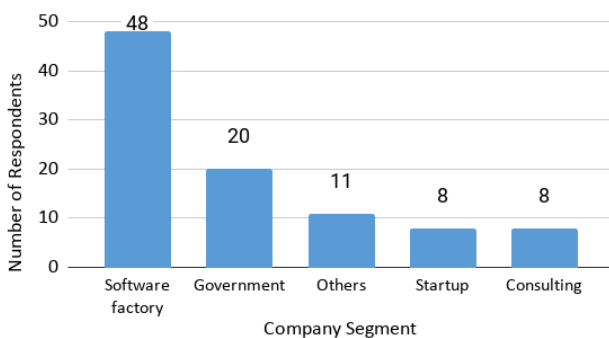


Figure 4. Distribution of respondents by companies segment

In terms of certifications related to the software development process, 48% of the respondents could not tell whether

the company in which they work has some certification. 24% reported that the company is not certified. The other 28% of respondents reported that the company has some of the following certifications: Capability Maturity Model Integration (CMMI) - Levels 3 and 5, MPS.Br (Levels of Improvement of the Brazilian Software Process) - Levels C and G, ISO 27001: 2007 (Information Security Management System), ISO 14001 (Quality Management System), and TMMI (Testing Maturity Model Integration) - Level 3.

### 3.3.3 Answering the research questions

We next discuss the results based on the stated RQs.

#### RQ1: Sustainability concepts

In this question, the goal was to observe the respondents' comprehension both in the general scope, with respect to the conceptual framework on sustainability, and to understand their perception regarding the adequacy of the companies in which they act to the sustainable practices.

Initially, seeking to observe the respondents' level of knowledge, we asked how respondents could self-assess their level of knowledge about Sustainability in the software development process. Figure 5 shows 46% had no knowledge, which was the first contact with the subject; 45% out of the respondents had low knowledge about the subject; 6% had a medium knowledge; and 3% had high knowledge on it.

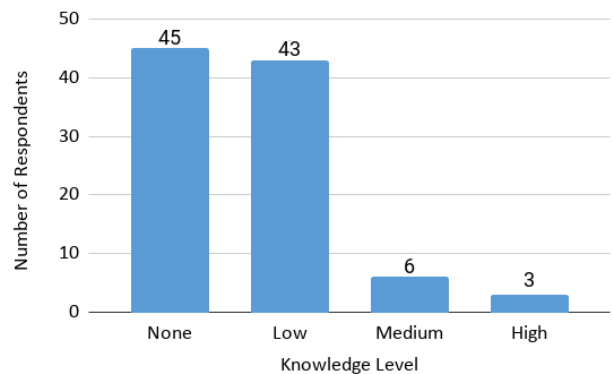


Figure 5. Distribution of respondents by knowledge level on sustainability

Furthermore, we presented six concepts (definitions) about "sustainable software" to participants, available in the literature of relevant authors to the domain. The respondents did not have access to the authors' names and could choose only one of the six options. Our objective was to identify which of those concepts the respondents would be more familiar with. The results are described next. We described each concept in the boxes, followed by the corresponding results.

**Definition 1:** "An application that produces as little waste as possible during its development and operation". Erdelyi (2013).

**34% of the respondents consider this one the most coherent definition.**

**Definition 2:** "Software developed and used in such a way that leaves minimal negative impact on users, environment, economy and society in general". Naumann et al. (2011).

### 28% of the respondents consider this one the most coherent definition.

**Definition 3:** “Software whose impacts on the economy, society, human beings and environment, resulting from the development, deployment and use of the software is minimal and/or has a positive effect on sustainable development”. Dick et al. (2010).

### 26% of the respondents identified themselves with this definition.

**Definition 4:** “Software code being sustainable, agnostic on purpose, or the purpose of the software is to support sustainability goals, i.e., to improve the sustainability of humanity on our planet”. Hilty et al. (2006).

### 4% of the respondents selected this option.

**Definition 5:** “Software whose purpose is to support sustainability goals, that is, to improve the sustainability of humanity on our planet”. Dick et al. (2010).

### 5% of the respondents preferred this definition.

**Definition 6:** “Environment friendly software that helps improve the environment”. Murugesan (2008).

### 3% of the respondents selected this option.

We can see that definitions 2 and 3 have similar proposals since they mention the impacts of software production according to sustainability dimensions. Together, they account for 54% of the respondents’ choice, which demonstrates that the public’s view encompasses other dimensions, not just technical.

The perceived increase in the awareness and adoption of sustainable practices in the software context might be related to the recognition of sustainability as a quality goal for software systems Penzenstadler et al. (2014a). In this sense, we asked the respondents whether sustainability should or should not be considered an NFR. This question aims to understand the practitioners’ perception of sustainability as an NFR. As a result, 58% of the respondents considered that sustainability should be considered an NFR, which converges to the SE literature discussions. However, only 18% of the respondents were able to provide reasonable statements supporting their opinion. Next, we cite the respondents’ justifications.

One respondent stated that it should be considered as an NFR “because of the impacts on the environment and consequently people’s quality of life” [#2]; another respondent stated that “In software that applies the idea of resource consumption, if requested by the customer, sustainability would be considered a non-functional requirement.” [#51]; another answer was “Because it demonstrates to the target audience the concern that the company has when delivering a product. It tries to solve a problem causing the least possible environmental impact, from its conception to the delivery of the solution” [#67].

## RQ2: Sustainability importance level

In analyzing the degree to which the respondents consider that companies should give importance to the sustainability issue in the software development process, we discovered that 33% treat the issue as “important” and 44% as “very important”. For another 19%, it is “neutral” and 3% see “no importance” in the subject. We observed that, for this minority, companies do not have a process to evaluate the quality of the software and its eventual sustainability. Consequently, they see no added value in making the software development process sustainable. By crossing this data with the question “What respondents understand that sustainability represents for companies?”, we could see from **Figure 6** that most respondents – 48% – see sustainability as an opportunity to gain new business. Nevertheless, 18% of the respondents believe that sustainability in the software development process represents costs and expenses for companies. It is worth mentioning that the total amount could exceed 100% as it was a multiple-choice question.

