UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CHARACTERIZING SUSTAINABLE SOFTWARE ENGINEERING THROUGH A MULTI-METHOD APPROACH

Leila Karita dos Anjos do Espírito Santo

DISSERTAÇÃO DE MESTRADO

Salvador/BA
MARÇO/2020

LEILA KARITA DOS ANJOS DO ESPÍRITO SANTO

CHARACTERIZING SUSTAINABLE SOFTWARE ENGINEERING THROUGH A MULTI-METHOD APPROACH

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Ivan do Carmo Machado

Salvador/BA
MARÇO/2020

Ficha catalográfica elaborada pela Biblioteca Universitária de
Ciências e Tecnologias Prof. Omar Catunda, SIBI - UFBA.

E77 Espírito Santo, Leila Karita dos Anjos do

Characterizing Sustainable Software Engineering through a Multi-Method Approach/ Leila Karita dos Anjos do Espírito Santo. – Salvador, 2020.

114 f.

Orientador: Prof. Dr. Ivan do Carmo Machado

Dissertação (Mestrado) – Universidade Federal da Bahia. Instituto de Matemática e Estatística, 2020.

1. Engenharia de Software. 2. Software. 3. Engenharia. I. Machado, Ivan do Carmo. II. Universidade Federal da Bahia. III. Título.

CDU 004

"*Characterizing Sustainability in Software Engineering through a Multi-Method Approach*"

Leila Karita dos Anjos do Espírito Santo

Dissertação apresentada ao Colegiado do Programa de Pós-Graduação em Ciência da Computação na Universidade Federal da Bahia, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Banca Examinadora

Prof. Dr. Ivan do Carmo Machado (Orientador-UFBA)

Prof.ª Dr.ª Rita Suzana Pitangueira Maciel (UFBA)

Profª. Drª. Monalessa Perini Barcellos(UFES)

RESUMO

O interesse da comunidade de Tecnologia da Informação e Comunicação (TIC) sobre sustentabilidade tem crescido consideravelmente nos últimos anos. Embora ainda em estágio inicial, a temática tornou-se de grande relevância uma vez que nos força a pensar sobre o que temos feito para garantir a continuidade das gerações futuras. No contexto da Engenharia de Software (ES), quando pensamos no desenvolvimento de software sustentável nos deparamos com lacunas, uma vez que esse assunto ainda é nebuloso para os engenheiros de software e comunidade de pesquisa em ES. Para que o software seja produzido de forma sustentável, os engenheiros de software precisam entender como os conceitos de sustentabilidade estão incorporados ao desenvolvimento de software, de forma que possam ter uma compreensão clara, comum e compartilhada desse conhecimento. No entanto, o desenvolvimento de um estudo recente sobre o estado-da-arte das abordagens de software que apoiam a ES sustentável mostrou que ainda há uma lacuna sobre o que, de fato, vem a ser o desenvolvimento de software sustentável. A falta de tal entendimento pode impedir a indústria de construir software com consciência sustentável. Esta pesquisa tem como objetivo caracterizar a Engenharia de Software Sustentável destacando as preocupações sustentáveis presentes no ciclo de vida de desenvolvimento do software. Para alcançar este objetivo, esta dissertação adotou uma abordagem multi-método e produziu uma série de estudos qualitativos. A abordagem multi-método é uma estratégia de pesquisa metodológica que combina dois ou mais métodos de pesquisa qualitativos. Foram elaborados: um estudo de mapeamento sistemático na intenção de conhecer o domínio de ES sustentável; um survey para obter uma percepção da indústria de software sobre a adoção de práticas sustentáveis; e uma grounded theory, com o objetivo de prover aos leitores um entendimento comum sobre o desenvolvimento de software sustentável. Esta dissertação concluiu que o desenvolvimento de software sustentável pode ser caracterizado através das seguintes sentenças: preocupações técnicas, ambientais e sociais estão presentes em todas as fases do desenvolvimento de software sustentável. Isso significa que os pesquisadores e os engenheiros de software estão preocupados em considerar a longevidade do software produzido, os recursos ambientais e o bem-estar social; a identificação de requisitos sustentáveis deve ocorrer na fase inicial do projeto com o apoio de especialistas engajados com a sustentabilidade, os quais devem fazer parte do grupo de stakeholders; e o uso de preocupações sustentáveis pode gerar trade-offs no projeto. Os resultados contribuem com uma maior compreensão à respeito do desenvolvimento de software sustentável, a partir das perspectivas da literatura e praticantes de software; e, consequentemente, com a evolução do estado da arte em ES Sustentável.

Palavras-chave: Engenharia de Software Sustentável, Engenharia de Software Verde, Engenharia de Software Empírica, Abordagem Multi-Método

ABSTRACT

The interest of the Information and Communication Technology community on sustainability has grown considerably in recent years. Although still at an early stage, the theme has become of great relevance since it forces us to think about what we have done to ensure the future generations continuity. In the Software Engineering context, when we think about sustainable software development, we face gaps, since this subject is still nebulous for software engineers and research community in Software Engineering. For software to be produced in a sustainable way, software engineers need to understand how sustainability concepts are incorporated into software development, so that they can have a clear, common and shared understanding of that knowledge. However, the development of a recent study on the state-of-the-art about software approaches that support sustainable Software Engineering showed that there is still a gap about what is sustainable software development, in fact. The lack of such an understanding can prevent the industry from building software with sustainable awareness. This research aims to characterize sustainable software engineering by highlighting the sustainable concerns present in the sustainable software development. To achieve this goal, this dissertation adopted a multi-method approach and produced a series of qualitative studies. The multi-method approach is a methodological research strategy that combines two or more qualitative research methods. We elaborated: a systematic mapping study with the intention of knowing the Sustainable Software Engineering domain; a survey to obtain the software industry perception on the adoption of sustainable practices; and a grounded theory, with the aim of to provide a common understanding of sustainable software development to readers. This dissertation concluded that the sustainable software development is characterized through the following sentences: technical, environmental and social concerns are present in all phases of sustainable software development. This means that researchers and software engineers are concerned with considering the longevity of the software produced, the environmental resources and the social welfare; the sustainable requirements identification must occur in the project initial phase with the support of experts engaged in sustainability, who must be part of the group of stakeholders; and the use of sustainable concerns can generate trade-offs in the project. Therefore, the results contribute to a greater understanding of sustainable software development, from the literature and software practitioners perspectives; and, consequently, with the evolution of the state-of-the-art in Sustainable Software Engineering.

Keywords: Sustainable Software Engineering, Green Software Engineering, Empirical Software Engineering, Multi-Method Approach.

AGRADECIMENTOS

Inicialmente, eu gostaria de agradecer à minha família e amigos. Especialmente aos meus pais, Miriam e José Alexandre e irmãs Lilane e Vanessa, pelo apoio incondicional em todos os aspectos da minha vida. Estejam cientes do meu grande amor por vocês! Agradeço por todos os esforços para tornar esta jornada possível.

Gostaria de expressar minha mais sincera gratidão ao meu marido Rogério por sua compreensão, paciência e incentivo para enfrentar mais um desafio na minha vida. E ao meu amado filho Pedro, que foi o maior incentivador para eu concluir essa etapa na minha vida.

Gostaria de agradecer ao professor Ivan Machado por me aceitar como aluna de mestrado, pela sua orientação e pelas muitas sugestões e incentivos durante o curso deste trabalho, o qual se mostrou bastante desafiador.

Meus sinceros agradecimentos aos amigos e colegas que conheci durante essa jornada que foram capazes de proporcionar um ambiente divertido e motivador.

Gostaria de expressar meus sinceros agradecimentos à Neyla Fontan, minha mentora de vida, por mais uma vez abrir portas para novas oportunidades; e à Brunna Caroline, por participar diretamente do nascimento desse projeto.

Finalmente, eu gostaria de agradecer a Deus que me abençoou com essa oportunidade, me protegeu e me guiou nessa jornada.

ACKNOWLEDGMENTS

Initially, I would like to thank my family and friends. Especially to my parents, Miriam and José Alexandre and my sisters Lilane and Vanessa for their unconditional support in all aspects of my life. Please be aware of my huge love for you! I warmly appreciate and thank you for all the efforts you have made to make my journey possible.

I would like to express my most sincere gratitude to my husband Rogério for his understanding, patience and encouragement to face yet another challenge in my life. And to my beloved son Pedro, who was the greatest encourager for me to accomplish this stage of my life.

I would like to thank Professor Ivan Machado for his guidance in accepting me as a Master Student and for the many helpful suggestions and encouragement during the course of this work, which has proved quite challenging.

My heartfelt thanks to my friends and colleagues who I have met during my Master's degree journey who were able to provide a fun and motivating environment.

I would like to express my sincere thanks to Neyla Fontan, my life mentor, for once again opening doors to new opportunities; and Brunna Caroline, for participating directly in the birth of this project.

In the end, I would like to thank God who blessed me with this opportunity, protected me, and guided me through this journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CS	Compute Science
CSR	Corporate Social Responsibility
ICT	Information and Communication Technology
GT	Grounded Theory
MDG2	Millennium Development Goals
NFR	Non-functional Requirement
RE	Requirements Engineering
RE4S	RE for Sustainability
SDLC	Software Development Life-Cycle
SE	Software Engineering
SE4S	SE for Sustainability
SMS	Systematic Mapping Study
SWEBOK	Software Engineering Body of Knowledge
UN	United Nations

Chapter 1

INTRODUCTION

This Chapter presents the context of the work, which motivated this research and its objectives. It also presents the employed research methodology and the dissertation structure.

This Chapter is structured as follows: Section 1.1 presents the context; Section 1.2 presents the research problem; Section 1.3 presents the main goal of this investigation; Section 1.4 presents the specific goals; section 1.5 presents the research method; and finally, Section 1.6 presents the structure of the dissertation.

1.1 CONTEXT

"To date, hardware improvements have yielded interesting results on the energy efficiency of components, thanks to the industry's keen interest in this regard"(Acar, 2017). However, while hardware is physically responsible for power consumption, its operations are influenced by the use of some software. A lot of research has been done around reducing the power consumption of the hardware. However, it is known that the software has an indirect interference on power consumption. Calero et al. (2013) claim that improving energy consumption allows for a sustainable software product.

Although still in its early stages, software sustainability is a very important research topic that will be of great relevance in the coming years (CALERO; PIATTINI, 2017). Oyedeji et al. (2016) cite some factors considered relevant that support this observation: the understanding, motivation and commitment of management and engaged personnel in relation to the economy, society and environment.

When it comes to the software development process, it is still a new practice for software engineers and developers, as well as researchers. The research community in SE has already applied its efforts to adopt sustainability as a quality attribute.

Developing software with quality, i.e., that meets both specifications and the expected quality attributes requires the use of well-defined software engineering processes. Therefore, it is necessary to define sustainability goals that will be followed in each phase of the software development process.

However, while sustainability is an emerging concept in SE, this view is still omitted during the software development process (PENZENSTADLER et al., 2014a). To Oyedeji et al. (2016), "Contrary to the notion that software is environmentally friendly from its virtual form, the processes and methods used to develop, maintain and deploy software have an environmental and social impact that is not normally considered by software development practices."

Dick et al. (2010), in turn, state that "the business sense SDLC model is not appropriate for identifying the effects of software on sustainability". The main focus of the current model is on the business, development and maintenance phases, without considering sustainability (OYEDEJI et al., 2016).

Supported by a general understanding that sustainable software development is an important issue in the SE field, there is a clear gap that the meaning of sustainability in software is not yet clear. According to domain researchers, a homogeneous and consistent definition is needed to help software practitioners to understand clearly and unambiguously the adoption of sustainability in the software development process (OYEDEJI et al., 2016; PENZENSTADLER et al., 2014a).

A number of definitions have emerged in the literature. Venters et al. (2014a) presented different viewpoints in the field of computer science and engineering to contribute to the question: what is software sustainability? In their work, the definition of sustainability is taken from the English Oxford dictionary as "the quality of being sustained" where sustained is defined as "capable of being supported and maintained". As a result, they suggested a distinction between "software for sustainability" which is related to the absolute definition and "sustainable software" which is related to the relative definition. The difference between them is that the first has fixed variables while the later requires that the variables are chosen based on the context.

Calero et al. (2013) has brought to light the definition of sustainability of a software product as "the ability to develop a software product in a sustainable manner and claim that this is totally uncertain and difficult to quantify". Other definitions can be found in (TAINTER, 2006; DICK; NAUMANN, 2010; JOHANN et al., 2011).

Given this context, it is observed that sustainability in software development is a theme that has several viewpoints with different researchers describing it from their own perspective in their area of expertise. This has hindered consensus on what sustainability really means and how it relates to software development (OYEDEJI et al., 2016).

1.2 RESEARCH PROBLEM

The preliminary study of this dissertation consisted of investigating the state-of-the-art in Sustainable SE domain, which is presented in details in Chapter 4. The objective was to synthesize the evidence available in the literature and to identify gaps and research opportunities. One of the results of this study led us to realize that the scope and context of this incipient area are not clearly defined yet. Consequently, the sustainability in software development is a theme that has several viewpoints with different researchers describing it from their own base perspective in their area of expertise and interest. This has hindered consensus about the sustainability concepts and how it relates to software development. In line with what we discussed in the previous section, this gap about *"what is the sustainable software development?"* is a problem that requires gather investigation.

This dissertation aimed to characterize the sustainable software development through a multi-method approach. We produced a set of qualitative studies: a Systematic Mapping Study (SMS) about the sustainable SE field, a survey about the software industry's perception and an initial theory about sustainable software development. The outcomes of this dissertation can contribute to a convergence in the understanding of the SE domain and to support the software industry and research community to understand the existence of sustainable concerns in the sustainable software development context.

1.2.1 Research Question

The overall research question that guides this work can be stated as follows:
How is it characterized the sustainable software engineering?

