Software industry awareness on green and sustainable software engineering: a state-of-the-practice survey

Leila Karita
leila.karita@ufba.br
Federal University of Bahia (UFBA)
Salvador, BA, Brazil

Brunna C. Mourão
brunna.caroline@ufba.br
Federal University of Bahia (UFBA)
Salvador, BA, Brazil

Ivan Machado
ivan.machado@ufba.br
Federal University of Bahia (UFBA)
Salvador, BA, Brazil

ABSTRACT

Sustainable computing is a rapidly growing research area spanning several areas of computer science. In the software engineering field, the topic has received increasing attention in recent years, with several studies addressing a range of concerns. However, few studies have demonstrated the awareness of software practitioners about the underlying concepts of sustainability in the software development practice. In this effect, in this study, we aim to provide some evidence about the practitioners' perception about the adoption of sustainability in software development, under four main perspectives: economic, social, environmental and technical. To accomplish such a goal, we carried out a survey study with twenty-five software engineers involved in projects in different domains. The yielded results indicate an overall lack of knowledge about the topic, in particular regarding the concepts about sustainable software, although it is a common understanding that sustainability should be treated as a quality attribute and should support the interaction between sustainability and the software development life cycle phases. Among the observed perspectives, the respondents indicate that the technical dimension is the most relevant and explored so far. This study contributes to the field with initial evidence and can be seen as a first step towards establishing a common understanding about how the software industry is receptive to the use of sustainability concepts in software development practices.

CCS CONCEPTS

• **General and reference** → **Surveys and overviews**; **Empirical studies**; • **Social and professional topics** → *Sustainability*.

KEYWORDS

Sustainable Software Engineering, Survey, Empirical Study.

ACM Reference Format:
Leila Karita, Brunna C. Mourão, and Ivan Machado. 2019. Software industry awareness on green and sustainable software engineering: a state-of-the-practice survey. In *XXXIII Brazilian Symposium on Software Engineering (SBES 2019), September 23–27, 2019, Salvador, Brazil*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3350768.3350770

SBES 2019, September 23–27, 2019, Salvador, Brazil

ACM ISBN 978-1-4503-7651-8/19/09…$15.00
https://doi.org/10.1145/3350768.3350770

1 INTRODUCTION

According to the United Nations World Commission on Environment and Development [1], sustainable development is compromised to meet the needs of today's human beings without harming the planet's ability to meet future generations. With the advent of Information and Communication Technology (ICT), people started to use software systems to facilitate their daily activities. However, this same technology has promoted the emergence of issues that need to be discussed, since these same resources could help to generate negative impacts in the environment.

Such a scenario is correctly discussed by Calero and Piattini [3], the impact of technology on our day-to-day lives should be seen from two perspectives. On the one hand, technology helps organizations address environmental issues when providing virtual meetings, materialization of activities, improvements in logistics, intelligent transportation systems, more efficient processes, etc. On the other hand, technology itself is often responsible for environmental degradation by consuming amounts of energy through engineering processes used to make products, for example.

Aligned with this concern, the Software Engineering community has increased its interest in unveiling the impacts of software on natural resources consumption. As far as the software development process is concerned about such an issue, it is still a new practice for engineers and software developers. Pinto and Castor [21] state that "Development currently do not fully understand how to write, maintain and evolve energy-efficient software applications".

The results of a recent literature review indicated a gap in industry participation in research [17]. The interests of the academia need to be aligned with the interests of the industry for the benefit of all. This perception is not yet clear in the literature, which could lead to confusion and misunderstandings about what "sustainable software development" means in practice. As a way of attacking that issue, this study aims to leverage the state-of-the-practice in Green Software development, under four main perspectives: economic, social, environmental and technical. To accomplish such a goal, we carried out a survey study with twenty-five software engineers involved in projects from different domains. Specifically, we are interested in the practitioners' opinion about several questions regarding the use of sustainable practices in software development. In addition, we aim to provide readers with an overview of the main discussions about software sustainability concepts.

To conduct the study we sent an online survey to fifty software practitioners. The survey received twenty-five responses that met our criteria for analysis. We adopted a systematic qualitative analysis approach and we built a coding scheme on the definitions of "sustainability" in the software development process based on the yielded results.

Our study confirms the gap identified in the literature review [17]. The survey showed that sustainability in the context of software is a novel concern for software practitioners. Respondents showed a low perception of the impacts of sustainability throughout the development life-cycle. However, despite the low awareness, we observe a high interest in this subject, being common the understanding that sustainability must be treated as an quality attribute. The evidence also shows that the technical dimension is more explored even than in either an unconscious or subtle fashion.

The remainder of this paper is organized as follows. Section 2 presents underlying concepts about green and sustainable software. Section 3 introduced the research questions defined for this research. Section 4 provides the research design of our study. Analyses and results are given in Section 5. Section 6 provides an in-depth discussion on the yielded results. Implications for research and practice is subject to discussion in Section 7. Section 8 discusses related work. Section 9 presents threats to validity, and Section 10 draws concluding remarks.