In a broad scope, we also asked the respondents if their companies adopted any sustainability practices, such as: proper disposal and recycling of waste and batteries, compliance with environmental legislation, saving water, energy, and paper, and others. Respondents could choose one of the following answers:

- *Expert*: Meets all legislation, performs and encourages various practices.
- *Intermediate*: Meets several legislation and performs various practices.
- *Beginner*: Meets few legislation and performs some practices.
- No knowledge
- Does not comply with legislation

**Figure 7** shows that 41% of the respondents could not answer whether the company in which they operate adopts sustainable practices or meets some environmental legislation. 24% indicated that they consider the company at the beginning level since they adopt some practice and comply with few legislation rules; 20% consider that the company is at the Intermediate level, taking into account several different laws and practices. Another 9% reported that their company did not comply with any legislation; and only 6% pointed out that the company complies with all laws and encourages the adoption of various practices. Most respondents that stated the company complies with all laws are from Consulting companies. The results of this study showed that employees do not know at what level the company they work for is with regard to the adoption of sustainable practices.

Next, we asked whether their companies were concerned with minimizing the negative impacts that traditional development process activities could have on the environment. 34% of the respondents reported that the company had a reasonable concern, neither so much nor so much. For 32% of the respondents, the company did not care about such an issue. 19% reported that the company cared a bit. Another 15% were concerned about the negative impacts.

Furthermore, the following responses point to the adoption of agile methodologies as an alternative to minimize negative impacts: “I believe that agile methodologies are more sustainable than the traditional development process because

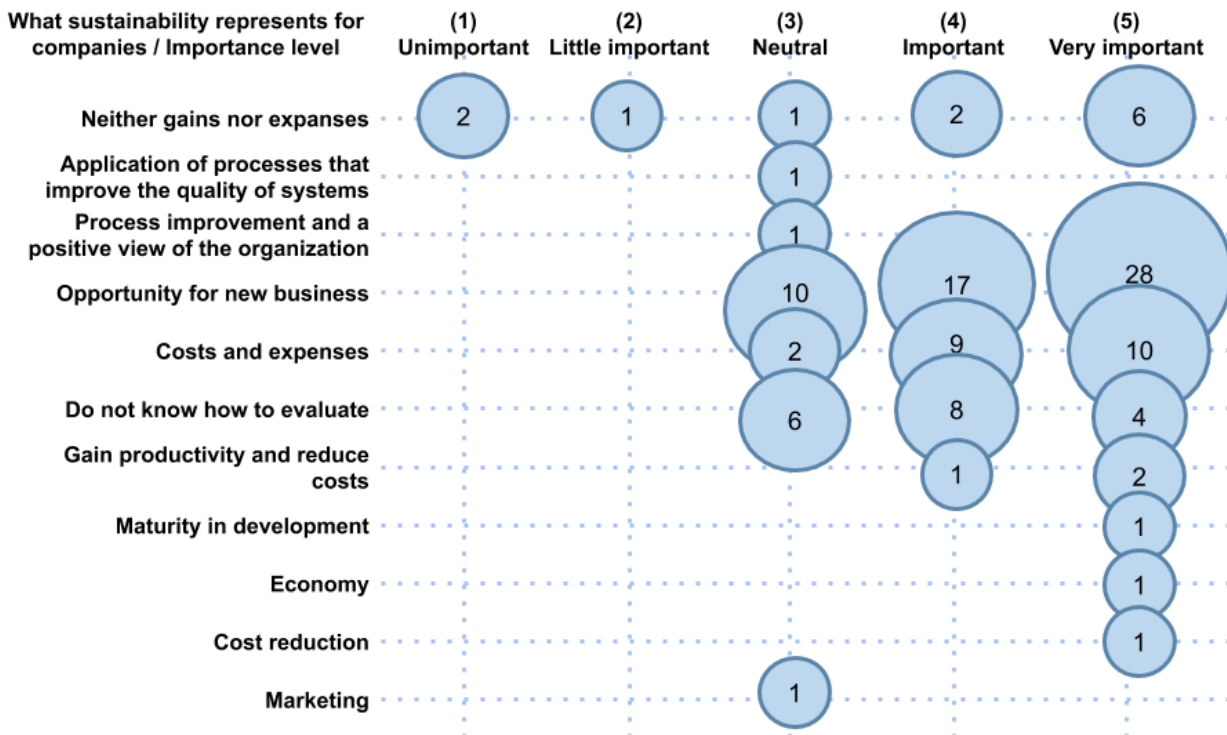


Figure 6. Company awareness level

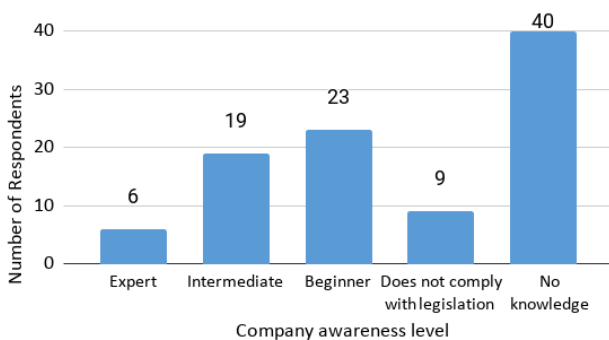


Figure 7. Distribution of respondents by company awareness level

the software is delivered in parts and in constant monitoring of the client, reducing the risk of developing software which is different from what the client expects and avoiding changes or rewriting software.” [#78]; “The company does not have a physical space and most of its employees work as a home office, as a small number of employees is concerned about the theme, the company is not concerned because its environmental impact is as little as possible.” [#90]. Regarding the data from the preliminary study, new results showed a small variation, from 1% to 3% more or less, which means the same trend in the results.