1.3 GENERAL OBJECTIVE

This research aims to investigate how the sustainable software engineering is characterized. We used the results and experiences obtained during the investigation to propose a set of fundamentals principles about understanding sustainable software development.

1.4 SPECIFIC OBJECTIVES

In order to achieve the general goal of this research, the following specific objectives were defined:

O1 To investigate the state-of-the-art in sustainable SE in order to synthesize the available evidence from the literature and to identify gaps and research opportunities in sustainable SE field;

O2 To investigate the industry's perception of adopting sustainable practices in the sustainable software development;

O3 To identify the sustainable software development inherent concerns from the perspective of researchers investigating the Sustainability in the SE field;

O4 To propose an initial theory to define a set of fundamentals principles about understanding sustainable software development based on the identified concerns.

1.5 RESEARCH METHODOLOGY

To achieve the defined objectives, the research methodology employed in this dissertation comprised a set of empirical studies: systematic mapping study, survey and grounded theory. Figure 1.1 shows this set of studies whose details are presented below:

Phase 1 First, we developed a systematic mapping study (MOURÃO et al., 2018) to investigate the state-of-the-art on Sustainability in SE in order to synthesize available evidence in the literature and to identify gaps and research opportunities. This

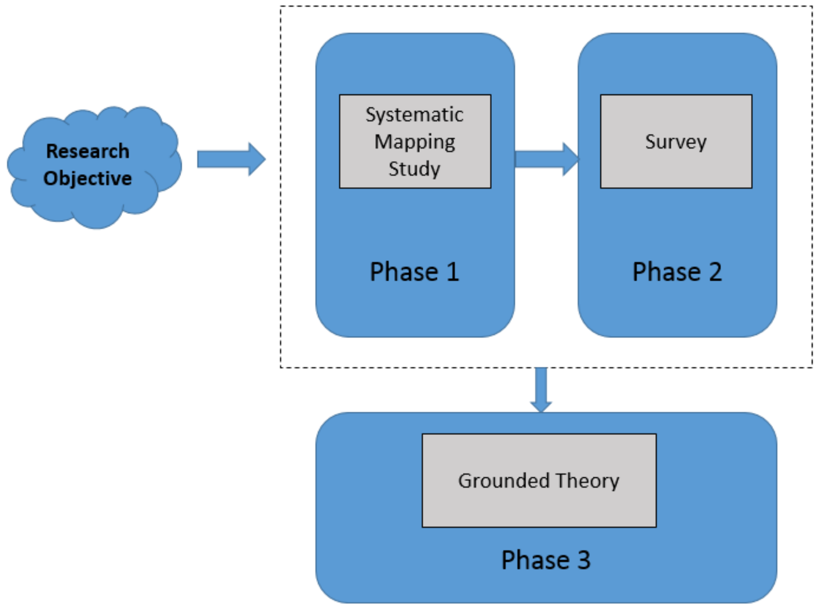

Figure 1.1 Research Methodology

study was carried out together with other researchers in our research group. The mapping study is a method defined to build a classification scheme and structure the research (PETERSEN et al., 2008);

Phase 2: We next performed an industry survey (KARITA et al., 2019) with twenty-five software engineers involved in projects of different domains. It aimed to provide evidence about the practitioners's perception about the adoption of sustainability in software development, under four main perspectives: economic, social, environmental and technical;

Phase 3: In this phase, we aimed to obtain a common understanding on sustainable software development on the perception of researchers in sustainable SE and practitioners of the software industry. We adopted the Grounded Theory (GT) method (GLASER et al., 1968) to identify the main sustainability concerns obtained from data retrieved in the phase 1 and phase 2, as aforementioned.

1.6 DISSERTATION STRUCTURE

This dissertation is structured in seven chapters. In this introductory chapter the motivations, the justification for this work and the objectives were presented. The rest of the dissertation is structured as follows:

Chapter 2 Presents the theoretical foundation necessary for understanding the context in which this work is inserted, the Sustainable SE;

Chapter 3 Presents the research design, research philosophy in relation to other philosophies and expose our research strategy, including the research methodologies adopted;

Chapter 4 Presents a systematic mapping study on the state-of-the-art in Sustainable SE;

Chapter 5 Presents a survey with software's practitioners to investigate the adoption of sustainable practices in software development process;

Chapter 6 Presents a qualitative analysis study to understand the sustainable software development;

Chapter 7 Presents the final considerations of the work by highlighting the contributions of the research and discussing opportunities for future work.

Chapter 2

BACKGROUND

In recent decades, research on sustainability has gained popularity through a number of world-leading initiatives in the key purpose of bringing people together for the protection of our planet. An initiative comes from the United Nations (UN) which ratifies the need to reduce energy consumption and carbon footprint, which is one of the issues discussed in the Millennium Development Goals (MDG2) (CALERO; PIATTINI, 2015).

Despite these efforts, although knowledge about how to achieve sustainable development has grown, political actions towards achieving the goal is still in its infancy (PENZENSTADLER, 2014).

Many investigations have been carried out to understand how the concepts of sustainability could be incorporated to the traditional SE concepts, in particular in terms of how sustainable software can be produced by a sustainability-aware software development process.

The term sustainable software has been interpreted in two ways in the literature: (1) green in software: the software code being sustainable, agnostic of purpose, or (2) green by software: the software purpose being to support sustainability goals, i.e., improving the sustainability of humanity in our planet (HILTY et al., 2006; PENZENSTADLER et al., 2014a). However, a sustainable software product can only be achieved if a developing organization is aware of both negative and positive impacts on sustainable development that will likely be caused when using it. Additionally, it is necessary that the development process itself is environment-friendly (NAUMANN et al., 2011).

This Chapter presents the theoretical foundation necessary for understanding the context in which this work is inserted. We present the sustainable SE context and describe the dimensions and requirements of sustainability.

Section 2.1 introduces the definition of sustainable development; Section 2.2 presents the relationship between SE and sustainability; Section 2.3 introduces the sustainable dimensions and finally, Section 2.4 summarizes the Chapter.

2.1 SUSTAINABLE DEVELOPMENT

Sustainability or sustainable development 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". In general, sustainability is the "capacity to endure", but interpreting this concept requires context. Murugesan (2008) states that it is all about meeting needs and seeking a balance between people, environment and economy. Calero and Piattini (2015) define sustainability as a widely used term and refers to the capacity of something to last for a long time.

The term "sustainable development" was coined in 1987 by Gro Harlem Brundtland, former Prime Minister of Norway, who served as chair of an UN commission. She 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" (CALERO; BERTOA, 2013). This is the most popular definition of sustainable development used by IT researchers.

According to Calero and Piattini (2015), 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".

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 was correctly discussed by Calero and Piattini (2017). They stated that *"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 instance"*.

2.2 SUSTAINABLE SOFTWARE ENGINEERING

Aligned with the aforementioned section, the SE community has increased considerably its interest in unveiling the impacts of software on natural resources consumption. Several definitions of sustainable SE have been founded in the literature (CALERO; PIATTINI, 2015).

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

In this context, the sustainable SE process might be documented, continuously monitored, measured and evaluated for optimization of the software product (NAUMANN et al., 2011; KHANDELWAL et al., 2017).

In the literature, authors have used the terms "sustainability" and "green" in interchange way. Green (2012) cites a subtle difference between them: "sustainability" encompasses environmental, social and economic dimensions; and "green" refers to environmental aspects. For all intents and purposes, in this dissertation, we adopt such terms as synonyms.

In the following subsections, we provide a snapshot of how terms "Sustainability *in* Software Engineering", "SE *for* Sustainability", "Green *by* software" and "Green *in* software" are interpreted within the sustainable SE context.

2.2.1 Sustainability in SE X SE for Sustainability

Green or Sustainability in Software Engineering consists of practices that use engineering principles in software development with a focus on environmental aspects. Thus, the software is developed, operated and maintained in a "green" manner to be a "green" software product (KHANDELWAL et al., 2017; CALERO; PIATTINI, 2017).

On the other hand, SE for Sustainability (SE4S) has been developed and become a current focus of research due to the result of the involvement of software engineers in issues related to the impact of software systems on global sustainability (PENZENSTADLER et al., 2014b). According to these authors, the aim of SE4S is to use adequate methods and tools to minimize the environmental impact of the software development process and to generate a positive impact on social and economic sustainability.

Aligned with the aforementioned, the first statement has a technical focus. The latter analyzes the software objective and context more broadly, looking at how it can support social and environmental concerns such as the adoption of a sustainable lifestyle that sustains the planet and society (LAGO; PENZENSTADLER, 2017).

2.2.2 Green by Software X Green in Software

According to Calero and Piattini (2017), "the definitions of sustainable software found in literature are rather chaotic as regards concepts, and it is possible to find terms such as green software, green through software, green in software, etc".

Conceptually, green software can be divided into "green by software" and "green in software". The main difference is whether the goal pursued is to have more environment friendly software or if it is rather to produce software that helps the environment (KHANDELWAL et al., 2017).

Figure 2.1 shows this relation. Next, we explain each one.

Green by software covers software developed for domains that work in the preservation of the environment, as well as software that helps to manage energy-intensive

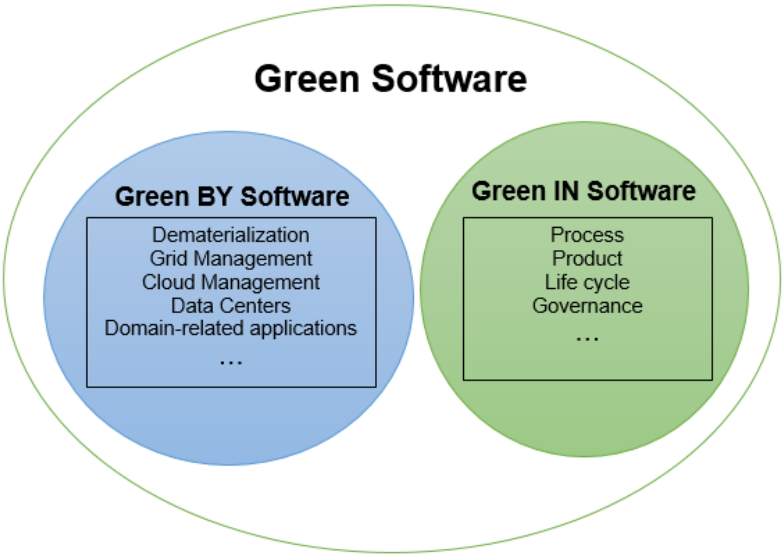

Figure 2.1 'Green BY' and 'Green IN' software (CALERO; PIATTINI, 2015)

applications.

Green in software is related to how to make software in a more sustainable way resulting in a more sustainable product. The practices which apply engineering principles to software by taking into consideration environmental aspects is referred to as green in software engineering.

2.3 SUSTAINABILITY DIMENSIONS

The concept of sustainability provided by the UN is supported by three main pillars or dimensions, which are divided into environmental, social and economic, as Figure 2.2 (CALERO; PIATTINI, 2015) shows.

The overall sustainability of our daily lives can only occur when the environmental, social, and economic aspects are in balance. These dimensions specify different focus points and are connected with different roles. Saputri and Lee (2016) state that "it is important to take different sustainability dimensions into account. Sustainability should be considered as an integrated concept". According to Calero and Piattini (2015), this has to be reflected in the software systems we create.

Although the focus of sustainable development is on balancing the environmental, social and economic dimensions, researches in Sustainable IT define five dimensions, with two additional: individual and technical. We next describe how these dimensions are represented in sustainable so far in the view of Penzenstadler and Femmer (2013). Each dimension encompasses a set of requirements:

- **Individual sustainability** refers to maintaining human capital (e.g., health, education, skills, knowledge, leadership, and access to services). Individual sustain-

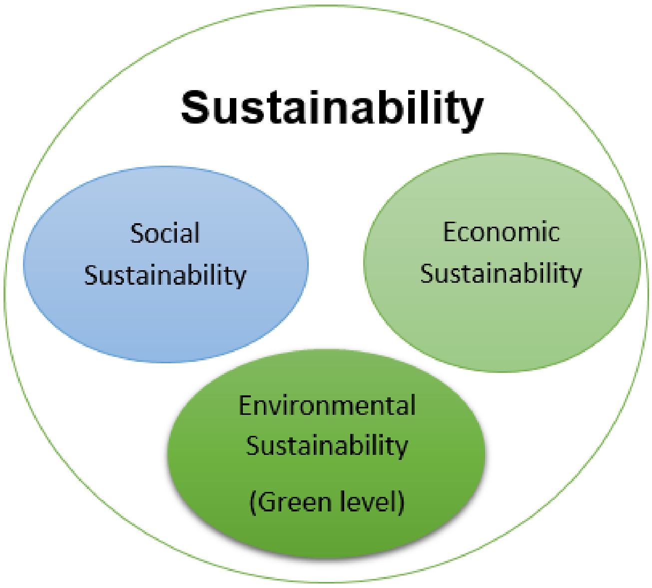

Figure 2.2 Sustainability dimensions (CALERO; PIATTINI, 2015)

ability can be covered by privacy, safety, security, human-computer interaction, usability, personal health, and well-being;

- **Environmental sustainability** seeks to improve human welfare by protecting natural resources, such as water, land, air, minerals, and the whole ecosystem. Any system applied in a real-world context is situated within a natural environment which means that it has an impact on the environment. Environmental sustainability can be managed by controlling the resources flow: waste management, life-cycle analysis, and environment impact assessment;

- **Social sustainability** aims at preserving the societal communities in their solidarity, services and solidarity of social communities. It could be handled via computer-supported collaborative work that aims to strengthening community building and improve community interaction;

- **Economic sustainability** aims to maintain capital assets and added-value (interest) assets. Economic sustainability can be interpreted in terms of costs, budget constraints, long-term business objectives, and market requirements among other economic requirements;

- **Technical sustainability** refers to software systems longevity and their adequate evolution with changing surrounding conditions and respective requirements. Technical sustainability requirements include all requirements which lead to the longevity of a system such as non obsolescence requirements and the ISO/IEC 9126 (International Standards Organisation (ISO), 1991) quality characteristics (eg, maintainability, reliability, and transferability). Moreover, energy efficiency is also part of technical sustainability requirements.