2 BACKGROUND

The terms Green and Sustainable software have been used interchangeably in recently published Software Engineering surveys. When one considers only the terms, without considering the semantics, both could be easily understood. However, when trying to apply it in order to provide readers with meaningful concepts, it could lead to a big misunderstanding. To get an idea of the complexity involved here, we describe some of the existing definitions:

Dick et al. [4] state that "sustainable software is software whose impact on the economy, society, humans and the environment resulting from the development, deployment and use of the software is minimal and / or has a positive effect on sustainable development".

Hilty et al. [8] and Penzenstadler et al. [20] in turn, agree that the term "sustainable software could be interpreted in two ways: (1) software code being sustainable, agnostic on purpose or (2) the purpose of software is to support sustainability goals".

An important requirement for sustainable software is the definition of a sustainable development process, which should consider environmental impacts during the software life cycle and involves the pursuit of sustainable development goals. In this effect, Sustainable Software Engineering could be defined as the art of developing sustainable software through a sustainable software engineering process. A process where software products are developed in a way that mitigates the negative impacts, which result or are expected to result in a software product that throughout its life cycle is continuously evaluated, documented and used for additional optimization of the software product [18].

Tate [23] defines Sustainable Software Engineering as a development which is able to make a balance between rapid release and long term sustainability. Johann et al. [10] explain the goal of Green and Sustainable Software as "the enhancement of SE which targets the direct and indirect consumption of natural resources and energy as well as the aftermath which are caused by software system during their entire life cycle".

The United Nations World Commission on Environment and Development [1] state that sustainable development takes place in three aspects: economic, social and environmental, increasing as the intersection of these increases. The social dimension encompasses people and their living conditions, such as education, health, violence, leisure, among other aspects. The environmental dimension refers to the natural resources of the planet and the way they are used by society. The economic dimension is related to the production, distribution and consumption of goods and services.

In addition, Calero and Piattini [2] have been associated to software sustainability the technical dimension. To Penzenstadler and Femmer [19], "When looking at (software) systems, we need technical sustainability as an additional dimension" and claims that "From a point of view of (software) systems engineering, technical sustainability has the central objective of long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements".

Saputri and Lee [22] state that "it is important to take different dimensions of sustainability into account. Sustainability should be considered as an integrated concept".

All of these dimensions could be analyzed from various perspectives. The choice of approach to take will take into account the purpose and scope to be investigated.

3 RESEARCH QUESTIONS

In this study, we are motivated to analyze sustainability in the context of software from the software professionals view point. In this sense, our objective is to identify the awareness of software professionals on the theme "Sustainability". To gain this understanding, the following research questions were formulated:

RQ1: Are the professionals familiar with the concepts of sustainability applied to the software development process? This question aims to investigate and determine if and at what level the professionals are familiar with concepts related to sustainability in the context of software.

RQ2: How important is software sustainability to practitioners? This question aims to investigate if and at what level professionals consider sustainability as an important factor in the software development process.

RQ3: What phases of the software development life cycle (SDLC) do sustainable practices apply? This question seeks to identify to which SLDC phase(s) the developers have adopted any Green SE practices.

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 [14].

RQ5: What models for sustainable software development have been adopted by the software industry? This question seeks to investigate whether and what models for sustainability in software have been adopted by professionals.

RQ6: What tools have been used to support sustainability in the software development process? This question seeks to investigate which software sustainability tools are used by professionals in the software development process.

4 RESEARCH METHODOLOGY

Since the six RQs are geared towards gathering the opinions of practitioners, we chose a survey as our research instrument. The remainder of this section describes the survey design, the participant selection criteria, pilot testing, data collection and qualitative data analysis.

4.1 Survey

Our goal in designing the survey was to keep it as brief as possible while still collecting all relevant information. Our research included questions to understand practitioners' motivations and knowledge regarding the topic, green practices in software development and understand the level of awareness and application of green concepts by companies.

We conducted a survey study within companies from Salvador, Brazil, with twenty-five professionals. These participants are part of software development teams in seven software companies selected by authors' convenience.

This section encompasses the planning details, execution procedures, and reporting of desired and achieved results. We used the methodology proposed by Kasunic [11] and applied the research survey principles defined by Kitchenham and Pfleeger [12]. Figure 1 shows the methodological steps employed in this study.

4.2 Identify and characterize the target audience

To ensure valid results, we only selected professionals with enough experience in software development processes. Three criteria were considered:

(1) Analysis of his/her profile in terms of experience in the Software Engineering field.
(2) Analysis of his/her 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.
(3) Whether the respondent works in one of the selected companies.

4.3 Questionnaire Design

We specified six groups of information to explore on the instrument. They were: Characterization of the respondent, Company characterization, Research object, Company development process, Difficulties encountered and Sustainability as an attribute of quality. The following describes the goal defined for each category.