When asked about the main barriers that hinder the adoption of sustainability actions and practices in the software development process of the corporate environment, 71% of the respondents stated that there is a lack of companies awareness. Another 58% understand that companies do not consider the subject as relevant. 35% of the respondents could not evaluate; 32% responded that their companies do not have qualified staff; and 21% reported difficulties in measuring likely earnings. In the view of 21% of the respondents, bureaucracy becomes a barrier. The remaining 10% consider it

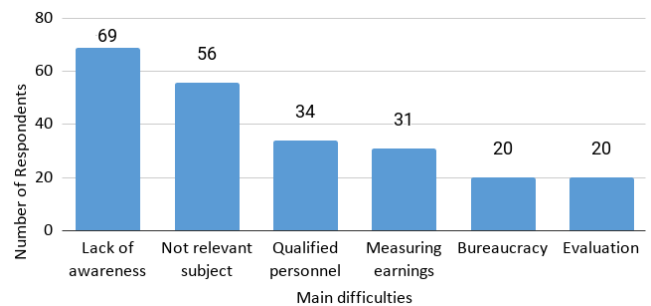


Figure 8. Main difficulties in adopting sustainable practices by companies.

as a very expensive investment, as **Figure 8** shows. Because it is a multiple-choice issue, the total ratio could exceed 100%.

### RQ3: Sustainable Software Development Process

We asked the respondents whether they felt that companies should give importance to the sustainability issue in the software development process. 44% answered that it was “Very important”. 33% considered it as being “Important”; 19% reported as “Neutral”; only 1% considered as “Less important”; and 3% did not consider the topic as important. In general, 77% of respondents think that companies should give importance or a lot of importance to sustainability in the software development process.

We sought to know what respondents think as mandatory features for a software development process to be considered sustainable. The codes obtained from this open question were mainly: *reuse, code quality, sustainable good practices* (using standards, green models and metrics), *agile methods, resource usage awareness, robust architecture, reduction of environmental impacts, efficient coding, maintainability, adaptability, accessibility, development standards, and optimized coding.*

When asked whether the companies they worked for used

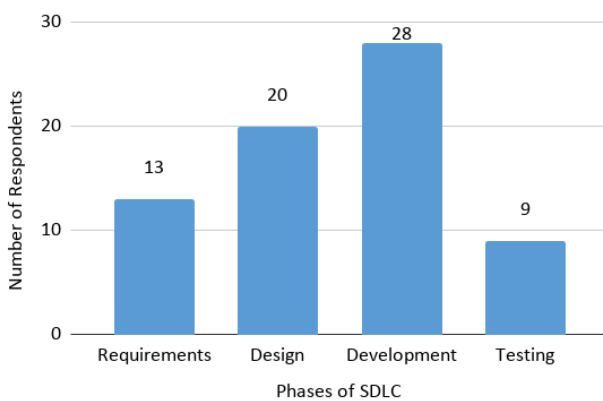


Figure 9. SDLC phases covered by sustainable practices

to encourage the adoption of sustainable practices, whether in general or specific, in the software development process, 44% were unable to answer; 27% of them stated this was a rather common practice; while other 29% reported that their companies do not encourage. Regarding the preliminary survey study, the data did not show a significant variation, from 1% to 5%, which allows us to observe a pattern in the results.

In addition, we also attempted to figure out, from the companies that encourage the use of sustainable practices, which are the covered SDLC phases. Figure 9 shows that 40% of the companies adopt such practices in the development phase; 29% in the design phase; 19% in requirements; and 13% in testing phase. The respondents were allowed to choose more than one SDLC phase.

We asked the respondents in which SDLC phases they could identify any deficiencies in terms of sustainability practices. 21% showed no deficiencies; 19% showed deficiencies for the development phase; 16% at the design phase; 15% in the requirements phase; and 13% in the testing phase. In relation to the preliminary study, the data varied 1% more or less, with the response pattern prevailing.

The identified deficiencies were related to:

- **Requirements:** Poorly designed requirements and low depth of functionality [#60].
- **Design:** Deficiency in thinking about solutions that need less space or computational power [#45], Poorly implemented code that has a high cost [#47], Limited view on the relations between the modules and systems functionalities. [#60].
- **Development:** Unstructured code [#57], Development with high coupling limited to functionality, only [#60].

In asking what could be done to improve the deficiencies pointed out in the previous question, the respondents suggested:

- **General:** “Maturity in software development from conception (requirement) to creation (implementation / coding) to have greater gain, less effort and higher quality (standardization usability, open architecture)” [#9], “Programs to encourage the study of the theme well such as broad dissemination and availability of materials that support enrichment on the subject to professionals” [#11], “Incentives and awareness” [#15], “Institutionally adopting sustainable policies to raise awareness of

people and business” [#19]. “Context of difficult change. But awareness should be the first step” [#20], “Study on the subject, understand what it means and evaluate ways to get started” [#23], “It is necessary to raise awareness of sustainable development practices and metrics for incorporation into the process of the company” [#25].

- **Requirements:** “Define techniques for assessing requirements and recording in notes for evaluation and adherence to sustainability” [#7].
- **Design:** “Think of an architecture that is sufficient to fit the software design. For example, specifying computers that spend less energy but still meet the project requirements” [#24].
- **Development:** “Develop the software with the maximum possible reuse” [#24].

#### RQ4: Sustainability dimensions

We list the contributions proposed by Lago et al. (2015), without showing their related dimension. The idea was to observe how the respondents perceived the dimensions of sustainability in their daily activities, and the importance level of each one was observed. For each feature, respondents were presented with a brief description and five unique response options.

Table 1 shows that, on average, 84% of the answers considered all characteristics as either “Important” or “Very important”, The “Very important” degree was attributed to the following characteristics: Adaptation to changes, Reusability, Performance, and System Quality. The degree “Important” was attributed to the characteristics: Longevity, Software evolution, Product roadmap, Awareness about the use of sustainable practices, Sustainable Ethics, Energy consumption, Environmental concern, Time to Market, and Development effort.

The results show that professionals consider the technical dimension as the most important one, with a mean of 93%, followed by other dimensions: Social (79%), Economic (78%), and Environmental (71%). For most respondents, the technical dimension is the most important one.

#### RQ5: Sustainability models

In a recent literature review (Mourão et al., 2018), we showed that there is not enough evidence in the literature on the use of a particular model. Most of the proposed solutions are strictly academic, with no proof of effectiveness in real environments. Therefore, in this question, we analyzed whether the professionals had adequate knowledge about the sustainable software engineering field and whether their companies apply any process model to support sustainability in SE practices. As this study is exploratory, the purpose is not to confirm the adoption of models but to obtain them from the industry that we did not identify beforehand.