2.4 CHAPTER SUMMARY

In this Chapter, an overview of the main sustainability concepts has been provided. We explored the theoretical foundation necessary for understanding the context in which this study is inserted. We presented the context of sustainability and SE and described the dimensions and requirements of sustainability.

Finally, we approached the relationship between Sustainability in SE X Sustainability for SE and Green by Software X Green in Software. This dissertation adopted the terms "Sustainability in SE" and "Green in Software", since the objective is the software itself and the practices which apply engineering principles to software.

Next Chapter, we presented the research methodology applied in this dissertation.

Chapter

3

RESEARCH METHODOLOGY

To develop science, scientists use a set of basic rules also known as scientific methods. Like other science areas, the field of Compute Science (CS) has also adopted scientific methods. In the recent years, the SE research community has shown a growing interest in empirical research and constantly has been seeking to improve both the quantity and quality of research evidence by using an appropriate research method.

In this chapter, we present a brief description of the research philosophies and approaches adopted by the SE community. We also present the multi-method research methodology selected in this dissertation. Multi-method is the methodological choice that combine different data sources within the same paradigm, which may be just two or more qualitative methods or two or more quantitative methods.

The remainder of this Chapter is organized as follows: Section 3.1 presents the research philosophy; Section 3.3 presents the research methods; Section 3.2 presents the research approach; Section 3.4 describes the design research; and finally, Section 3.5 presents the Chapter Summary.

3.1 RESEARCH PHILOSOPHY

Philosophical assumptions underpin the research process which disposes researchers towards divergent paradigms and methodologies (COLEMAN; O'CONNOR, 2007). The research conduction is conceived according to the research strategy adopted through the chosen research philosophy and, consequently, of the research instruments used to answer the research objective. Our research question and research objectives have been earlier outlined in Chapter 1.

A research philosophy is a belief about the way in which data about a phenomenon should be gathered, analysed and used. It deals with the source, nature and knowledge development (BAJPAI, 2011). Figure 3.1 shows the vast research philosophy subject. Identifying it is the first step in the research methodology. Two major research philosophies have been identified in the Western tradition of science: (i) positivism or scientific and (ii) interpretivism, also known as antipositivism. (GALLIERS; LAND, 1987; COLEMAN; O'CONNOR, 2007; COLLINS, 2018). Next, we describe both research philosophy kinds.

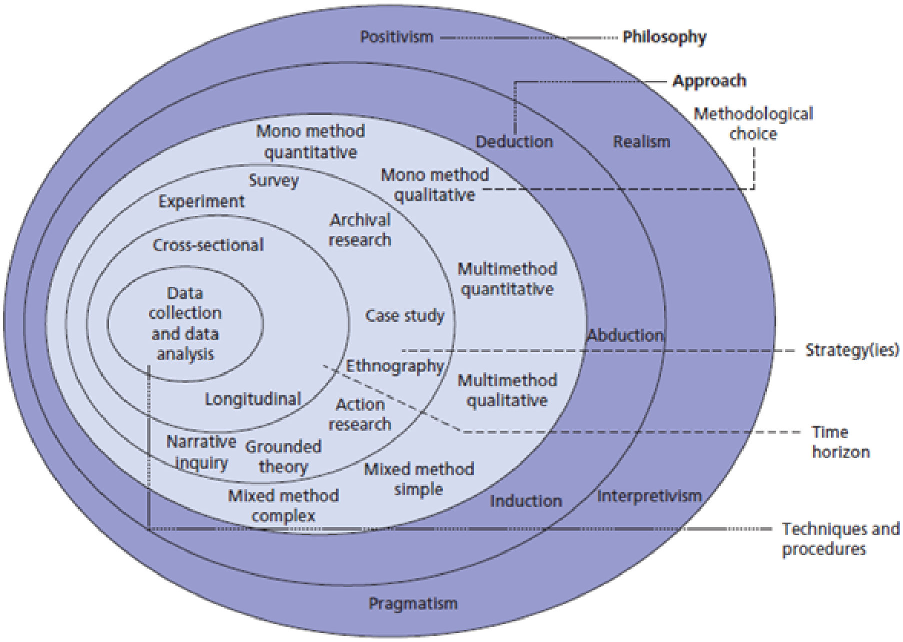

Figure 3.1 Research Philosophy (SAUNDERS et al., 2007)

3.1.1 Positivism

The positivist philosophy claims that phenomena must be isolated and that observations must be repeatable. Reality is stable and can be observed and described from an objective point of view and should not interfere with the phenomena under study. According to Collins (2018), *"as a philosophy, positivism is in accordance with the empiricist view that knowledge stems from human experience. It has an atomistic, ontological view of the world as comprising discrete, observable elements and events that interact in an observable, determined and regular manner"*.

The literature states that, in its essence, positivism is based on the idea that science is the only way to learn about truth. In positivist studies, the research results are observable and quantifiable. In such studies, the researcher role is limited to the collection and objective data interpretation.

Positivist studies generally adopt the deductive approach, which is based purely on facts (CROWTHER; LANCASTER, 2012). Wilson (2014) argues that if the researcher takes a positivist choice is because he believes his research is purely objective. Also, that there is a minimal interaction with research participants. Alavi and Carlson (1992), in a review of 902 articles on Information Systems, found that all empirical studies adopted positivism.

3.1.2 Interpretivism

Interpretivism philosophy integrates human interest in the study and involves researchers to interpret the elements of the study. Thus, *"interpretive researchers assume that access to reality is only through social constructs such as language, consciousness, shared meanings and instruments"* (MYERS, 2013).

Interpretivists claim that reality can only be fully understood through subjective interpretation. For interpretative philosophy, the study of phenomena in their natural environment is fundamental. This development is based on the critique of positivism in the social sciences. Thus, this philosophy emphasizes qualitative over quantitative analysis. Additionally, it generally focuses on meaning and may employ multiple methods to reflect different aspects of the issue (SAUNDERS et al., 2007).

Interpretivism is *"associated with the philosophical position of idealism, and is used to group together various approaches, including social constructivism, phenomenology, and hermeneutics; approaches that reject the objectivist view that meaning resides within the world regardless of consciousness"* (COLLINS, 2018).

Table 3.3 presents a brief comparison between the two major research philosophy paradigms.

Table 3.1 Differences between positivism and interpretivism, adopted from Pizam and Mansfeld (2009, p.1).

Assumptions	Positivism	Interpretivism
Nature of reality	Objective, tangible, single	Socially constructed, multiple
Goal of research	Explanation, strong prediction	Understanding, weak prediction
Focus of interest	What is general, average and representative	What is specific, unique, and deviant
Knowledge generated	Laws Absolute (time, context, and value free)	Meanings Relative (time, context, culture, value bound)
Subject/Researcher relationship	Rigid separation	Interactive, cooperative, participative
Desired information	How many people think and do a specific thing, or have a specific problem	What some people think and do, what kind of problems they are confronted with, and how they deal with them

3.2 RESEARCH APPROACH

There are three main types of research approaches: (i) Deductive; (ii) Inductive; and (iii) Abductive.

The relevance of hypotheses to the study is the main distinctive point between deductive and inductive approaches. Deductive approach tests the validity of assumptions (or theories/hypotheses) in hand, whereas inductive approach contributes to the emergence of new theories and generalizations. On the other hand, the abductive research, starts with "surprising facts" or "puzzles" and the research process is dedicated to explaining (BELL et al., 2018).

While quantitative research operates in a deductive way, qualitative research operates in an inductive way. A deductive process begins with an existing theory, uses this to draw some hypotheses, and through testing these hypotheses tests the theory itself. By contrast, inductive research attempts to gather explanation and meaning through the collection and empirical data analysis.

Table 3.2 illustrates the major differences between deductive, inductive and abductive research approaches in terms of logic, generalizability, use of data and theory (SAUNDERS et al., 2007).

Table 3.2 Differences between dedictive, inductive and abductive approaches (SAUNDERS et al., 2007).

	Deduction	**Induction**	**Abduction**
Logic	In a deductive inference, when the premises are true, the conclusion must also be true.	In an inductive inference, known premises are used to generate untested conclusions.	In an abductive inference, known premises are used to generate testable conclusions.
Generalizability	Generalizing from the general to the specific.	Generalizing from the specific to the general.	Generalizing from the interactions between the specific and the general.
Use of data	Data collection is used to evaluate propositions or hypotheses related to an existing theory.	Data collection is used to explore a phenomenon, identify themes and patterns and create a conceptual framework.	Data collection is used to explore a phenomenon, identify themes and patterns, locate these in a conceptual framework and test this through subsequent data collection and so forth.
Theory	Theory falsification or verification.	Theory generation and building.	Theory generation or modification; incorporating existing theory where appropriate, to build new theory or modify existing theory.

3.3 RESEARCH METHODS

Research methods are generalized and established ways of approaching research questions. Research methods can be either qualitative or quantitative and involve the specific study activities of collecting and analyzing research data in order to answer a particular research question.

The most commonly used terms to differentiate these paradigms with respect to their associated methods and techniques are quantitative and qualitative, respectively, with quantitative methods being based on the positivism paradigm while qualitative methods are built on a phenomenological world view (COLEMAN; O'CONNOR, 2007).

For many researchers, qualitative methods are reserved exclusively for use by interpretivist researchers, and can not to be mixed with quantitative methods or positivist view points. However, in recent decades, researchers in information systems, human–computer interaction, and software engineering have begun using qualitative methods, even though the predominant, implicit philosophical position of these research areas remains positivist (SHULL et al., 2007).

Next, we discuss the main differences between qualitative and quantitative research methods.

- **Quantitative methods:** These are appropriate when factual data are required to answer the research question; when general or probability information is sought on opinions, attitudes, views, beliefs or preferences; when variables can be isolated and defined; when variables can be linked to form hypotheses before data collection; and when the question or problem is known, clear and unambiguous (HAMMARBERG et al., 2016). Some examples of research strategies used in quantitative methods are: experimental, semi-experimental, survey and those based on the analysis of large-scale data sets, such as statistics.

- **Qualitative methods:** In contrast, these methods are used to answer questions about experience, meaning and perspective, most often from the standpoint of the participant. These data are usually not amenable to counting or measuring (HAMMARBERG et al., 2016). They are very good for answering both what and who questions, but not well suited to answering why and where research questions. Some examples of research strategies used in qualitative methods are: case study, action research, ethnography, survey and grounded theory. According to Seaman et al. (2003),"The principal advantage of using qualitative methods is that they force the researcher to delve into the complexity of the problem rather than abstract it away".

Table 3.3 summarizes the differences between qualitative and quantitative research methods.

Table 3.3 Differences between qualitative and quantitative research methods.

Characteristic	**Quantitative research**	**Qualitative research**
Type of data	Phenomena are described numerically	Phenomena are described in a narrative fashion
Analysis	Descriptive and inferential statistics	Identification of major schemes
Scope of inquiry	Specific questions or hypotheses	Broad, thematic concerns
Primary advantage	Large sample, statistical validity, accurately reflects the population	Rich, in-depth, narrative description of sample
Primary disadvantage	Superficial understanding of participants' thoughts and feelings	Small sample, not generalizable to the population at large

3.3.1 Muti-method Research

Qualitative and quantitative research methods are usually selected for the purpose of directing the steps necessary to carry out the research. As we saw in the previous section, any research method includes strengths and weaknesses. However, each one has something to offer according to the research purpose. One way to compensate for individual weaknesses is to use the methods in a integrated way into a pluralistic approach, more commonly called multi-method (GERRING; THOMAS, 2011).

The multi-method research combines two or more research methods within the same project paradigm. These methods work in a complementary way and strictly following its guidelines (CRESWELL; CLARK, 2017).

In this approach, quantitative and qualitative methods *"should not be viewed in opposition, or thought of as a matter of numbers versus words, or a debate about what can and cannot be quantified, but rather from the production of different levels and types of explanation, focusing on differences in how accurate, explicit and broad comparisons and explanations can be"* (GERRING; THOMAS, 2011).

According to Creswell and Clark (2017), some authors have drawn attention to the care of distinguishing 'multi-method studies' from 'mixed method studies'. The distinction between them is that mixed methods combine qualitative and quantitative methods, while multi-methods use qualitative methods.

3.4 OVERVIEW OF THE RESEARCH DESIGN

The research area explored in this study is relatively new and still is in evolution. Therefore, multi-method was the methodological choice considered most appropriate to collect data from different sources to answer the research question. Our research methodology combined primary (survey, grounded theory) and secondary (systematic mapping study) studies. Along the evolutionary trajectory of this dissertation, the positivism was the methodological philosophy adopted to value multiple perspectives and develop understanding about the phenomenon.

Table 3.4 shows the research philosophy, approach, methods and strategies adopted in this study. We combined three qualitative methods with an inductive research approach with the intention of facilitating the research object exploration, identifying patterns and constructing a theory. Next, we briefly present the research strategies used in this study.

Table 3.4 Philosophy, approach and methods adopted

Research philosophy	Positivism
Research approach	Inductive
Research method	Multi-method qualitative
Research strategies	Systematic Mapping Study, Survey and Grounded Theory

- **Systematic Mapping Study**

  Mapping study is an evidence-based approach, applied in order to provide an overview of a research area, and identify the quantity and type of research and results available within it (PETERSEN et al., 2008).

With the purpose of identifying and characterizing the evidence of the Sustainable SE in the literature, the goal of this study was to identify, evaluate, and synthesize the state-of-the-art of the sustainable practices on SE in order to suggest important implications for practice, as well as, identifying research trends and open issues in the field. The mapping study is discussed in details in Chapter 4.