- **Respondents characterization:** In this category the goal was to investigate the respondent profile, with information about gender, name, age group, level of education and length of professional experience.
- **Companies characterization:** In this category, the goal was to investigate 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.
- **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:** In this category the goal was to investigate the likely benefits expected regarding the application of sustainability in the software development processes.
- **Sustainability as a quality attribute:** In this category the goal was to investigate the interviewee's perception regarding the importance of using sustainability as a quality attribute in their projects.

4.4 Pilot test questionnaire

To help ensure the understandability of the survey, we asked professionals and researchers with experience in Software Engineering, and experience in survey design to review the survey to ensure the questions were clear and complete. The feedback only suggested minor edits. The changes we made include: adding more answer choices to several questions, exchange words to improve understanding and change the order of some questions.

The questionnaire was applied on November 19, 2018 to five employees of a software development team of a public organization. The goal was to remove any misunderstanding and hence improve the instrument. Based on the improvements identified, a second version of the instrument was generated, which also included the inclusion of new open questions.

It was still evaluated whether the research objectives are significant and of interest to the respondents of the pilot test, how interested the respondents are (in person) about the results of the survey, how long the respondents take to complete the questionnaire, how the interviewees felt about the length of time to fill out the questionnaire and overall satisfaction with the research process. The questionnaire applied at the end of the test contained the following questions:

(1) Does the questionnaire contain anything expected to reach our goals?
(2) Does the questionnaire contain any undesirable or unnecessary information to the context and purpose of the research?
(3) Were you able to properly understand the questions?
(4) Is there an error or inconsistency in the questionnaire?

4.5 Distribute the questionnaire

On November 20, 2018, we sent each of the fifty software practitioners in our list a personalized email mentioning the link to the survey hosted on Google Forms. On November 26 and December 03, 2018, we sent a reminder email. We closed the survey on January 04, 2019.

A brief introduction was made with basic information about the purpose of the study, justification of choice and importance of the

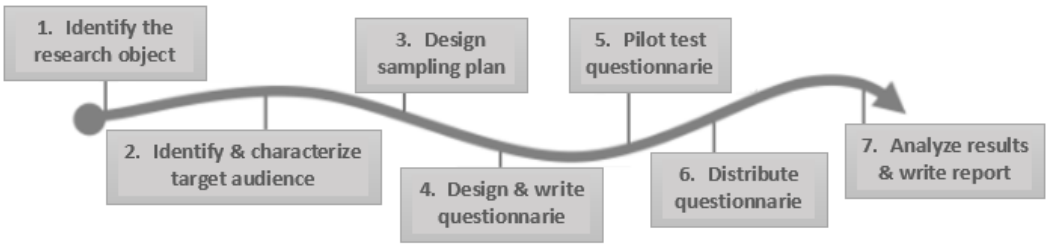

Figure 1: Survey design.

respondent's participation. Participants were also informed about the privacy policies of the study in a clear and detailed manner.

4.6 Analyze results and write report

In this section, we report the results of the analyzes of our research study. This research has an exploratory characteristic with a qualitative approach. To achieve the defined objectives, we adopted the following assumptions about the instrument:

(1) For closed questions that could combine multiple responses, the sum of percentages could be greater than 100%.
(2) For the 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, coding was applied. 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, to help clarify the results, we include excerpts from the qualitative answers. 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 respondent's answer number 1.

5 RESULTS

In this section, we report the results of our survey study. The number before the parentheses represents the number of responses and the number in parentheses represents the corresponding percentage value.

We obtained twenty-five answers for analysis. Some facts are worth mentioning from this survey application: 22% of the questions of instrument were optional; 76% of respondents answered all the questions; and 24% omitted some of the questions; only one respondent did not meet the defined characterization criteria (by not working in a local company) and her data was then excluded from the final analysis.

5.1 Respondents' Demographics

This section describes the demographics of the respondents. We investigated their age, education level and experience time, in an attempt to draw the profile of the observed sample. Overall, regarding their gender, 64% of the respondents were men and 36% were women. About their age, 8% had between 25 and 29 years, 20% had between 30 and 34 years, 32% had between 35 to 39 years, 24% has between 40 to 44 years, 12% had between 45 to 49 years and 4% has more than 55 years. More than 50% of the respondents are concentrated mainly in the 35 to 44-year-old range.

In terms of their professional experience in software development, 8% had up to 3 years of experience, 4% had between 4 and 6 years, 12% had between 7 and 10 years and 76% had more than 10 years of experience in industry.

Finally, the respondents played many different roles during their careers: 35% had already worked as a system analyst, 35% had already worked as a project leader, 31% had already worked as a requirements analyst, 27% had already worked as a software developer, 19% had already worked as a business analyst, 15% had already worked as a software architect, 15% had already worked as a project management, 12% had already worked as a porftolio management, 8% had already worked as a software tester, 4% had already worked as a product owner, and 4% had already worked as a designer. In this question, the respondents could answer more than one option.