The answers were: 96% of the respondents answered that they are not aware of any applied models. For the 4% of the positive responses, only two respondents specified which model the company uses for supporting sustainability: CMMI (Capability Maturity Model Integration) and EPEAT (Electronic Product Environmental Assessment Tool). While the



**Table 1.** Sustainability Dimensions Analysis

Dimensions	Sustainability concern	Irrelevant (1)	Less important (2)	Neutral (3)	Important (4)	Very important (5)
Technical	Longevity	1	1	3	<b>47</b>	45
Technical	Resilience to uncertainty	1			33	<b>63</b>
Technical	Performance	1		4	39	<b>53</b>
Technical	Software Evolution	1	1	10	<b>47</b>	38
Technical	Reusability	1	4	5	29	<b>58</b>
Technical	System Quality	1	4		24	<b>68</b>
Social	Product Roadmap	1	7	15	<b>41</b>	33
Social	Awareness	2	4	13	<b>44</b>	34
Social	Ethics	2	2	16	<b>41</b>	36
Environmental	Energy consumption	4	5	18	<b>40</b>	30
Environmental	Environmental concern	2	7	21	<b>37</b>	30
Economic	Time to Market	4	6	16	<b>38</b>	33
Economic	Development effort	1	2	13	<b>49</b>	32

CMMI helps to improve processes, the EPEAT assesses various environmental criteria of the full product life-cycle.

#### RQ6: Sustainability tools

Similar to the previous question, this one explores tools that have been adopted in practice, which collaborate with sustainability through the intuitive knowledge of the respondents. We analyzed whether the company adopts tools, techniques, or methods to measure sustainability and also if there is the adoption of some sustainable design pattern in the software development process. 64% of the respondents stated they did not know about any or did not know how to report on their use in the company. Analyzing the 36% positive responses, we could notice that the respondents use ordinary tools, techniques, or methods to improve sustainability. However, most of them did not explicitly mention which ones they use.

Regarding the economic and environmental dimensions, one respondent stated that the company adopts a process for reducing expenses through the efficient use and reuse of equipment, which avoids spending on superfluous consumption resources [#4]. Other respondents stated that the companies use agile methodologies for software development as Scrum [#65, #16], which could be related to the capacity for time and costs management when developing the software.

Regarding the technical dimension, the responses encompass aspects of software maintenance. Some respondents [#21, #55] focused on the development of reusable components that can be combined into a well-defined architecture. Other respondents stated that they use tools for automated tests [#28], continuous integration purposes (e.g., Jenkins), and for applying quality metrics (e.g., Sonar) [#16, #54]. It may help to minimize problems with legacy and also to maintain quality aspects.

Besides the other RQ's revealed that the respondents consider sustainability an important topic to be addressed in the industry, this result shows that specific tools, techniques, and methods to support sustainability are still unknown by practitioners. However, as some of the respondents tried to relate ordinary tools, techniques, and methods with sustainability,

which may indicate that they understand the importance of sustainability when developing software.

## 4 Reaching a common understanding of Sustainable Software

Based on the evidence obtained from this survey, we observed that sustainability in SE is still an incipient subject. To reach a common understanding of sustainable software, we applied the Grounded Theory (GT) method (Glaser et al., 1968) whose emphasis is on the generation of new theories. GT has a set of procedures that provide a comparative data analysis, which is able to generate, in a systematic way, a theory based on these data (Glaser et al., 1968). The result is a set of categories and relationships between them. We next describe the method steps applied: open coding and selective coding.

By following the *open coding step*, we identified the most relevant aspects of sustainability in SE from the following open question: "4.2. How do you define 'Sustainability' in the software development process?". Although we found that 91% of participants have either *no* or *low* knowledge about sustainable software, they were able to infer or assimilate the general concept. This conclusion was based on our earlier study (Karita et al., 2019). Therefore, we proposed an *initial* taxonomy with the converging points, based on the participants' common understanding. It could bring a common sense about the practitioners' perception of the topic.

Then, we proceeded with the data analysis and captured the codes. The theoretical saturation step was achieved when no new code was identified in the steps of data collection and analysis. The GT application allowed us to group these codes into categories to produce a high abstraction level. The categories were defined to the four sustainability dimensions (technical, social, environmental, and economic).

**Figure 10** shows the taxonomy created for the identified codes. In the frame of each code, we mentioned the total amount of code citations. Five new codes emerged from the extended survey, which are: *environmental awareness* (7%),