- **Industry Survey**

  Surveys are used extensively in software and systems engineering studies to provide insights into issues, assist with problem-solving, and support effective decision making (KASUNIC, 2005).

  The industrial survey was performed in order to gain a better understanding about the software industry awareness on sustainable SE and the adoption of sustainable practices on software development process in the intention to provide new findings. The survey study is discussed in details in Chapter 5.

- **Grounded Theory**

  GT is a set of procedures that provides the comparative data analysis and able to generate, in a systematic way, a theory based on these data. We presented an initial theory constructed by employing the GT method (GLASER et al., 1968) to characterize the sustainable software development by the concerns identified in the literature and software industry. The GT study is discussed in details in Chapter 6.

3.5 CHAPTER SUMMARY

In this Chapter, we presented main research philosophies, approaches and methods used by the SE community. This dissertation followed the positivist research philosophy and the qualitative multi-method methodological choice combining three research methods. First, a systematic mapping study was performed. Next, an survey study was distributed to software practitioners to confront evidence of lack of industry participation in sustainable SE research. Finally, we used GT method and produced a theory.

Chapter 4

SYSTEMATIC MAPPING STUDY ON SUSTAINABLE SOFTWARE ENGINEERING

Understanding how the concepts of sustainability could be incorporated to the SE concepts, in particular in terms of the SDLC, has gained increasing attention in the last years. Many studies have addressed the impact of sustainability in the SE practice, from a set of perspectives.

This Chapter presents a systematic mapping study that aggregates, summarizes and discusses the results of 75 relevant primary studies concerning methods, processes, tools and metrics to develop software in a sustainable way. The included primary studies were selected using inclusion and exclusion criteria applied to relevant papers published until 2017. The studies were analyzed based on a set of classification criteria, including contribution types, SDLC phases, evidence types, research types, application domains, publication venues, contribution types and research methods.

The results indicated the growing interest of the SE research community in the Green domain, including the focus on mobile environments. There is a need for more studies on techniques, tools and metrics covering the construction, test and maintenance phases. The results also point out that there is a clear view of the SE community on the need for an alignment between research and real-world practice and more evaluation studies. This text was published in full in (MOURÃO et al., 2018).

4.1 INTRODUCTION

Approximately 97% of climate scientists agree that global warming trends over the last century are likely to have been the result of human activities (COOK et al., 2016). Aligned with this concern, the Software Engineering (SE) community has increased its interest in unveiling the impacts of ICT on excessive consumption of natural resources.

Such a role has been played by the Green and Sustainable SE field, Green SE for short. Green SE aims to create reliable and durable software that meets users' needs while reducing environmental impacts (AMSEL et al., 2011). It consists of practices that enable linking Software Engineering to the principles of sustainability (CALERO; PIATTINI, 2017). In this sense, the development, operation and maintenance of the software, when carried out in a sustainable way, might lead to the production of greener software products and services.

Although current discussions state that sustainability should be considered in software development process, actuals models and quality standards, such as ISO 9126 and ISO 25010, do not consider sustainability as a quality attribute. However, there are already reports of studies in the SE literature (CALERO; BERTOA, 2013; VENTERS et al., 2014b; BECKER, 2014; PENZENSTADLER et al., 2014a) that discuss the relationship between software quality and sustainability. These authors agree with the need to think sustainability as a new attribute of quality and investigate how this new attribute can be incorporated into the software development process in terms of energy efficiency and resource efficiency.

With the intention of conducting researches future in the field of sustainable SE, we need to have a broad understanding of the field and to have a clear view of the SE research community's discussions/investigations on how to achieve sustainability in the SDLC. For this, we carried out a systematic mapping study in order resume theses approaches and get a starting point. This Chapter also highlights the gaps and identifies research opportunities. Moreover, it is based on analysis of interesting issues, guided by a set of research questions.

The applied research method was based on the (PETERSEN et al., 2008) approach and we elaborated a research protocol, based on the (BARBARA; STUART, 2007) model. 8 research questions were defined. Data collection included the application of inclusion and exclusion criteria. We carefully selected 4,912 candidate primary studies that were extracted from 6 digital databases and published until 2017. This process resulted in the identification of 75 relevant primary studies. These studies were analyzed and classified according to quality criteria applied. Finally, we describe the classification scheme, the extraction of the data and the analysis of the results.

The remainder of this Chapter is organized as follows: Section 4.2 discusses related work. Section 4.3 describes the applied systematic mapping study methodology. Section 4.4 describes the data collection. Section 4.5 describes the classification schema. Section 4.6 reports and discusses the results of this systematic mapping study. In section 4.7 the threats to validity are described. Finally, in Section 4.8, we draw concluding remarks and point out opportunities for further research.

4.2 RELATED WORK

We identified several studies reporting on literature reviews or surveys aimed to gather and evaluate existing research on Green SE.

Penzenstadler et al. (2014b) reported on a SLR on SE practices for sustainability which provided an overview of 83 existing research published prior to 2013. The study analyzed the existing literature from six perspectives, as follows: research topics, models and methods used in practice, research type, application domains, research groups most active and distribution between academics and practitioners. However, this research paper covered a specific point of sustainable SE by directing research into models and methods only. Furthermore, selected studies published until 2013.

Berntsen et al. (2016) performed a SLR with the purpose of outlining the state-of-the-art of Sustainable SE with a focus on existing models, guidelines and practices, as well as proposals in this regard. However, this study selected 36 works until 2010 and considered three research questions, only. It investigated the most cited/reported guidelines and models in Sustainable SE, the evolution of interest and the most important authors and venues on this topic.

García-Mireles (2016) performed a SLR to investigate software process improvement approaches focused on sustainability. This paper analyzed 7 primary studies published prior to 2016. Four research questions was investigated: (1) publication trends in regards to sustainability enhancement based on software process improvement; (2) research approaches; (3) software processes; and (4) the main features of identified approaches to improve sustainability in software processes. However, this paper researched specifically sustainability from a software process improvement perspective.

Anwar and Pfahl (2017) reported a SLR in the field of Sustainable SE, focusing on the role of software analysis. There were classified 50 studies between 2015 and 2016 to respond 6 research questions: sub-domains of Sustainable SE, contribution types, research types, role of software analysis and the potential for future research in the area. However, this research paper focus specifically in software analysis and cover two year only.

Marimuthu and Chandrasekaran (2017) summarizes in SLR the body of knowledge of methods for green and sustainable software development and provides a platform for conducting future research. They analyzed 82 primary studies published between 2010 and May 2016 and defined 7 research questions: type of research, research goals, research topics, research contributing, type of validation methods, tools proposed and publication venues. However, this research paper did not used digital libraries, only snowballing method and popular publication venues to find out the relevant studies.

Wolfram et al. (2017), in turn, presented the state-of-the-art of sustainability in the context of SE. 168 publications of 1980 until 2013 were classified. However, although of this research paper investigates how sustainability is defined in the context of SE, it defines only 3 research questions: sub-disciplines of SE, time scope considered in the definition and the definition develop over time.

García-Mireles et al. (2017) carried out a SLR aimed to provide an overview of the approaches found in the literature for dealing with interactions between software product quality and sustainability in the context of application software. In order to accomplish that, the authors classified 66 primary studies and answered two research questions about profile of papers and specific interactions between environmental sustainability goals and product quality characteristics. However, this research paper did not investigate full sustainable SE domain.

In order to develop this Chapter, we considered every mentioned study, since they bring relevant information. However, we observed that these SLR's considered publications until 2016. Furthermore, that studies explore the specific issues such as software analysis, practices, models, methods and software process improvement. In this work, we defined eight research questions to be answered on intention to cover the full Sustainable

SE area, not only specific aspects. We considered the primary studies published until 2017. As the research in this field is incipient, it becomes important to observe the state-of-the-art considering recent publications. In addition, we apply a differentiated approach to select primary study, where we have mixed digital libraries and *Snowballing* method. Finally, in order to carry out critical analyzes and evaluations, we apply a rigorous quality assessment criterion based on Dyba et al. (2007).

4.3 METHODOLOGY

This Chapter was developed based on the systematic process proposed by Petersen et al. (2008). Figure 4.1 illustrates the process steps. The execution of each phase will be explained in the following sections.

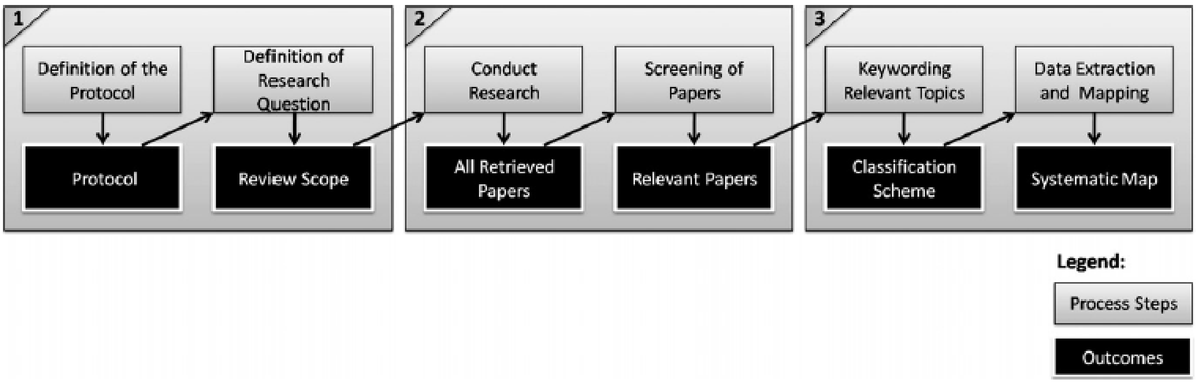

Figure 4.1 Systematic Mapping Process.

4.3.1 Protocol Definition

A protocol is a important resource to help researchers in plan, execution and replication of an empirical study. To guide this MS, a research protocol was developed with the support of the tool Start [1].

In this Chapter, the focus was to investigate the domain of Sustainable SE in the context of software, specifically: methods, models, processes, life cycle, tools, methodologies and metrics. Therefore, the scope of this research does not include hardware-related issues.

The Table 4.1 details the protocol defined in this search. The PICOC structure (Population, Intervention, Comparison, Outputs, Context) proposed by Barbara and Stuart (2007) was used.

The "Comparison" criterion was not applied, since it was not the purpose of this study to make any comparison.

[1] Start ⟨http://lapes.dc.ufscar.br/tools/start_tool⟩

Table 4.1 PICOC structure.

Population	Papers published in the area of Software Engineering.
Intervention	Papers that present proposals in the area of Green Software Engineering.
Comparison	Not applied.
Outcomes	1. Emerging Domains of Sustainable Software Engineering; 2. Lack of evidence and research opportunities in Sustainable SE.
Context	Software (methods, models, process, life-cycle of development, tools, methodologies and metrics).

4.3.2 Research Questions

To guide this MS, the following main research question has been defined: **"What are the existing software approaches to support Sustainable SE?"**

To answer the main question, we derived eight following sub-questions. These are detailed below, with respective rationale.

SB-Q1: What types of SE contribution have been investigated? This question is intended to investigate / determine which types of contribution among traditional SE processes, methods, tools and metrics have been investigated in light of the Sustainable SE domain. We used the contribution types defined by Petersen et al. (2008) which are: Process, method, tool, metric and model.

SB-Q2: If so, what stage of the SDLC have the contributions raised been applied? This question is intended to identify where the sustainability efforts were dedicated throughout the SDLC. The steps adopted in this study are those provided in SWEBOK - *Software Engineering Body of Knowledge* - [2] which are: requirements, design, construction, testing and maintenance.

SB-Q3: What types of evidence have been identified? This question seeks to identify which types of evidence are most applied in empirical studies. The levels of hierarchy of evidence suggested by Barbara and Stuart (2007), which are: No evidence, evidence obtained from demonstration or elaboration of toy examples, evidence obtained from expert opinions or observations, evidence obtained in academic studies, evidences obtained in industrial studies and evidences obtained from the industrial practice.

SB-Q4: What types of research were conducted? This question aims at identifying the research approaches that were presented in the primary studies, based on the classification scheme proposed by Wieringa et al. (2006), which consists of evaluation, proposal of solution, validation, philosophical, opinion article and article experience.

SB-Q5: What search methods are available? The objective of this research question is to analyze, within the research methods applied by the primary studies, which are the most used approaches to evidence the efforts of Green SE based on Castellan (2010), which defined as: case study, theory grounded, survey, meta-analysis.

SB-Q6: Which application domains were considered? The objective is to investigate which application domains are being searched. (Example: Mobile, Cloud, IoT, Embedded Systems, etc)

[2] https://computer.org/web/swebok/v3

SB-Q7: Which publication locals are most commonly used? The intent is to identify which publishing locals have accepted articles on Sustainable SE.

It is believed that this data will help researchers identify the main place of publication and find relevant studies in the field.

SB-Q8: What is the distribution between academia and industry? The goal is to investigate where research efforts are being directed at Sustainable SE, whether for academia, industry or both.

4.4 DATA COLLECTION

In order to answer the research questions, data was collected from the research literature. In this section, we present the activities to implementation of systematic mapping, including search strategies, identifying data sources, inclusion/exclusion criteria, and the selection process.

4.4.1 Search Strategy

Initially the survey of the keywords and synonyms related to the research question was carried out. The selection of the set was defined based on the guidelines proposed by Barbara and Stuart (2007). These, systematically determine a set of keywords that most represent the object to be investigated. The study used in this MS as a reference for surveying the keywords was performed by Anwar and Pfahl (2017).