5.2 Companies' Demographics

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

Respondents worked in companies of different segments: 36% in software factories, 32% in government companies and 8% in Research and Development Centers. The others add up to 20% working in other segments. About company size, 88% of the respondents reported that the size of the company is "large", that is, it has more than 99 employees.

5.3 Research object

We next show the results of this empirical evaluation, based on the set of research questions previously stated in Section 3.

5.3.1 RQ1: Sustainability concepts.

In this section, the goal is 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 level of knowledge of the respondents, we asked how respondents could self-assess their level

of knowledge about Sustainability in the software development process. 52% out of the respondents had low knowledge about the subject, 44% had no knowledge and that this was the first contact with the subject. 4% had a medium knowledge of the subject.

We then asked respondents to define sustainable software. The question was open and we applied the coding on the results. We identified the codes and after discussion between two of the authors, we group them into the four dimensions of sustainability (technical, social, environmental and economic). The category system is illustrated in Figure 2.

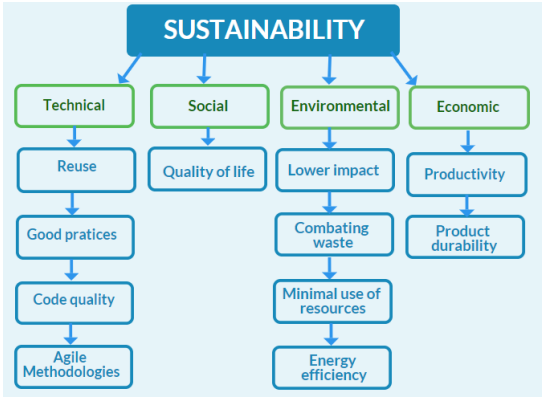

Figure 2: Coding RQ1.

In another question, we have listed six concepts about "sustainable software" available in the literature of relevant authors in domain. The respondents did not have access to authors' name and could choose only one option. Our objective was to identify with which of these concepts the respondents would be more familiar. The results are described next. We provided the description of each author in boxes and presented the corresponding results.

> *"An application that produces as little waste as possible during its development and operation". [5]*

48% of the respondents consider this the most coherent definition.

> *"Software developed and used in such a way that leaves minimal negative impact on users, environment, economy and society in general". [18]*

24% of the respondents consider this the most coherent definition.

> *"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". [4]*

16% of the respondents identified themselves with this approach.

> *"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" [8]*

4% of the respondents selected this option.

> *"Software whose purpose is to support sustainability goals, that is, to improve the sustainability of humanity on our planet." [4]*

4% of the respondents identified with this definition.

> *"Environment friendly software that helps improve the environment". [7]*

4% of the respondents selected this option.

Next, we asked the respondents whether sustainability should or should not be considered as a Nonfunctional Requirement (NFR). 58% of the respondents considered that sustainability should be considered an NFR. However, only 12% of the respondents were capable to provides reasonable statements supporting their opinion. Next, we cite the justifications of each of the respondents.

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 this as being important to *"Gain in growth / evolution of the system"* [#3]; and the last stated that *"the use would make the software more quality for the user"* [#7].

5.3.2 *RQ2: Sustainability importance level.*

In analyzing the degree to which respondents consider that companies should give importance to the sustainability issue in the software development process, we discovered that 52% treat the issue as "important" and 32% as "very important". For another 12%, it is "neutral" and 4% see "no importance" in the subject. By crossing this data with the question "What respondents understand that sustainability represents for companies?", we could see from Figure 3 that most respondents – 52% – see sustainability as an opportunity to gain new business. Nevertheless, 28% of the respondents believe that the use of sustainability in the software development process represents costs and expenses for companies. It is worth to mention that the total amount could exceed 100% as it was a multiple choice question.

In order to understand if companies have an ecological bias, in a general scope, with a focus not only on software, we asked the respondents if the company they worked for adopted any sustainability practice such as: proper disposal and recycling of waste, batteries, compliance with environmental legislation, saving water, energy and paper or others. Respondents chose one of the following answers:

- *Expert*: Meets all legislations, performs and encourages various practices.
- *Intermediate*: Meets several legislations and performs various practices.
- *Beginner*: Meets few legislations and performs some practices.
- No knowledge