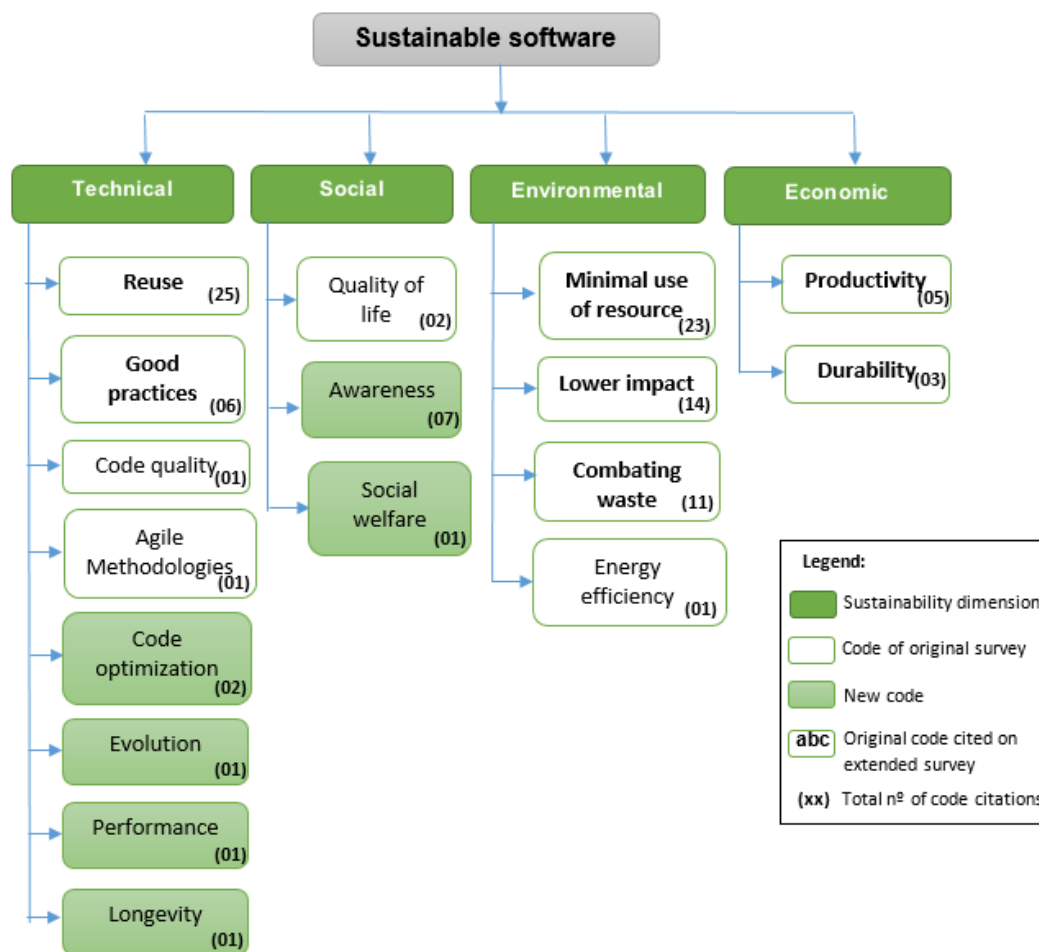


Figure 10. Sustainable software taxonomy

code optimization (2%), social welfare (1%), software evolution (1%) and added value (1%). They are present in **Figure 10** at the end of each category (light green frame). The three most cited codes that define sustainability in software are: reuse (24%), minimal use of resources (22%), and low impact (13%).

We also analyzed the relationships between codes and categories using selective coding step, as **Figure 11** shows. The findings show that, although a few knowledge gaps still exist on the subject, practitioners idealize that what drives sustainability in software is the adoption of **Reuse** in the codification phase. We founded this feature in various responses, such as: “It is a style of development of digital systems where the reuse of source code is prioritized to avoid rework” [#4] and “Reuse of source code with the creation of components to avoid rework, minimizing the use of available resources, thus enabling greater productivity.” [#9]. The second most cited feature was the **Minimal use of resources**. This feature was exploited indirectly as a consequence of adopting the practice of reuse. According to the SE community, reuse has many advantages for software development, such as increased productivity, increased software quality, decreased delivery time, etc. The cause/effect relationship between reuse and other characteristics can be seen in some answers, for example: “Sustainability collaborates with combating waste, improving quality of life, creating more durable products, recycling, etc. and this can be applied in the development

process software when practicing code reuse, for example.” [#2] and “It means to focus on reusing source code to reduce the amount of effort and resources allocated during software development, which in turn can help to reduce the impact on the environment.” [#24].

Therefore, sustainable software for the industry is related to the software produced on the adoption of reuse and development good practices. Consequently, second-order results would bring benefit to the environment, such as low impact, combating waste, and energy-efficiency. This concept is also known, in the literature, as *Green in Software*. The practitioners understand that a more sustainable way to develop sustainable software is using practices that apply SE principles, taking into account environmental aspects. This perspective distorts the idea that software development has effects that go beyond its boundaries.

## 5 Discussion

In this section, we discuss the results in the light of collected data, based on the set of analyzed sustainability dimensions.

### • Technical dimension

According to Penzenstadler et al. (2014b), the technical dimension has a central interest in the requirements related to the software **longevity** and **evolution**, such

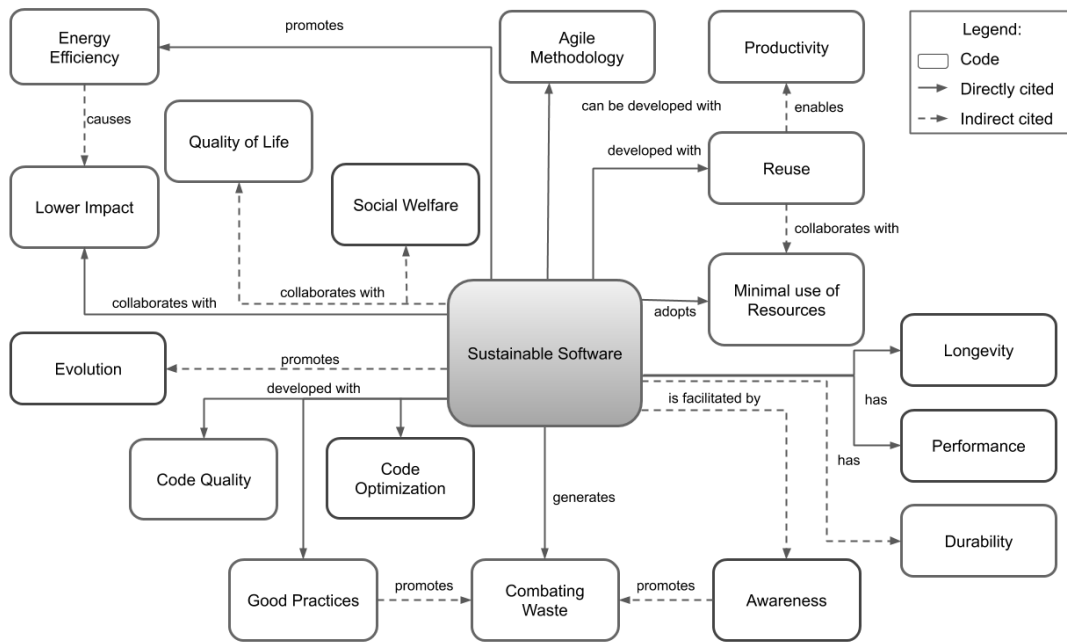


Figure 11. The relationship of leveraged codes.

as non-obsolence and quality characteristics. Both requirements were only cited in the extended study. The study confirmed that software practitioners have a narrow perception of sustainability concepts in the software development process. This is because most practitioners have targeted their perceptions about sustainable software specifically in the quality attribute, such as **reuse**, **optimization**, and **performance**. This skewed view of sustainability covers only one of the five dimensions defined in the literature, the technical dimension, and confirms the results presented in Lago et al. (2015).