In addition, the keywords used by the International Workshop on Green and Sustainable Software (Greens) were consulted in their summary of the call for papers. At the end, the set formed was: **Software, Green, Sustainability, Sustainable, Energy, Eco and Power.**

After the definition of the keywords, a combination of these was made, in order to obtain the most satisfactory results. The criteria used to obtain a relevant search string were: greater number of results retrieved from the bases and articles strongly related to the topic. Eleven (11) combinations were carried out until the *string* was considered adequate for the search, being defined as follows:

((“green software engineering”) OR (“software engineering” AND “eco-sustainability”) OR (“sustainable software engineering”))

A calibration of the selected string was performed by executing it on the selected bases. The objective was to verify the quantity and quality (articles strongly related to the research topic) of the returned works as the units were concatenated with the “OR” and “AND” logical operators.

For the purpose of verifying and testing the quality and coverage of the selected *string*, a cross between the papers retrieved by the validated string and the papers provided by Greens, the workshop of ICSE, was also carried out. The choice of the Greens was due to its importance in the SE community, since it has a trail dedicated to the theme of this mapping. Table 4.2 shows the quantity and percentage of articles retrieved from search repositories and also published in the Greens.

Table 4.2 Number of articles recovered.

Repository	Total papers Greens	% covering Greens
1. IEEEXPLORE	5	16
2. ACM DL	1	3
3. SPRINGERLINK	0	0
4. SCIENCE DIRECT	0	0
5. WILEY ONLINE LIBRARY	0	0
6. GOOGLE SCHOLAR	3	9

4.4.2 Data Sources

In order to establish a set of search repositories that would satisfactorily illustrate the area of interest of this work, a survey was made of the most used / relevant bases in Computer Science based on the study carried out by Nakagawa et al. (2017). We considered publications retrieved from digital libraries web search engines. They are: IEEEXPLORE, ACM DL, SPRINGERLINK, SCIENCE DIRECT, WILEY ONLINE LIBRARY and GOOGLE SCHOLAR.

We restricted the search to studies published up to December 2017. An inferior year-limit was not defined, in order we will identify when research community begins to cite the term Sustainable SE. Reference papers were retrieved in the SE area whose access to authors was possible in the research environment.

The search was also performed using the *snowballing* process, following up the references in papers retrieved of the repositories.

4.4.3 Studies Selection

After the execution of the search strategy, it was necessary to establish inclusion and exclusion criteria to select the articles in the resulting set. These studies will be subsequently classified and will continue to the quality verification stage.

The inclusion and exclusion criteria take into account the quality of the article in terms of language, type, year of publication and relation with the topic of this research.

It is intended to focus only on studies that address processes, methods, metrics and tools in the context of software related to Sustainable SE. A study will be selected for the next step if it meets all the inclusion criteria, and will be discarded if it meets only one of the exclusion criteria. The table 4.3 presents the criteria defined in this research.

The conduction of the selection stage of the study occurred in three phases, as shown in Figure 4.2.

- **Phase 1.** It consisted of the application of the automatic search on the digital libraries. The particularities in the search syntax of the repositories made it impossible to execute a standard string. It was therefore necessary to prepare a specific

Table 4.3 Exclusion and inclusion criteria.

Criteria	**(I)nclusion / (E)xclusion**
Studies related to Sustainable SE	I
Latest version of paper published	I
Papers published to 2017	I
Papers in English	I
Studies focused on Green, without bias in SE	E
Gray literature (theses, dissertations, reports, reports of experiences, short paper, books, magazines, expanded abstracts, opinion articles, workshops, courses, tutorials).	E
Secondary and tertiary studies	E
Duplicate papers	E

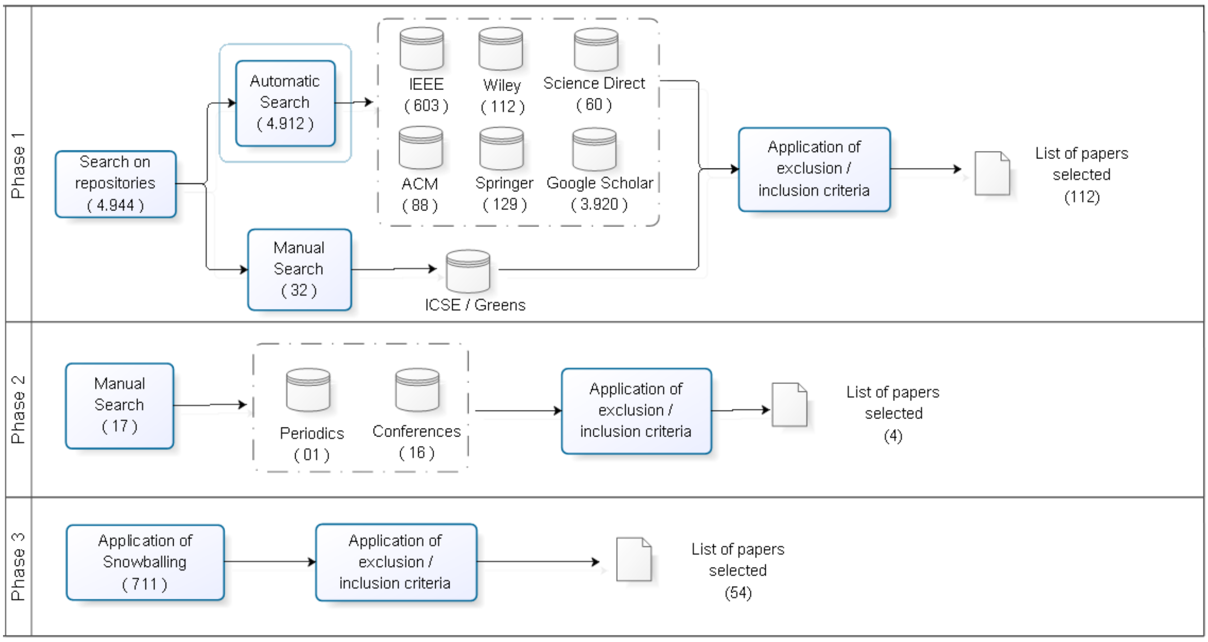

Figure 4.2 Selection process of primary studies.

syntax for each base. The table 4.4 displays the strings used and their study numbers retrieved from the bases. On the Springer Link database, Computer Science articles from the Software Engineering sub-discipline, written in English and published until 2017, were filtered. For Science Direct, the filters were applied: Year of publication (up to 2017) and Type of article“Research articles” and “Review articles”). In the other databases, the default string was run in Advanced Search, and no specific filters were required.

The automatic collection recovered 4.912 primary studies. The manual collection of articles was done only in the GREENS workshop, where 32 studies were retrieved.

Of the 4.944 studies retrieved, the inclusion and exclusion criteria were applied. As Figure 4.2 shows, at the end of this phase, 112 articles remained.

- **Phase 2.** The manual search was expanded and applied on the main conferences of the area in SE: SBQS, ICSE, GREENS e SBES. Another 17 studies were found on Greens, only. After the application of the inclusion and exclusion criteria, they remained 4.
- **Phase 3.** The *Snowballing* procedure was applied to articles retrieved from phases 1 and 2, adding 771 recovered primary studies. On these, the inclusion and exclusion criteria were applied, obtaining a total of 54 articles selected.

Table 4.4 Strings applied and papers recovered.

Repository	String	Papers
1.IEEEXPLORE	("software engineering" OR "software") AND ("green" OR sustain* OR energy* OR "eco" OR "power")	603
2.ACM DL	(("software engineering") AND (sustain* OR green* OR energy* OR "resource optimization"))	88
3.SPRINGER LINK	("software engineering" AND sustain OR green OR energy OR resource optimization)	129
4.SCIENCE DIRECT	("software engineering" OR "software") AND (sustain OR green OR energy OR resource optimization)	60
5.WILEY ONLINE LIBRARY	"Green Software Engineering" in Abstract OR "sustainable" in Abstract OR "sustainability" in Abstract OR "Energy Efficiency" in Abstract OR "Energy Consumption" in Abstract OR "Green"	112
6.GOOGLE SCHOLAR	(("green software engineering") OR ("software engineering" AND "eco-sustainability") OR ("sustainable software engineering"))	3.920
TOTAL:		**4.912**

At the end of the paper selection process, the list of primary studies selected for full reading counted 170 publications.

4.4.3.1 Reliability of inclusion decisions Each paper was analyzed by the author of this dissertation and a master student individually. We read the title, abstract, keywords and introduction (when necessary) and applied the inclusion and exclusion criteria. Before this procedure, it was verified the need to define a method for prevention/treatment of possible divergences.

For this purpose, the Likert Scale[3] was used, showed in Table 4.5. Papers marked with the “Totally Agree” scale were accepted and those marked with “Strongly Disagree” were deleted. Articles marked with ”Partially Agree”, “Partially Disagree” or “Neutral” were discussed between the two researchers to decide whether they would be included or excluded from the final set. Without consensus, the advisor of this dissertation acted with the analysis and the final decision making.

To ensure the reliability of agreement among researchers during the selection of primary studies, this mapping applied the method “Inter-rater reliability” [4]. This procedure measures the agreement between the researchers in the evaluation of the studies. The achievement of the percentage agreement measure occurred with the application of the scores, varying from -2 to 2, in each of the levels of the Likert Scale, from the level strongly disagree until strongly agree. Subsequently, the scores of the first researcher were subtracted from the notes of the second researcher and the number of zeros obtained was counted. Finally, the number of zeros was divided by the total number of studies. The percentage agreement obtained was 70%, which corresponds to a ”Strong agreement” level.

Table 4.5 Likert Scale.

Scale	**Description**
Strongly agree	It should be granted in the case where the paper presents in the text, the criteria that fully address the issue.
Agree	It must be granted in the case where the paper meets partially the criteria of the question.
Neither agree nor disagree	It should be granted in the event that the paper does not make it clear whether or not it answers the question.
Disagree	It should be granted in the case where the criteria contained in the question are not met by the evaluated paper.
Strongly disagree	It should be granted in the event that there is nothing in the paper that meets the criteria of the question.

4.4.4 Quality Evaluation

The application of quality criteria is performed to try to ensure that the final set contains the most related primary studies possible with the purpose of the mapping. This research used the approach made by Dyba et al. (2007) whose set of issues addresses the main quality problems, namely: reporting, credibility, rigor and relevance.

The articles were read in full by two researchers, who worked individually, applying a note to the quality issues described in Table 4.6. For each of the eleven questions defined, a note within the range [0.0, 0.5 and 1.0] was applied according to criteria:

- The publication does not meet the quality criterion.

[3] Likert Scale: ⟨https://en.wikipedia.org/wiki/Likert_scale⟩
[4] Method inter-rater reliability: ⟨https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/⟩

- The publication partially meets the quality criterion.
- The publication satisfies the quality criteria.

It was also defined that if the study met the exclusion criterion, it would be removed. Therefore, throughout the reading process, 95 articles were excluded and 75 were evaluated and sent to the classification phase.

Table 4.6 Quality Criteria.

Question	**Criteria**
1. Is the research related to any Green Software Engineering domain?	REPORT
2. Are the objectives clearly stated?	REPORT
3. Is the research context clearly expressed?	REPORT
4. Is the phase of the developmental life cycle covered by the study clearly stated?	REPORT
5. Is the application domain clearly expressed?	REPORT
6. Are the results, limitations and future work clearly described?	CREDIBILITY
7. Is the contribution clearly expressed?	CREDIBILITY
8. Is it possible to identify the place of publication of the survey?	CREDIBILITY
9. Is the type of research conducted clearly expressed?	RIGOR
10. Is the research method clearly expressed? In the case of non-empirical research, are the arguments adequately presented?	RIGOR
11. Does the research make clear who contributes?	RELEVANCE

At the end, the points were summarized by papers. The accepted studies were those that obtained the percentage of at least 63.64 % of the maximum predicted mark (11.0). 100 % of the articles evaluated reached the minimum score, an average score of 81.82 %. All the studies are listed in Appendix A.

4.4.5 Data Extraction

The data extraction forms must be designed to collect all the information needed to address the research questions and the quality criteria (NETO et al., 2011). The following information was extracted from each study: title, authors, publication year, publication venue, domain of SE, phases of SDLC, kind of evidence, research type, application domain, contribution type, research method and quality criteria score.

4.5 CLASSIFICATION SCHEME

In this section, we describe the classification scheme and the results of data extraction.

The classification scheme of the accepted primary studies was structured in eight different categories. These were defined based on the eight research sub-questions defined, in order to answer them. A brief description of the classifications is given below.

- **Classification of contribution types**

  The types of contributions proposed by the primary studies were defined as Petersen et al. (2008). They are: process, method, tool, metric and model. Throughout the readings, other types of contributions have been identified and added to the selection list, they are: framework, catalog, guide, technique and approach.

- **Classification of software development lifecycle phases**

  This work followed the classification suggested by Software Engineering Body of Knowledge (SWEBOK) - Software Engineering Body of Knowledge. Each selected study was categorized into one of the following SE topics:

  - **Software requirements:** Jobs that collect and specify requirements related to sustainability.
  - **Software design:** Articles that care about the energy usage patterns of high-level, low-level design decisions.
  - **Software Building:** Works that discuss software and programming libraries with energy efficiency or power recognition.
  - **Software Testing:** Articles that seek to find and correct problems related to software power.
  - **Software Maintenance:** Works that discuss the role and impact of power usage during the software maintenance phase.
  - **All:** Works that address all stages of the software development life-cycle.

- **Classification of types of evidence** To classify the studies as to the type of evidence, the levels of evidence hierarchy suggested by Barbara and Stuart (2007) were used. They are: No evidence, Evidence obtained from demonstration or elaboration of examples of toys, Evidence obtained from expert opinions or observations, Evidence obtained in academic studies, Evidence obtained in industrial studies and Evidence obtained from industrial practice.

- **Classification of research types** The primary studies were classified according to the type of research used, according to the approach, below, proposed by Wieringa et al. (2006).

  - **Evaluation:** Researches that evaluate and report the research works that are implemented in practice. Generally, industrial case study, controlled experiments, researches of professionals, research of action and ethnography can be categorized in this category.
  - **Solution:** Propose a new technique or improvement of existing techniques with proof of concept in the form of a small example or good arguments.
  - **Validation:** Research that investigates/evaluates the proposed solutions or new techniques (not implemented in practice) through simulations, laboratory experiments, mathematical analysis and proof.