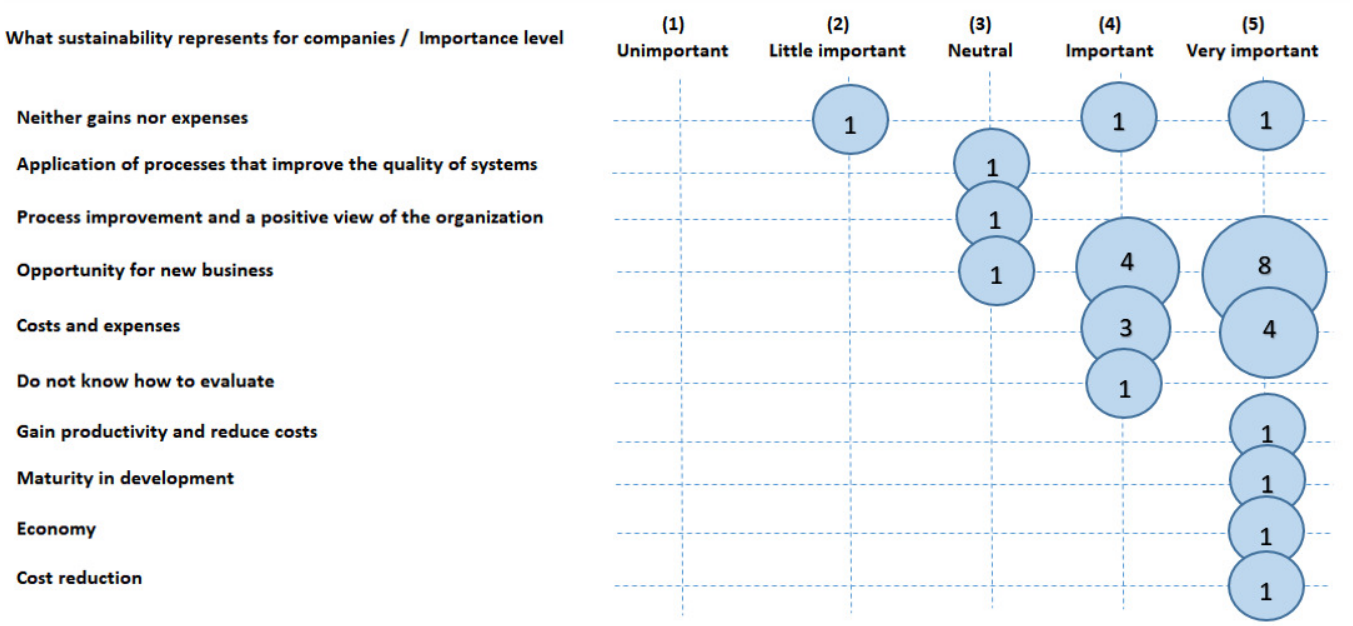

Figure 3: Company awareness level

- Does not comply with legislation

As we can see from Figure 4, 36% of the respondents could not answer if 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, 16% consider that the company is at the Intermediate level, taking into account several different laws and practices. Another 16% reported that their company did not comply with any legislation, and only 8% pointed out that the company complied with all laws and encourages the adoption of various practices. The following Brazilian laws were cited: Law No. 9,605[1], Sanitary Law No. 11,445/07[2], State Law No. 12,377[3], Decree No. 14,024[4] and Law No. 11,612[5].

Next, we asked whether their companies concerned with minimizing the negative impacts that traditional development process activities could have on the environment. 36% of the respondents reported that the company had a reasonable concern, neither so much nor not so much. For 32% of the respondents, the company did not care about such an issue. 16% reported that the company cared a bit. Another 16% are really concerned about the negative impacts. One of the justifications was *"More modern computer upgrades that consume less energy, electronic disposal policy, awareness to suspend and shut down the unused PC"* [#22].

When asked about the main barriers that hinder the adoption of sustainability actions and practices in the software development process of the corporate environment, 60% of the respondents stated that there is a lack of awareness of the companies. Another 57% understand that companies do not consider the subject relevant. 28% of the respondents could not evaluate, 24% responded that their companies do not have qualified staff and 20% reported difficulties to measuring likely earnings. In the view of 16% of the respondents, bureaucracy becomes a barrier. The remaining 16% consider it as a very expensive investment like Figure 5 shows. Because it is a multiple choice issue, the total ratio could exceed 100%.

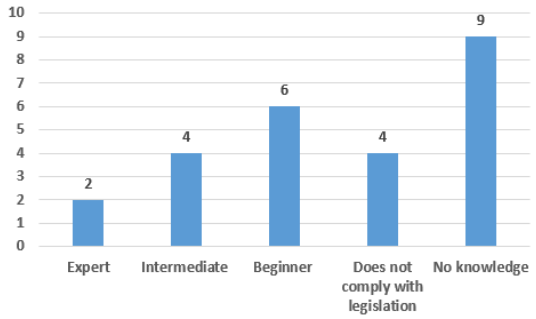

Figure 4: Company awareness level

5.3.3 *RQ3: Sustainable Software Development Process.*

We asked the respondents if they felt that companies should give importance to the sustainability issue in the software development process. 52% answered that it was "Very important". 32% considered this as being "Important", 12% reported as "Neutral" and only 4% considered as "Less important".