In terms of the software development processes, we could see that companies could not yet be considered green companies or aspiring to be sustainable companies because they do not use models, processes, methods, and tools to support their software development. Although the professionals do not have in-depth knowledge of the subject, they could see the advantages and importance of sustainability in software development.

The adoption of agile methodologies is another point of discussion. We observed that this topic is relatively new for the surveyed companies. Despite the various benefits that agile software development could offer, such as the development of the minimum viable product in a short production cycle, its interaction with sustainability is a gap that needs to be explored.

• **Social dimension**

The social dimension refers to the effects of software systems in society (e.g., product roadmap, ethics, etc.). In this sense, the study showed that the number of professionals who perceive the impact of sustainability on people’s **quality of life** and **welfare** is still low.

Although the **awareness** was firstly mentioned in this study as a social concern, similarly to the preliminary

survey result, we could observe that all participants in the software’s production process need to create a critical sense in relation to the negative impacts that software production could cause on the planet. Based on this understanding, the industry would have professionals engaged in providing sustainability.

Achieving a sustainable software development environment is possible. However, it is very important to encourage software teams to employ sustainability practices, thus considering existing tools, methods, and processes support, as well as proposing new ones.

According to Lago and Penzenstadler (2017), conducting interviews is another way of creating awareness since the results contribute to social sustainability. Additionally, practitioners need to think about sustainability in all spheres of software development, not only from a technology perspective. Something has been said about code reuse, maintainability, efficiency, but awareness goes beyond technical bias. The four dimensions interrelate and need to happen in an integrated way so that sustainability could happen in all stages of the software development process, from the customer’s need to the customer satisfaction.

Therefore, all dimensions could be better disseminated so that greater compliance could be achieved by companies and especially by people. In this way, we could attract conscious and sustainable software companies.

• **Environmental dimension**

In this dimension, our purpose was to obtain evidence of how professionals perceive the impacts of software development and maintenance in the environment.

According to Penzenstadler et al. (2014b), environmental sustainability could be achieved by analyzing the software development life-cycle and assessing the envi-

ronmental impact it could cause.

Concerning legislation, some Brazilian laws aimed at sustainability were mentioned in the study. However, what could be observed is that environmental issues focused on the environment, such as waste recycling, water saving, are still seen as the main factors associated with the term sustainability by these companies.

Despite the practitioners' low knowledge on the subject, the participants attributed to this a high importance. In the software bias, this dimension is directly related to energy consumption and environmental interests (Lago and Penzenstadler, 2017). Most professionals reported that their companies do not have quality requirements related to sustainability. This insight reinforces the need for the research community to increasingly join the effort to make sustainability a software quality requirement.

Through this study, it was possible to observe that understanding the homogenization of concepts used in this area is still uncertain. For software to be produced sustainably, software professionals must agree on the inherent concepts from this domain and its properties, so that they could have a clear and shared understanding of environmental knowledge and concern. We understand that it is important for practitioners to understand the central pillars of sustainability so that they could have a broader understanding of their likely effects.

- **Economic dimension**

The economic dimension is one of the main concerns of companies. It is related to market requirements such as budget and cost restrictions (Raturi et al., 2014; Penzenstadler et al., 2014b).

Regarding this category, the few codes classified on it were indirectly mentioned. This perception goes against the findings presented by Lago and Penzenstadler (2017).

For professionals, sustainable software development creates an additional effort of development, and current projects do not foresee this type of cost to implement sustainable software. We also noticed that companies do not promote sustainable development, which could encompass hiring qualified people with a good understanding of software engineering principles. Therefore, there would be more time and resources to design and develop software with the expected quality associated with sustainable requirements.

Another aspect that permeates the economic dimension has to do with customer satisfaction (Groher and Weinreich, 2017). In this sense, few participants mentioned this factor. Only 3 reported that sustainability is important, but it does not interfere with customer service functions. Therefore, it must be a product obligation, a requirement on the part of the customer.

In light of these discussions, we believe that companies should incorporate investments in business decisions to produce more sustainable software and implement sustainable Software Engineering practices. The lack of companies' vision of not exploring sustainability practices in software production causes them to reduce their competitive advantage.

However, for industry, the expectation of return on these investments is still a gap. From an economic point of view, this gap makes the issue an urgent and strategic concern.

In general, the extended study confirmed that the practitioners' perception of all dimensions of sustainability is subtle. It could be better worked together, not just for the technical direction. Therefore, those four factors need to be integrated into practice so that sustainability actually occurs within the scope of software production.

The knowledge of software professionals needs to be expanded in all dimensions concerns, such as: knowing that software production has environmental impacts, accessing information, tools, methods, transferring knowledge into actions, and raising awareness of these issues around them.

## 6 Implications for Research and Practice

In this section, we provide readers with a synthesis of the relevant implications that emerged from the analysis of this qualitative study:

- Green SE field is still incipient, and it needs to be disseminated in companies so that teams can start thinking in a sustainable way about software development;
- There is a lack of professionals' knowledge about the topic, in particular regarding how to adopt sustainable practices in the SDLC;
- Although software professionals have limited or no knowledge about sustainability in the context of software development, they realize that its adoption has benefits for both company and society;
- The technical and environmental dimensions are the most relevant and explored ones. Most practitioners have targeted their perceptions about sustainable software in "Green in Software". They understand that a more sustainable way to develop sustainable software is using practices that apply SE principles, taking into account environmental aspects. This perspective distorts the idea that software development has effects that break its boundaries. In this sense, it is important to analyze the role of software and investigate the impacts of its use in society in all dimensions of sustainability.
- Since there is not enough evidence in the literature on the use of a particular tool and model - most reports are strictly academic in character, without proof of effectiveness in real environments (Mourão et al., 2018) - this study showed that companies not yet use models, processes, methods and tools to support sustainable software development. As such, they cannot be considered green companies or aspiring sustainable companies.

## 7 Related Work

We next discuss recently published studies that are related to our work.

A survey conducted with fifty-three software professionals in seven different companies was reported by Koçak et al. (2015). The goal was to identify the perception of software professionals about the impact of energy quality related software in order to develop an environmentally sustainable software product. Through this research, the authors explored the correlation between software quality and energy efficiency. They used statistical analysis. The results of this study showed that there are significant negative correlations between functional adequacy and compatibility; efficiency and safety of performance; reliability and compatibility with regard to energy efficiency.