  - **Philosophical:** Articles that describe a new way of viewing the existing field or new conceptual structures.
  - **Opinion articles:** Express the author's opinion on the good and evil of certain techniques or tools.
  - **Experience papers:** Present the author's personal experience (eg, professional) about the use of a particular technique or tool.

- **Classification of application domains** The classification of the studies with respect to the field of application was made as they were identified in the readings and made available in the selection list for the categorization of the following articles. Examples of application domains are: Mobile Applications, Cloud Systems, IoT, Embedded Systems, etc.

- **Classification of publication local** The study followed the simpler classification for the types of publication sites. The articles selected were categorized among periodicals, conferences, symposia and international workshops, among other relevant events in the SE area. At each new identification, the event was recorded in the set of publication locations to be selected in the following classifications.

- **Classification of contribution types** The classification used to classify the primary studies according to their contribution was:

  - **Academy:** The contribution of the study is intended for researchers.
  - **Industry:** The contribution of the study is intended for industry professionals.
  - **Both:** The contribution of the study is intended for both researchers and industry professionals.

- **Classification of research methods**

  Based on Castellan (2010), the studies were classified into:

  - **Case study:** They provide an in-depth understanding of how and why certain phenomena occur and can reveal the mechanisms by which cause-effect relationships occur.
  - **Grounded Theory:** Inductive process of generation or discovery of a theory or scheme from the coding and categorization of data.
  - **Survey:** Selection of a representative sample of a well-defined population, and data analysis techniques used to generalize from that sample to the population, usually to answer basic rate questions. They can be conducted using structured interviews or data recording techniques.
  - **Meta-analysis:** Used to select previous studies within a domain for analysis. This research method is generally used in studies that summarize the state of the art or the state of practice.

In addition to the categorization of (CASTELLAN, 2010), with the reading of the primary studies, it was possible to identify another set of Easterbrook et al. (2008). They are:

– **Controlled Experiment:** Investigating a testable hypothesis in which one or more independent variables are manipulated to measure their effect on one or more dependent variables.
– **Multi-method approach:** Also characterized as mixed-method research - a more complex research strategy that has emerged in the recognition that all methods have limitations, and the weaknesses of a method can be offset by the strengths of other methods.

The non-empirical theoretical research presents itself as an approach that is independent of practical application, it is characterized as methodology or theoretical basis of empirical research.

The classification procedure of the studies was performed after careful reading of each of the 75 studies. The classification was done by extracting the relevant data using a spreadsheet. The list of selected studies and the data collection form is available in Annex.

4.6 RESULTS

In this last step of the process, the data were extracted and interpreted in order to express the state-of-the-art in the field of Sustainable SE. The results of this study were reported for each of the research sub-questions, SB-Q1 to SB-Q8.

SB-Q1: What types of SE contribution have been investigated?

The mapping identified ten different types of proposal, as can be seen in Figure 4.3.

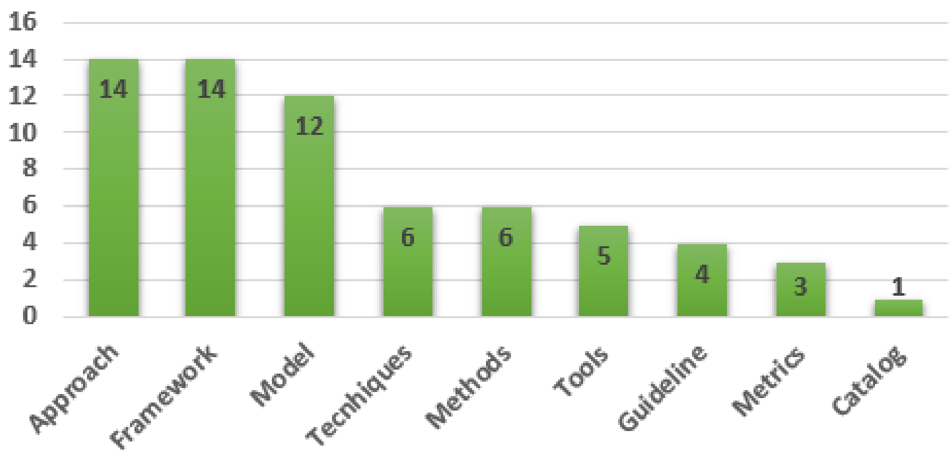

Figure 4.3 Number of papers by type of contribution.

The contributions are distributed as follows: frameworks (19%), approaches (19%), models (16%), techniques (8%) guidelines (5%), metrics (4%) and catalogs (1%). It was not possible to identify the contribution in 13% of the selected studies.

SB-Q2: If so, at what stage of the software development life-cycle were the contributions raised applied?
The intention was to identify which were the most interesting phases of the researchers in the last fifteen years. The result was: requirements (24%), design (21%), coding (11%), test (8%) and maintenance (4%). 19% of the studies performed at all stages of the life cycle and 13% did not leave evidence of action in any of the phases.

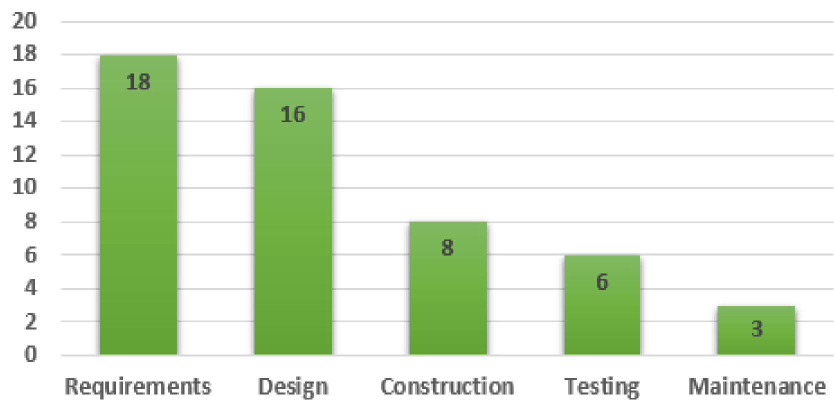

Figure 4.4 Quantity of papers by phases of the life cycle.

By crossing the contribution types with the software development life-cycle phases (Figure 4.5), it was observed that the approach proposals meet the design phase (50%), followed by requirements (25%), the entire life cycle (17%) and construction (8%). Frameworks, in turn, are more targeted to the requirements phase (31%) and to the life-cycle (31%) as a whole, followed by design (23%), test (15%), and maintenance (15%).

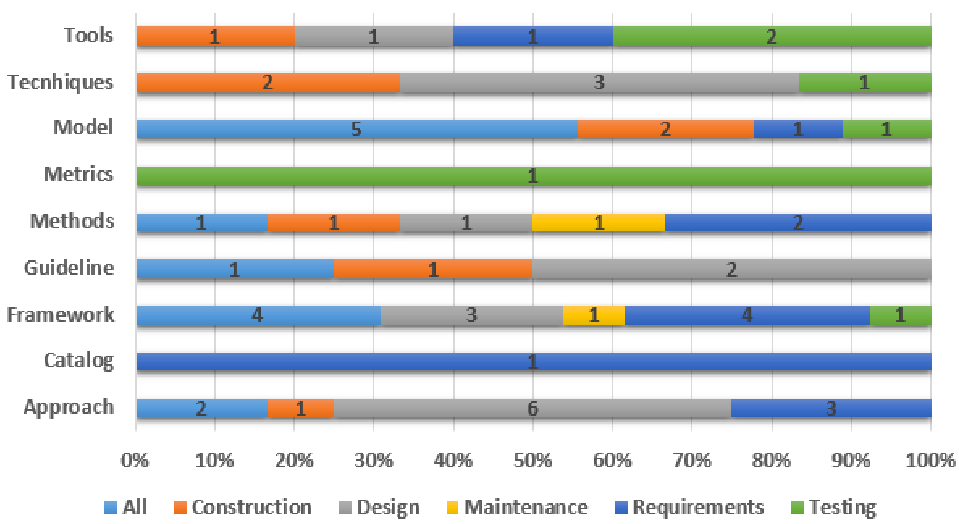

Figure 4.5 Type of contribution x Software life- cycle.

SB-Q3: What types of evidence have been identified?

About the evidence produced by the primary studies, as shown in the figure, 88% of the studies obtained evidence from academic studies (49%) and industry studies (39%). The others sought evidence from the opinion of experts (8%) and industry practices (4%).

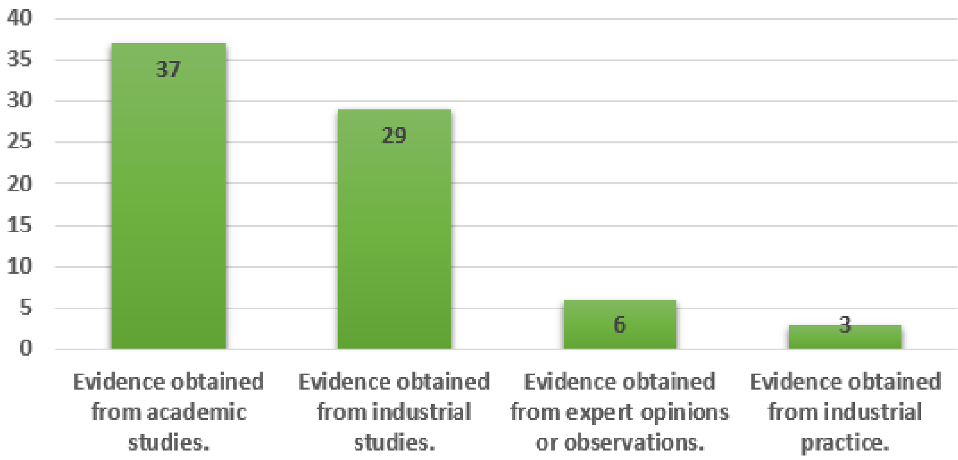

Figure 4.6 Number of papers by type of evidence

SB-Q4: What types of research were conducted?

On the most important types of research, the figure shows that most of the primary studies focused on a solution proposal (68%), followed by evaluation studies (21%) and philosophical research (3%). It can be noted that of the six types of research proposed by Wieringa et al. (2006) and used in the classification of studies, three were not cited: opinion, validation and experience.

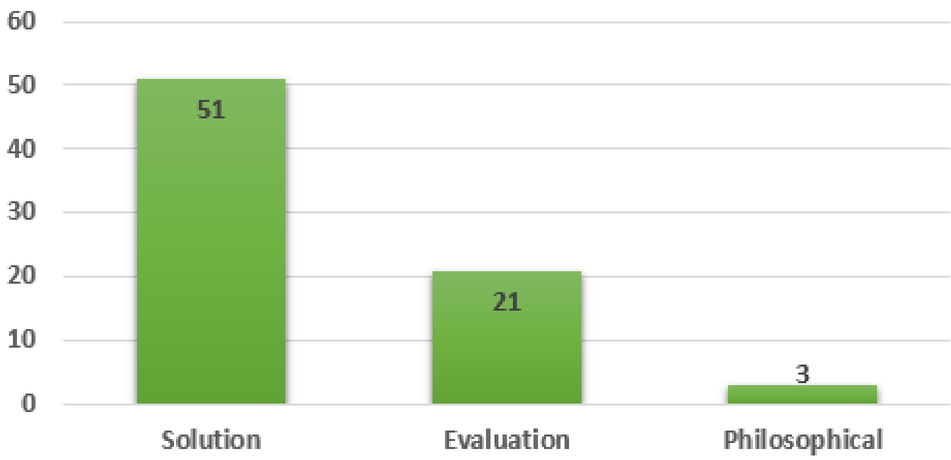

Figure 4.7 Number of articles by type of search

SB-Q5: What search methods are available?

Figure 4.8 shows the findings in relation to the most commonly used search methods in the Sustainable SE research community. It can be observed that 59% of the publications carried out case studies (35%) or controlled experiments (24%). 31% of the studies were classified as follows: Non-empirical research (11%), Multi-method (7%), Survey (7%) and Meta-analysis (4%). The remaining 13% did not specify the use of a search method.

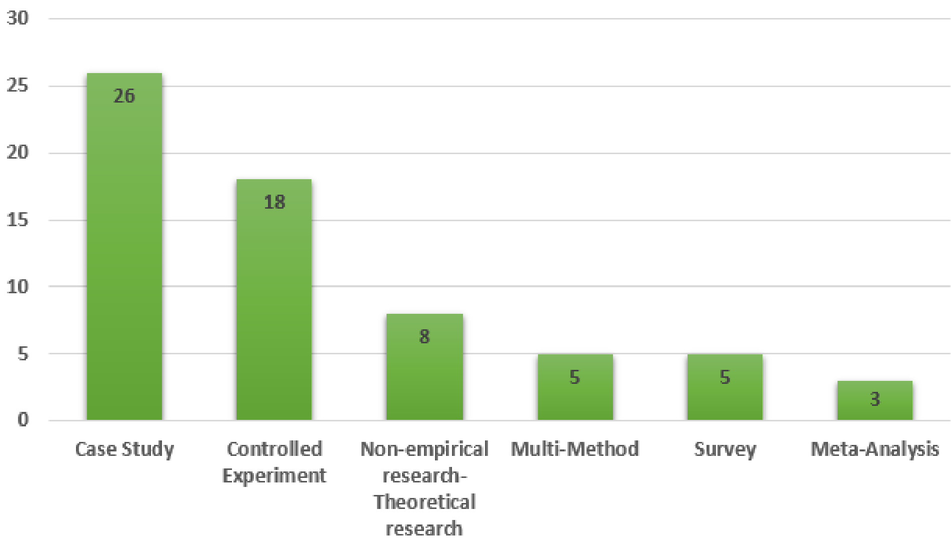

Figure 4.8 Number of papers per search method

From a perspective of which types of research are most adopted by research methods, it can be observed in Figure 4.9, that both the controlled experiment and the case study are applied, in the same proportion, in proposals of solution (75%), followed by evaluation studies (25%).