[1]Law 9,605 http://legislacao.planalto.gov.br/legisla/legislacao.nsf/Viw_Identificacao/lei%209.605-1998?OpenDocument
[2]Law 11,445 http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2007/lei/l11445.htm
[3]Law 12,377 http://www.seia.ba.gov.br/legislacao-ambiental/leis/lei-n-12377-de-28-de-dezembro-de-2011
[4]Decree 14,024 http://www.seia.ba.gov.br/sites/default/files/legislation/Decreto%2014024_2012.pdf
[5]Law 11,612 http://www.seia.ba.gov.br/legislacao-ambiental/leis/lei-n-11612

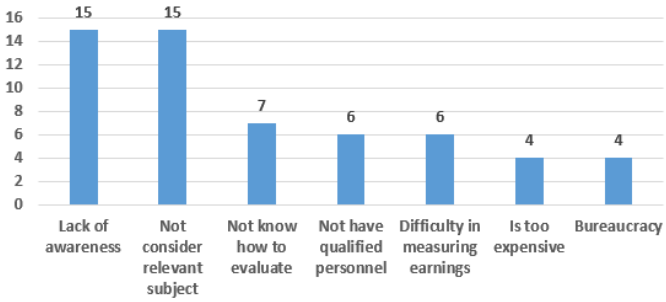

Figure 5: Main difficulties in adopting sustainable practices by companies.

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

When asked whether the companies they worked for used to encourage the adoption of sustainable practices in the software development process, 40% were unable to answer. 32% of them stated this was a rather common practice, while other 28% reported that their companies do not encourage. In addition, we also attempted to figure out, from the companies that encourage the use of sustainable practices, which are the covered SDLC phases. As Figure 6 shows, 38% of the companies use to adopt such practices in the development phase, 29% in the design phase, 24% in requirements and 10% in testing phase. The respondents were allowed to choose more than one SDLC phase.

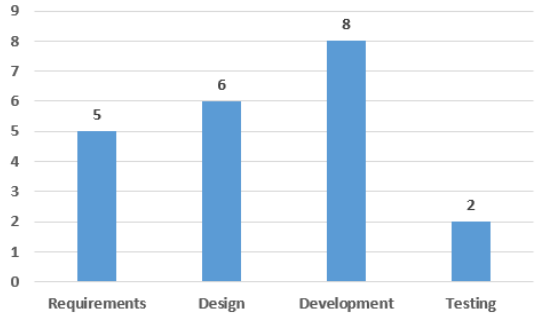

Figure 6: Phases of SDLC

We asked the respondents in which SDLC phases they could identify any deficiencies in terms of sustainability practices. 22% showed no deficiencies, 20% showed deficiencies in the development phase, 16% at the design stage, 14% in the requirements phase and 12% in the testing phase. The identified deficiencies were:

- **General**: Poorly defined processes [#1], Lack of initiatives [#15], All phases need practices aimed at sustainable software development because it is not a knowledge that the team has [#25].
- **Requirements**: Does not translate software needs by imagining future generations [#7], lack of professional qualification regarding the subject in the requirements phase [#11].
- **Design**: Lack of creation of a framework and availability of open architecture that allows the addition of new items [#9], There is a deficiency because no design pattern is applied to it [#23] Systems architecture is not thought of in order to minimize the use of energy of the software [#24].
- **Development**: Does not develop with reuse of item [#5], Lack of reuse of code [#8].

5.3.4 *RQ4: Sustainability dimensions.*

In this survey, we list the contributions proposed by Lago et al. [14], without to show 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 was observed. For each feature, respondents were presented a brief description and five unique response options.

Table 1 shows that, on average, 89% of the respondents considered all characteristics as either "Important" or "Very important". The "Very important" degree was attributed to the following characteristics: Adaptation to changes, Reusability and Quality of the system. The degree "Important" was attributed to the characteristics: Development oriented to features, Software evolution, Product roadmap, Awareness about the use of sustainable practices, Energy consumption, Environmental interest, Time to Market and Development effort. and Sustainable Ethics tied for grades 4 and 5.

This results show that professionals consider the technical dimension as the most important with a mean of 95%, followed by other dimensions: Social (88%), Economic (86%) and Environmental (72%).

5.3.5 *RQ5: Sustainability models.*

In this section it was investigated whether the professionals had an adequate knowledge about the green and sustainable software engineering field, and if in their companies they applied any process model to support sustainability in Software Engineering practices. As a result, 96% of the respondents said that they did not know about any applied models and 4% of them stated that the company uses the EPEAT tool[6] to compare and select technological peripherals based on their environmental attributes to make their purchases.