Manotas et al. (2016) performed the first empirical study on how professionals think about energy when writing requirements, design, construct, test, and maintain their software. The authors reported the findings of a quantitative and targeted survey of 464 professionals from the companies ABB, Google, IBM, and Microsoft. This research was motivated and supported by qualitative data from 18 detailed interviews with Microsoft employees. The study concluded that Green SE practitioners take care and think about energy when building their applications. The results show that awareness has changed the discussion about software power consumption. In relation to the awareness stimulus, the authors agree that appropriate support such as the creation of organizational policies and knowledge banks could help to create green software products.

Pang et al. (2016) conducted a survey with 122 programmers to understand their knowledge and awareness regarding software energy efficiency and consumption. The results show that the programmers' knowledge about energy consumption is consistent, and 60% of them consider energy consumption when choosing a mobile development platform. However, 80% of the programmers do not take energy consumption into account when developing software.

Jagroep et al. (2017) reported a multi-core study incorporated with two over two commercial software products. The goal was to identify how to create and maintain awareness of an energy consumption perspective for software among stakeholders involved in the development of software products. During the study, they followed the development process of two commercial software products and provided direct feedback to stakeholders on the effects of their development efforts, specifically on energy consumption and performance, using a power control panel. The authors defined a main research question and three sub-questions. To measure awareness, the authors constructed a survey but did not report the details of the planning, target audience, and instrument.

To understand how software sustainability is currently addressed in the practice of software development projects, Groher and Weinreich (2017) conducted an interview with 10 software project team leaders from 9 companies in Austria. The study analyzed the data using the deductive categorization method. The study found that professionals consider software sustainability important, but are technically concerned with sustainability. Organizational and economic issues are addressed, but environmental considerations are lacking. The perceived influence of various project factors on sustainability is partially diverse, suggesting that the meaning of sustainability needs to be refined to the specific context of design

and application.

Pinto and Castor (2017) conducted a survey with software developers in order to understand the perceptions of those professionals on issues related to energy consumption by the software. The authors interviewed 62 software developers who performed at least one commit on an open-source mobile application. The results of the study suggested that there is a lack of knowledge about how to develop energy-efficient software. In addition, they noted that there is a need for tools to help developers achieve this goal.

In order to develop this work, we considered every mentioned study since they bring relevant information on the topic. However, we observed that these studies were usually focused on particular issues, such as the correlation between sustainability and software quality attributes, the energy use in software applications. As the research in this field is incipient, it becomes important to explore the software professionals' perception with broader coverage.

## 8 Threats to Validity

*Construct Validity:* During the pilot test, some respondents reported that the filling time of the instrument was extensive. As such, our survey respondents may not have adequately answered questions, preferring short answers to more detailed descriptions. To reduce the threat to validity, we group the questions into specific sections to better target questions and answers. Another threat was the respondents' understanding of the questions. To help ensure the understandability of the survey, we asked professionals and researchers with experience in SE and experience in survey design to review the survey instrument to ensure the questions were clear and complete.

*Internal Validity:* An internal limitation may be the selection of companies and practitioners to the sample. We understand that both the number of companies and the number of responses obtained may not adequately represent the entire population of companies and software professionals, characterizing a threat to internal validity. However, as we decided to include only professionals from companies that work in different domains (mostly have offices in several Brazilian states), we believe this set might be representative.

*External Validity:* The respondents of our survey may not adequately represent all software practitioners. Most respondents reported that they work as software developers, which may have skewed the results. Nevertheless, we believe that the number of responses that we analyzed provides a rich source of qualitative data that could reveal promising insights.

*Reliability:* Although Grounded Theory offers rigorous data collection procedures and analysis, qualitative research is generally subject to researcher bias. Certainly, other researchers could make a different interpretation and theory after analyzing the same dataset, but it is believed that at least the main insights would be preserved. Then, the results of RQ1 might be a threat to validity. However, to mitigate it, the qualitative analysis was performed on the codes recovered and grouped according to the correlations between dimensions of sustainability and sustainable concerns proposed by Penzenstadler et al. (2014b) and Raturi et al. (2014).

Although the research results may have been influenced by interpretation, to mitigate this threat, the coding process was performed by two authors working together. Disagreements in the assignment of codes were discussed until consensus was reached.

## 9 Concluding Remarks

Although the SE community has increased its interest in the Green and Sustainable SE field, the software industry has not explored this area in an adequate fashion yet. Consequently, Green and Sustainable practices are not completely known and substantially applied by software practitioners.

This study is an extended survey from a previous work designed to gather data from software practitioners from Brazilian companies in this respect, and provide data on the software industry's perception of sustainability in the software development process. The yielded results confirm the findings identified in the original survey. They indicate an overall lack of knowledge about the topic, in particular regarding the concepts about sustainable software, although there is a common understanding that sustainability should be treated as a quality attribute and should support the interaction between sustainability and the SDLC.

This study contributes to the field with an initial set of evidence. We could see it as an important step towards establishing a common understanding of how the software industry is receptive to sustainability concepts in software development practices. As future work, we aim to conduct interviews with participants of the survey, in order to enrich and detail the professionals' perceptions. In addition, we plan to carry out more in-depth studies about already validated techniques and methods that could improve and compose a green checklist for software development.

## Acknowledgements

This research was partially funded by FAPESB grants JCB0060/2016 and BOL0188/2020, INES 2.0, CNPq grant 465614/2014-0, and CAPES - Finance Code 001.