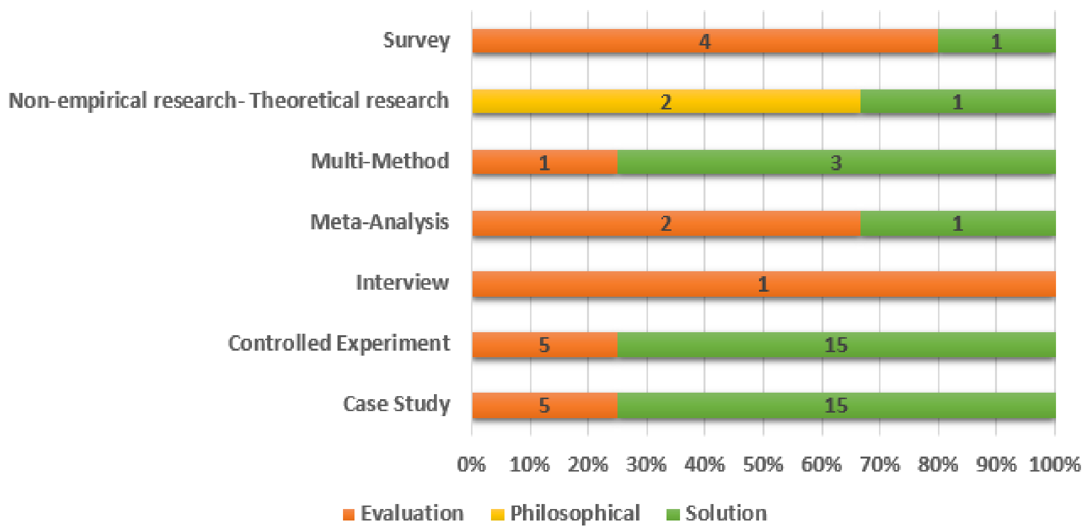

Figure 4.9 Types of research x Research methods

SB-Q6: Which application domains were considered in the evaluation?
As can be seen in Figure 4.10, of the selected studies, only 24% addressed some application domain. From this set, the majority (61%) sought evidence in mobile systems, followed by applications in the cloud (22%). (18%) focused on distributed systems, legacy systems and IoT, in the same proportion.

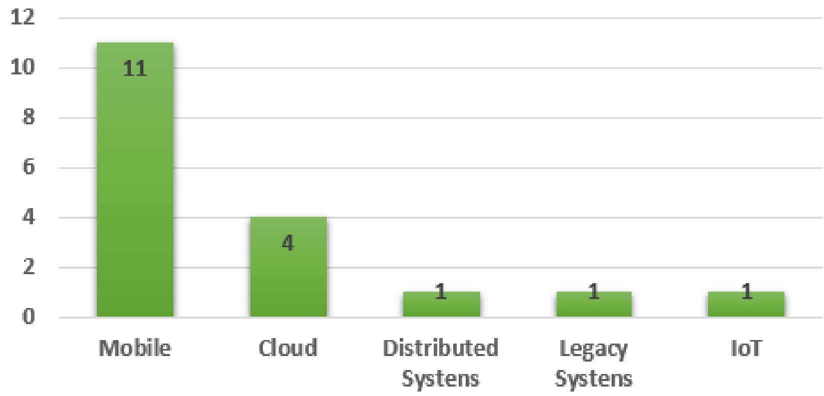

Figure 4.10 Number of papers per application domain

SB-Q7: Which publication venue are most commonly used?
In analyzing the most popular publication locals in researches focusing on Sustainable SE, it is possible to observe, through Figure 4.11, that the highest concentration of the selected articles in this mapping study is in ICSE and Greens (13,5%, each). The others (73%) are distributed in several other conferences and periodicals. In all, 41 different publication sites have been identified, 41% of these with only one published article.

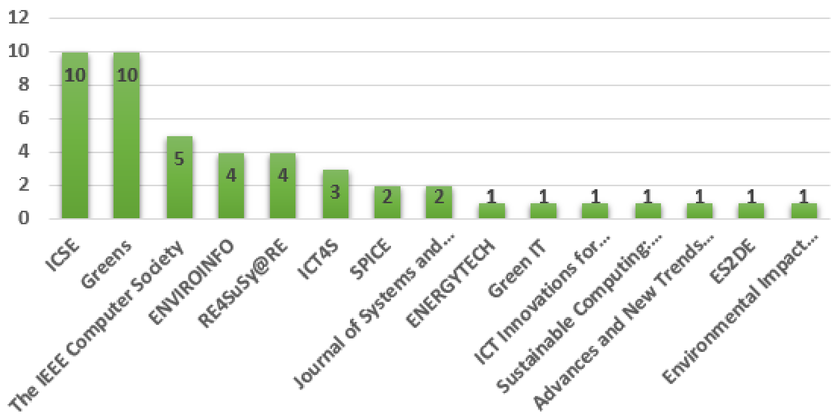

Figure 4.11 Number of papers by place of publication

In relation to the growth in the number of publications in recent years, all studies are classified between 2003 and 2017, with the majority concentrated in the last five years. As Figure 4.12 shows, there has been a low number of publications by 2012, followed by a significant increase from 2013. Specifically this year, the number of publications has doubled, a considerable drop in 2016, followed by a further increase the following year.

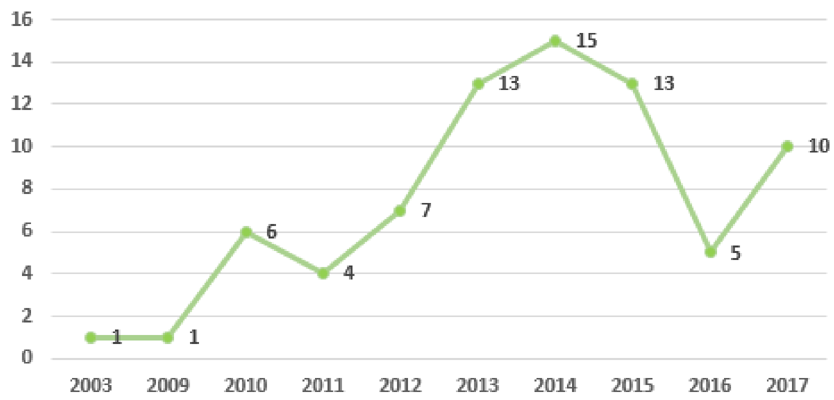

Figure 4.12 Number of papers per year.

SB-Q8: What is the distribution between academia and industry?
As Figure 4.13 shows, most searches have industry input (56%). The contribution of the academy is comparatively smaller (5%). The studies also revealed that the contribution of both industry and academia is 20%. It was not possible to identify the type of contribution in 19% of the primary studies.

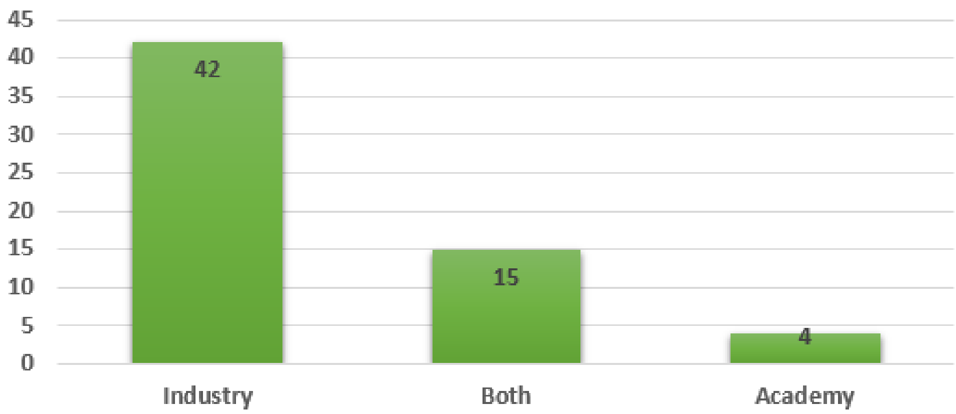

Figure 4.13 Number of papers by type of contribution

4.6.1 Analysis of the results and mapping studies

The analysis of the results enables us to present a number of studies corresponding to each question of research directed in this study.

SB-Q1: What types of SE contribution have been investigated?
Initially, the SE approaches that concern the exploited domain will be treated and analyzed, which can be observed in Figure 4.3 of SB-Q1.

It is worth mentioning that there is a lack of conceptual standard to characterize the types of contribution in SE. This may be evidenced in the studies Anwar and Pfahl (2017), Marimuthu and Chandrasekaran (2017), García-Mireles et al. (2017), García-

Mireles (2016), which use the terms "topics", "approaches" and "sub-domains" as synonyms for such a definition.

Such research emphasizes that most types of SE contribution need in-depth studies. Such investigations could improve understanding of how they could affect energy consumption to encourage the development of sustainable software without losing its quality.

In our study, we observed that the works are mainly defining their research products as: frameworks and approaches. However, this leads us to speculation about the researchers' understanding of how to categorize their proposals, within the set of types of contributions provided in the traditional SE, as defined in SB-Q1.

SB-Q2: If so, at what stage of the software development life-cycle were the contributions raised applied?

The results of Berntsen et al. (2016), García-Mireles (2016), Anwar and Pfahl (2017), Marimuthu and Chandrasekaran (2017), García-Mireles et al. (2017), Wolfram et al. (2017) demonstrate that the SE community is investigating the interaction between sustainability and SDLC. Linking SB-Q1 in this study, we observed that the proposed approaches, frameworks and models seek to address issues related mainly to the design and requirements phases, followed by codification and test, the latter in the same proportion, as pointed out by the authors.

In addition, García-Mireles et al. (2017) points out the need to characterize the impact of the interaction between software quality and sustainability and to create methods and practices to support SDLC.

SB-Q3: What types of evidence have been identified?

Regarding the types of evidence, our study presents a perspective that observes the academic (simulated) and industrial (real) scenarios used by the research. Figure 4.6 shows a categorization where it is possible to verify that 49% of the primary studies use simulated scenarios to carry out their investigations. On the other hand, 39% use the real scenarios. These numbers demonstrate that the area is incipient and relies mainly on theoretically identified evidence, which needs to be evaluated in practice.

Penzenstadler et al. (2014a), García-Mireles (2016), García-Mireles et al. (2017), Anwar and Pfahl (2017) noted the area needs contributions from industry professionals. The participation of professionals should be encouraged, they can contribute with their experiences and observations of the real world, allowing a better prioritization of the research objectives. Although the industry recognizes this need, there is little evidence in practice.

SB-Q4: What types of research were conducted?

Penzenstadler et al. (2014a), Anwar and Pfahl (2017), Marimuthu and Chandrasekaran (2017), unveil the need for assessment validation and opinion research by industry experts and practitioners to help better understand the current industry needs for SE Green. García-Mireles (2016), García-Mireles et al. (2017) show that the lack of empirical data becomes a constraint for the application of solutions proposals in the industry. The data raised by our research point to the need for evaluation and experience studies, since most of the studies portray proposed solutions, which as previously mentioned, suggest that current solutions are still immature and that the research community is still young.

SB-Q5: What search methods are available?

In addition to the previous analysis, when correlating the types of research with the research methods, it is possible to observe that there is a predominance of a solution proposal, where in the majority of the primary studies analyzed (59%) applied both the case study and experiment controlled. This ratifies again that the area is in the growth phase.

SB-Q6: Which application domains were considered in the evaluation?

We also observed that researchers (61% of studies retrieved) focused their research efforts on the mobile application domain. It can be inferred that the preference for this domain occurs due to the recent prevalence of mobile application market growth.

SB-Q7: Which publication venue are most commonly used?

Regarding the publication venue, in agreement with the works of Marimuthu and Chandrasekaran (2017), García-Mireles et al. (2017), there is a trend of publication in a specific sustainable development workshop, the "Greens". In addition, Penzenstadler et al. (2014b), García-Mireles et al. (2017), Marimuthu and Chandrasekaran (2017) note that there is a small distribution of publications in periodicals when compared to conferences and workshops. Penzenstadler et al. (2014b) further states that distribution in a variety of journals is a consequence of the fact that the research community is still forming.

SB-Q8: What is the distribution between academia and industry?

Our findings confirm that the community remains interested and has focused on the problems of the industry, this can be observed in figure 14. Despite this observation, the analyzes carried out regarding the evidences and types of research applied, one notices the necessity of participation of industry professionals to align the research focus.

4.7 THREATS TO VALIDITY

In this study, the following potential threats to validity were identified:

Keywords bias: The set of keywords selected in the search may not be the most representative of the domain. Therefore, it may not have returned the best set of articles aimed at Sustainable SE. The mitigation of this bias, however, is difficult: Sustainable SE is a relatively new discipline and there are reports that the confusion of the lack of conceptual terms and understandings hampers the accuracy of the searches for searches that permeate the domain. We attempted to mitigate this threat by researching other mapping studies in Sustainable SE to identify which keywords were used.

Repositories bias: Because the researchers did not have the internal query engine mastery of each repository used, it was not possible to know how the query was computed by the databases. To mitigate this threat, the search string executed in each of the repositories has been adapted according to the constraints imposed by each database.

Quality evaluation bias: The evaluation of the quality and the classification of the articles was influenced by the knowledge and understanding of the authors. To mitigate this threat, the inclusion, exclusion and quality criteria of the selected studies were defined. In addition, the differences were discussed, resolved and agreed with the intervention of the advisor.

4.8 CHAPTER SUMMARY

The main motivation for this Chapter was to investigate the state-of-the-art on Sustainability in SE, through the systematic mapping of the literature, in order to synthesize the available evidence in the literature and to identify gaps and research opportunities.