5.3.6 *RQ6: Sustainability tools.*

In this topic we seek to investigate if the company adopts some tool, technique or method to measure sustainability and also if there is the adoption of some sustainable design pattern in the software development process. 36% of the respondents stated they did not know about any, or did not know how to report on their use in their companies. Only one of the respondents stated that they consider energy efficiency when developing software. The respondent said: *"Efficient coding is in mind to minimize code and hardware resource lines (memory, disk and processing). The verification of good practices and adherence to items aimed at efficient coding, software and metric software (SONAR), CI / CD (JENKINS) and others that you would not know need due to the size of the company".*

[6]https://www.epeat.net/

Table 1: Sustainability Dimensions Analysis

Dimensions	Sustainability concern	Irrelevant (1)	Less important (2)	Neutral (3)	Important (4)	Very important (5)
Technical	Longevity		1		**12**	**12**
Technical	Resilience to uncertainty				11	**14**
Technical	Performance			1	9	**15**
Technical	Software Evolution		1	1	**13**	10
Technical	Reusability				7	**18**
Technical	System Quality				5	**20**
Social	Product Roadmap		1	3	**13**	8
Social	Awareness		1	1	**13**	10
Social	Ethics			3	**11**	**11**
Environmental	Energy consumption	1	1	5	**10**	8
Environmental	Environmental concern		1	6	**13**	5
Economic	Time to Market	1		3	**11**	10
Economic	Development effort			3	**12**	10

6 DISCUSSION

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

- ***Technical dimension***
  This survey revealed, according to the results of the analysis of RQ3, 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 reuse of source code. 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 the study [14].
  In terms of the software development process, we could see that companies could not yet be considered as green companies or aspiring to be sustainable companies because they do not use models, processes, methods and tools to support the development of their software. The professionals, although they do not have in-depth knowledge in the subject, visualize the advantages and importance of thinking sustainable in software development.
- ***Social dimension***
  We could observe, from the open questions, that the practitioners' awareness needs to happen 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 really happens 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 on how professionals perceive the impacts of software development and maintenance on the environment.
  With regard to 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 low knowledge of the practitioners on the subject, the participants attributed this as holding high importance. In the software bias, this dimension is directly related to energy consumption and environmental interests [15]. Most professionals reported that their companies do not have quality requirements related to sustainability. This insight reinforces the need for the academic community to increasingly join effort to make sustainability a software quality requirement.
  Through this study, it was possible to observe that the understanding about the homogenization of concepts used in this area is still uncertain. For software to be produced sustainable, 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***
  For professionals, the development of sustainable software 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. These could encompass hiring qualified people with a good understanding of the principles of software engineering. Thus, 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 [6]. In this sense, few participants mentioned this factor. Only 3 reported that sustainability is important, but it does not interfere with customer service functions. It must therefore be a product obligation, a requirement on the part of the customer.
In light of these discussions, we believe that companies must incorporate investments in business decisions to produce more sustainable software and implement sustainable Software Engineering practices. However, for industry, the expectation of return on these investments is still a gap.

In general, we see that practitioners' perception of all dimensions of sustainability is subtle. It is in the unconscious, but it could be better worked together, not just in the technical direction. To do so, these 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: knowing that software production has environmental impacts, accessing information, tools, methods, transferring knowledge into actions and raising awareness of these issues around them.

7 IMPLICATIONS FOR RESEARCH AND PRACTICE

In this section, we present the relevant implications that emerge from the analysis of this qualitative study:

- There is a clear indication that the Green Software Engineering field is still incipient and needs to gain more attention from the industry. There is a lack of knowledge about the topic, in particular regarding the concepts about sustainable software;
- There is a common understanding of the practitioners that sustainability should be treated as a quality attribute. Although they do not have in-depth knowledge in the subject, they visualize the advantages and importance of thinking sustainably in software development;
- The technical dimension is the most relevant and explored by professionals. Most practitioners have targeted their perceptions about sustainable software specifically in the quality attribute reuse of source code. The perception needs to be expanded in all dimensions concerns;
- Companies could not yet be considered as green companies or aspiring to sustainable companies because they do not use models, processes, methods and tools to support the green software development.

8 RELATED WORK

We have identified in recent literature few survey studies developed in the Green Software Engineering field.

A survey conducted with fifty-three software professionals in seven different companies was reported by Koçak et al. [13]. 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 analyse. 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. [16] 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 an appropriate support such as the creation of organizational policies and knowledge banks could help to create green software products.

Jagroep et al. [9] 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 questions 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 [6] 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.

In order to develop this work, we considered every mentioned study, since they bring relevant information. However, we observed that these studies were 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 a broader coverage.

9 THREATS TO VALIDITY

Construct Validity: During the pilot test, some respondents reported that the filling time of the instrument was extensive. As such,

respondents in our survey 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 in order to better target questions and answers.

Another threat was the respondents understanding about the questions. To help ensure the understandability of the survey, we asked professionals and researchers with experience in Software Engineering and experience in survey design to review the survey 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 which work in different domains (and which mostly have offices in several Brazilian cities) we believe this set might be representative.

External Validity: The respondents of our survey may not adequately represent all software practitioners. The response rate was 50%. Thus, our results could not be statistically relevant. Nevertheless, we believe that the 25 responses that we analyzed provide a rich source of qualitative data to reveal promising insights.

Reliability: It is a threat that the results of the research are influenced by interpretation. The coding process was performed by two authors working together. Disagreements in the assignment of codes were discussed until consensus was reached.