## References

- Becker, C. (2014). Sustainability and longevity: Two sides of the same quality? *mental*, 20:21.
- Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., and Venters, C. C. (2015). Sustainability design and software: The karlskrona manifesto. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 467–476. IEEE.
- Calero, C. and Bertoa, M. (2013). 25010+ s: A software quality model with sustainable characteristics. *Sustainability as an element of software quality. Green in Software Engineering Green by Software Engineering (GIBSE 2013) co-located with AOSD*.
- Calero, C. and Piattini, M. (2015a). *Green in software engineering*, volume 3. Springer.
- Calero, C. and Piattini, M. (2015b). *Introduction to Green in Software Engineering*, pages 3–27. Springer International Publishing, Cham.
- Calero, C. and Piattini, M. (2017). Puzzling out software sustainability. *Sustainable Computing: Informatics and Systems*, 16:117–124.
- Dick, M., Naumann, S., and Kuhn, N. (2010). A model and selected instances of green and sustainable software. In *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, pages 248–259. Springer.
- Erdelyi, K. (2013). Special factors of development of green software supporting eco sustainability. In *2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 337–340. IEEE.
- Fernandez, N. C., Lago, P., and Calero, C. (2016). How do quality requirements contribute to software sustainability? In *REFSQ Workshop*, pages 7–10.
- García-Mireles, G. A., Moraga, M. Á., García, F., Calero, C., and Piattini, M. (2017). Interactions between environmental sustainability goals and software product quality: a mapping study. *Information and Software Technology*.
- Glaser, B. G., Strauss, A. L., and Strutzel, E. (1968). The discovery of grounded theory; strategies for qualitative research. *Nursing research*, 17(4):364.
- Groher, I. and Weinreich, R. (2017). An interview study on sustainability concerns in software development projects. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 350–358. IEEE.
- Hilty, L. M., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., and Wäger, P. A. (2006). The relevance of information and communication technologies for environmental sustainability—a prospective simulation study. *Environmental Modelling & Software*, 21(11):1618–1629.
- Imperatives, S. (1987). Report of the world commission on environment and development: Our common future. Accessed Feb, 10. Available at <http://www.un-documents.net/our-common-future.pdf>.
- Jagroep, E., Broekman, J., Van Der Werf, J. M. E., Lago, P., Brinkkemper, S., Blom, L., and Van Vliet, R. (2017). Awakening awareness on energy consumption in software engineering. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, pages 76–85. IEEE.
- Johann, T., Dick, M., Kern, E., and Naumann, S. (2011). Sustainable development, sustainable software, and sustainable software engineering: an integrated approach. In *Humanities, Science & Engineering Research (SHUSER), 2011 International Symposium on*, pages 34–39. IEEE.
- Karita, L., Mourão, B. C., and Machado, I. (2019). Software industry awareness on green and sustainable software engineering: a state-of-the-practice survey. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 501–510, Salvador, Brasil.
- Kasunic, M. (2005). Designing an effective survey. Technical Report CMU/SEI-2005-HB-004, CarnegieMellon University, Software Engineering Institute, PA, USA.
- Khandelwal, B., Khan, S., and Parveen, S. (2017). Cohesive

- analysis of sustainability of green computing in software engineering. *International Journal of Emerging Trends Technology in Computer Science*, 6:11–16.
- Kitchenham, B. A. and Pfleeger, S. L. (2002). Principles of survey research part 2: designing a survey. *ACM SIGSOFT Software Engineering Notes*, 27(1):18–20.
- Koçak, S. A., Alptekin, G. I., and Bener, A. B. (2015). Integrating environmental sustainability in software product quality. In *RE4SuSy@ RE*, pages 17–24.
- Lago, P., Koçak, S. A., Crnkovic, I., and Penzenstadler, B. (2015). Framing sustainability as a property of software quality. *Commun. ACM*, 58(10):70–78.
- Lago, P. and Penzenstadler, B. (2017). Reality check for software engineering for sustainability—pragmatism required. *Journal of Software: Evolution and process*, 29(2):e1856.
- Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspán, C., Sadowski, C., Pollock, L., and Clause, J. (2016). An empirical study of practitioners’ perspectives on green software engineering. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 237–248. IEEE.
- Moraga, M. Á., García-Rodríguez de Guzmán, I., Calero, C., Johann, T., Me, G., Münzel, H., and Kindelsberger, J. (2017). Greco: Green code of ethics. *Journal of Software: Evolution and process*, 29(2):e1850.
- Mourão, B. C., Karita, L., and Machado, I. C. (2018). Green and sustainable software engineering—a systematic mapping study. In *Proceedings of the 17th Brazilian Symposium on Software Quality (SBQS)*, pages 121–130. ACM.
- Murugesan, S. (2008). Harnessing Green IT: Principles and Practices. *IT Professional*, 10(1):24–33.
- Naumann, S., Dick, M., Kern, E., and Johann, T. (2011). The greensoft model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems*, 1(4):294–304.
- Pang, C., Hindle, A., Adams, B., and Hassan, A. E. (2016). What do programmers know about software energy consumption? *IEEE Software*, 33(3):83–89.
- Penzenstadler, B. (2014). Infusing green: Requirements engineering for green in and through software systems. In *RE4SuSy@ RE*, pages 44–53.
- Penzenstadler, B. and Femmer, H. (2013). A generic model for sustainability with process-and product-specific instances. In *Proceedings of the 2013 workshop on Green in/by software engineering*, pages 3–8. ACM.
- Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H., and Franch, X. (2014a). Systematic mapping study on software engineering for sustainability (SE4S). In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 14. ACM.
- Penzenstadler, B., Raturi, A., Richardson, D., and Tomlinson, B. (2014b). Safety, security, now sustainability: The non-functional requirement for the 21st century. *IEEE software*, 31(3):40–47.
- Pinto, G. and Castor, F. (2017). Energy efficiency: A new concern for application software developers. *Communications of the ACM*, 60(12):68–75.
- Raturi, A., Penzenstadler, B., Tomlinson, B., and Richardson, D. (2014). Developing a sustainability non-functional requirements framework. In *Proceedings of the 3rd International Workshop on Green and Sustainable Software*, pages 1–8. ACM.
- Saputri, T. R. D. and Lee, S.-W. (2016). Incorporating sustainability design in requirements engineering process: A preliminary study. In *Asia Pacific Requirements Engineering Conference*, pages 53–67. Springer.
- Tate, K. (2005). *Sustainable software development: an agile perspective*. Addison-Wesley Professional.
- Venters, C., Lau, L., Griffiths, M., Holmes, V., Ward, R., Jay, C., Dibsdale, C., and Xu, J. (2014). The blind men and the elephant: Towards an empirical evaluation framework for software sustainability. *Journal of Open Research Software*, 2(1).
- Venters, C. C., Seyff, N., Becker, C., Betz, S., Chitchyan, R., Duboc, L., McIntyre, D., and Penzenstadler, B. (2017). 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)*, pages 3–12. IEEE.