10 CONCLUDING REMARKS

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

This study was designed to gather data from software practitioners on this respect, and unveil details about the software industry's perception of sustainability in the software development process. The yielded results 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. Among the observed perspectives, the respondents indicate that the technical dimension is the most relevant and explored so far. This study contributes to the field with an initial set of evidence and we could deem it as a first step towards establishing a common understanding about how the software industry is receptive to the use of sustainability concepts in software development practices.

As future work, we aim to extend this research by applying with professionals from other geographic regions, in order to understand whether the achieved results still holds true when considering another set of scenarios. 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 in software development.

ACKNOWLEDGMENTS

This research was partially funded by INES 2.0; CNPq grants 465614/2014-0 and 408356/2018-9 and FAPESB grant JCB0060/2016.

REFERENCES

[1] 1987. Report of the World Commission on Environment and Development: Our Common Future. Available at http://www.un-documents.net/our-common-future.pdf. Last accessed in April 2019.
[2] Coral Calero and Mario Piattini. 2015. Introduction to green in software engineering. In *Green in Software Engineering*. Springer, 3–27.
[3] Coral Calero and Mario Piattini. 2017. Puzzling out software sustainability. *Sustainable Computing: Informatics and Systems* 16 (2017), 117–124.
[4] Markus Dick, Stefan Naumann, and Norbert Kuhn. 2010. A model and selected instances of green and sustainable software. In *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*. Springer, 248–259.
[5] Krisztina Erdelyi. 2013. Special factors of development of green software supporting eco sustainability. In *2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 337–340.
[6] Iris Groher and Rainer Weinreich. 2017. An interview study on sustainability concerns in software development projects. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 350–358.
[7] IT Harnessin Green. 2012. Principles and practices. *San Murugesan, GR* (2012).
[8] Lorenz M Hilty, Peter Arnfalk, Lorenz Erdmann, James Goodman, Martin Lehmann, and Patrick A Wäger. 2006. The relevance of information and communication technologies for environmental sustainability–a prospective simulation study. *Environmental Modelling & Software* 21, 11 (2006), 1618–1629.
[9] Erik Jagroep, Jordy Broekman, Jan Martijn EM Van Der Werf, Patricia Lago, Sjaak Brinkkemper, Leen Blom, and Rob Van Vliet. 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)*. IEEE, 76–85.
[10] Timo Johann, Markus Dick, Eva Kern, and Stefan Naumann. 2011. Sustainable development, sustainable software, and sustainable software engineering: an integrated approach. In *2011 International Symposium on Humanities, Science and Engineering Research*. IEEE, 34–39.
[11] Mark Kasunic. 2005. *Designing an effective survey*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
[12] Barbara A Kitchenham and Shari Lawrence Pfleeger. 2002. Principles of survey research part 2: designing a survey. *ACM SIGSOFT Software Engineering Notes* 27, 1 (2002), 18–20.
[13] Sedef Akinli Koçak, Gülfem Isiklar Alptekin, and Ayse Basar Bener. 2015. Integrating Environmental Sustainability in Software Product Quality.. In *RE4SuSy@ RE*. 17–24.
[14] Patricia Lago, S Aklini Kocak, Ivica Crnkovic, B Penzensradler, et al. 2015. Framing sustainability as a property of software quality. (2015).
[15] Patricia Lago and Birgit Penzenstadler. 2017. Reality check for software engineering for sustainability—pragmatism required. *Journal of Software: Evolution and process* 29, 2 (2017), e1856.
[16] Irene Manotas, Christian Bird, Rui Zhang, David Shepherd, Ciera Jaspan, Caitlin Sadowski, Lori Pollock, and James Clause. 2016. An empirical study of practitioners' perspectives on green software engineering. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 237–248.
[17] Brunna C Mourão, Leila Karita, and Ivan do Carmo Machado. 2018. Green and Sustainable Software Engineering-a Systematic Mapping Study. In *Proceedings of the 17th Brazilian Symposium on Software Quality*. ACM, 121–130.
[18] Stefan Naumann, Markus Dick, Eva Kern, and Timo Johann. 2011. The greensoft model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems* 1, 4 (2011), 294–304.
[19] Birgit Penzenstadler and Henning Femmer. 2013. A generic model for sustainability with process-and product-specific instances. In *Proceedings of the 2013 workshop on Green in/by software engineering*. ACM, 3–8.
[20] Birgit Penzenstadler, Ankita Raturi, Debra Richardson, Coral Calero, Henning Femmer, and Xavier Franch. 2014. Systematic mapping study on software engineering for sustainability (SE4S). In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 14.
[21] Gustavo Pinto and Fernando Castor. 2017. Energy efficiency: a new concern for application software developers. *Commun. ACM* 60, 12 (2017), 68–75.
[22] Theresia Ratih Dewi Saputri and Seok-Won Lee. 2016. Incorporating sustainability design in requirements engineering process: A preliminary study. In *Asia Pacific Requirements Engineering Conference*. Springer, 53–67.
[23] Kevin Tate. 2005. *Sustainable software development: an agile perspective*. Addison-Wesley Professional.