The analysis of related studies that address the sustainable aspects of software construction was a difficult task, as these do not address the same objectives. However, the set of issues raised helps us to understand how researchers are doing the work in this area. Through this study, we were able to identify a set of previously unexplored variables, such as type of evidence, research methods, application domains and contribution focus. The research points identified throughout this study can be considered an important input for planning new research.

The results of the study highlight the growing interest of the SE research community in the Sustainable domain, including focus on mobile environments.

Regarding the contributions, we realized that, although there are a number of proposals for proposed approaches, frameworks and models, there is a need for more studies on techniques, tools and metrics. In addition, in general, there is a gap in evaluation studies. Regarding the life-cycle phases, the study reports that software requirements and design receive more attention, which leads us to realize that, because it is a new area, the concern is in understanding sustainability at a high level.

The results also point to the need for greater participation of industry in research in order to align their interests with those of academia for the benefit of all and reduce the impact of technology on environmental resources.

Next Chapter presents an industrial survey, performed in local organizations in order to gain a better understanding about the software industry awareness on Sustainable SE and the adoption of sustainable practices on software development process in the intention to provide new findings.

Chapter 5

SOFTWARE INDUSTRY AWARENESS ON GREEN AND SUSTAINABLE SOFTWARE ENGINEERING: A STATE-OF-THE-PRACTICE SURVEY

In the previous chapter, we performed a systematic mapping study on Sustainable SE. The goal was to map out the new SE context, synthesizing available evidence to suggest important implications for practice, as well as, identifying research trends and open issues. The study identified that 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 chapter, 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 research 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. This text was published in full in (KARITA et al., 2019).

The chapter is organized as follows. Section 5.1 discusses related work. Section 5.2 presents research questions defined. Section 5.3 provides the research design of our study. Results are given in Section 5.4. Section 5.5 discuss the results in the light of collected data, based on the set of analyzed dimensions. Section 5.6 presents the relevant implications that emerge from the analysis of this qualitative study. Section 5.7 presents threats to validity, and Section 5.8 finalizes our work by giving a conclusion and future works.

5.1 RELATED WORK

We have identified in recent literature few survey studies developed in the Sustainable SE field.

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 Sustainable 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. (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 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 (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.

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.

5.2 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 Sustainable 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 (LAGO et al., 2015).

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.

5.3 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.

5.3.1 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 (2005) and applied the research survey principles defined by Kitchenham and Pfleeger (2002). Figure 5.1 shows the methodological steps employed in this study.

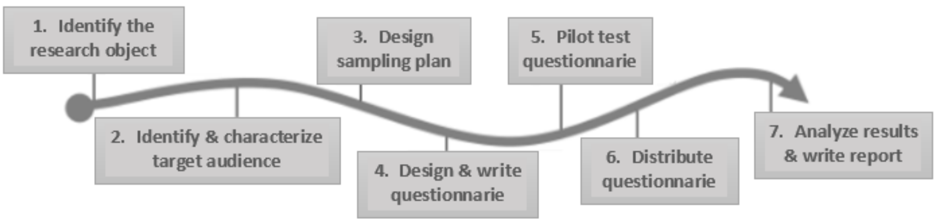

Figure 5.1 Survey design.

5.3.2 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.

5.3.3 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. The instrument is available in Appendix B.

- **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.

5.3.4 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?

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

5.3.6 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.

5.4 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.

5.4.1 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.

5.4.2 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.

5.4.3 Research Object

We next show the results of this empirical evaluation, based on the set of research questions previously stated in Section 5.2.

5.4.3.1 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 the author of this dissertation and a master student, we group them into the four dimensions of sustainability (technical, social, environmental and economic). The category system is illustrated in Figure 5.2.

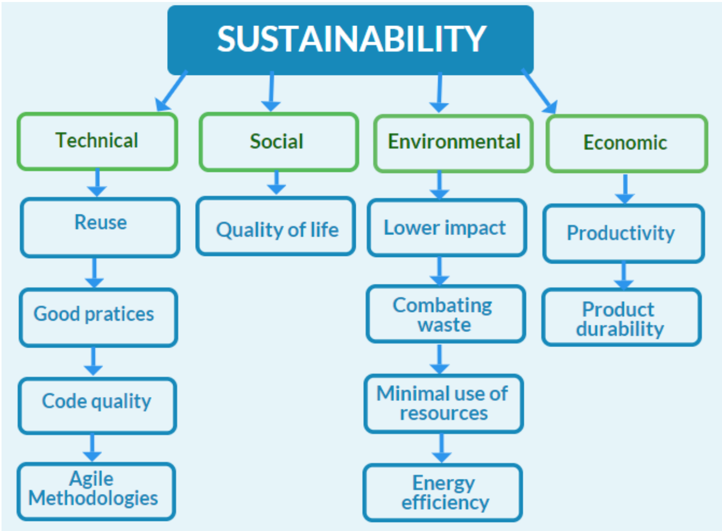

Figure 5.2 Coding RQ1.

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". (ERDELYI, 2013)

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". (NAUMANN et al., 2011)

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". (DICK et al., 2010)

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" (HILTY et al., 2006)

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." (DICK et al., 2010)

4% of the respondents identified with this definition.

"Environment friendly software that helps improve the environment". (GREEN, 2012)

4% of the respondents selected this option.

Next, we asked the respondents whether sustainability should or should not be considered as a Non-functional 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].

5.4.3.2 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 **??** 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
- Does not comply with legislation

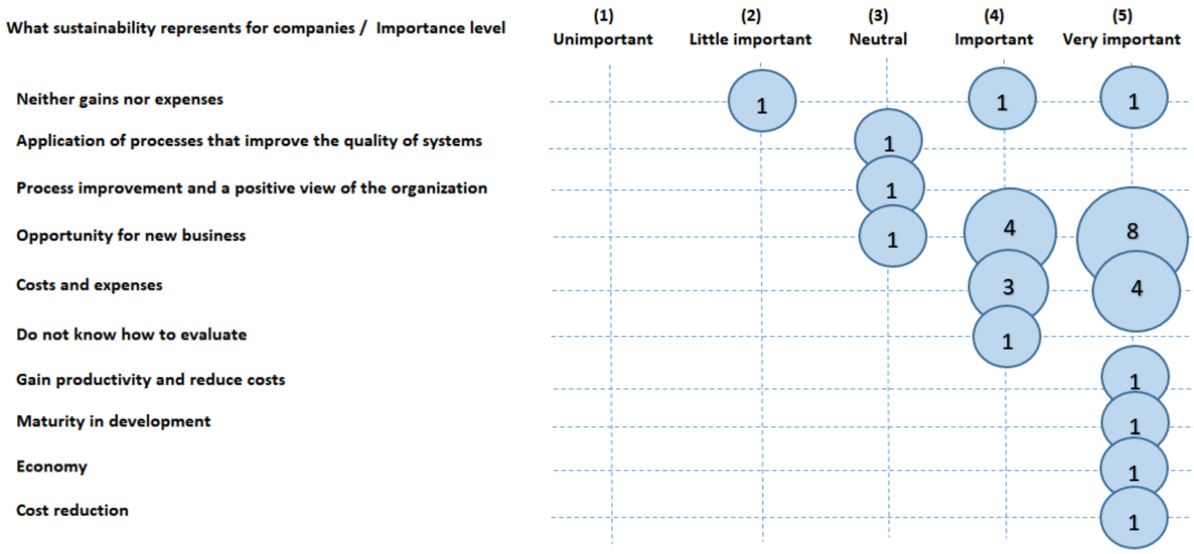

Figure 5.3 Company awareness level

As we can see from Figure 5.4, 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%

[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⟩

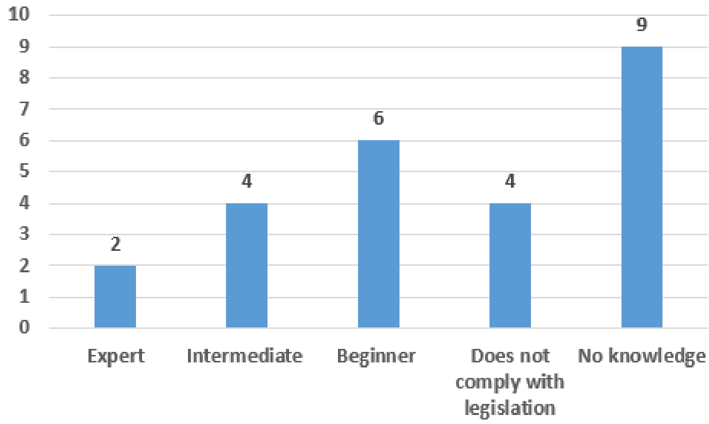

Figure 5.4 Company awareness level

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 5.5 shows. Because it is a multiple choice issue, the total ratio could exceed 100%.

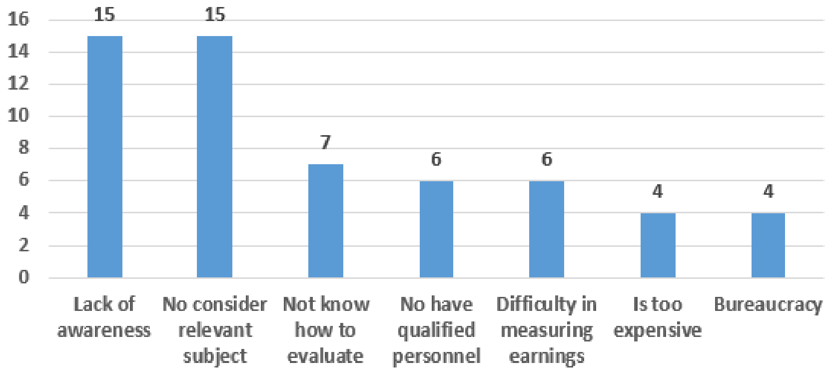

Figure 5.5 Main difficulties in adopting sustainable practices by companies.

5.4.3.3 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".

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 5.6 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.

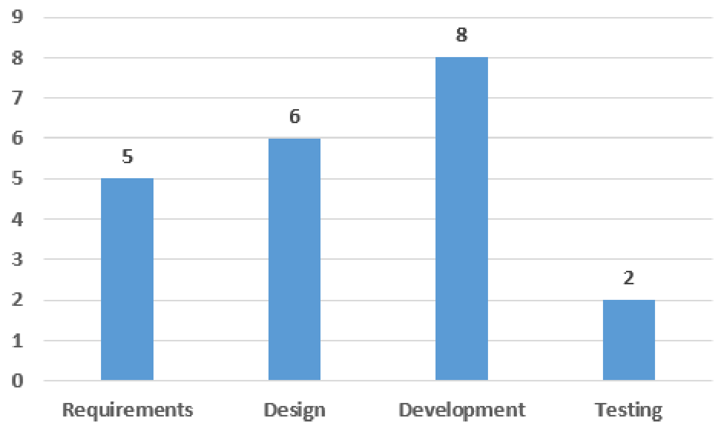

Figure 5.6 Phases of SDLC

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].

Table 5.1 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

5.4.3.4 RQ4: Sustainability dimensions

In this survey, we list the contributions proposed by Lago et al. (2015), 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 5.1 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%).

5.4.3.5 RQ5: Sustainability models

In this section it was investigated whether the professionals had an adequate knowledge about the Sustainable SE field, and if in their companies they applied any process model to support sustainability in SE 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.

5.4.3.6 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"*.

5.5 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 (LAGO et al., 2015).

  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.

[6] ⟨https://www.epeat.net/⟩

- ***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 (LAGO; PENZENSTADLER, 2017).

  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 SE. 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 (GROHER; 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. 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 SE 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.

5.6 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 Sustainable SE 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 sustainable software development.

5.7 THREATS TO VALIDITY

We discuss the following threats to validity (EASTERBROOK et al., 2008):

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 SE 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.

5.8 CHAPTER SUMMARY

This Chapter reported on a survey which analyzed the software industry's perception of sustainability in software development.

Although the Software Engineering community has increased its interest in such a field, the software industry has not explored it in an adequate fashion yet. Consequently, the sustainability practices are not completely known and applied by software practitioners.

The yielded results indicate an overall lack of knowledge about the topic, in particular regarding its underlying concepts, 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, there is a clear indication that the technical dimension is the most relevant and explored so far.

Next Chapter presents an initial theory to characterize the development of sustainable software and discuss, from a qualitative perspective, the main concerns extracted from both the literature and the software industry when they propose to build sustainable software.

Chapter 6

REACHING A COMMON UNDERSTANDING ON SUSTAINABLE SOFTWARE DEVELOPMENT: A GROUNDED THEORY

Based on the evidence obtained from the studies reported in previous Chapters, we observed that sustainable SE is still an incipient research area. Therefore, it is not possible to observe a consensus of the research community in SE on sustainable software development yet. Most recent studies tend to focus on isolated contributions at each SDLC phase. Consequently, software engineers still have a limited understanding about sustainable software development in practice. In particular, regarding the main concerns that permeate such a novel scenario.

In order to reach common understanding, we carried out a qualitative analysis which addressed the following research question: "How is it characterized the sustainable software development?". To answer it, we applied the Grounded Theory (GT) method (GLASER et al., 1968) whose emphasis is on the generation of new theories. We adopted the classic version (GLASER et al., 1968), as it relies on the method to answer a research question. We investigated a single of interest: the adoption of sustainable concerns in sustainable software development. In addition, research in the SE field currently demonstrates a predominance in the classical version.

This research was conducted using as data sources the primary studies retrieved in the systematic mapping presented in Chapter 4 and the main results from the survey presented in Chapter 5. The results may offer insightful propositions and explanations about how to produce sustainable software. In addition, it serves as a contribution for improving the state-of-art in the sustainable SE field.

The remainder of this Chapter is organized as follows: Section 6.1 presents the methodology. Section 6.2 presents the proposed characterization. Section 6.3 presents the theory evaluation. Section 6.4 discusses the main threats to the validity of this study. Finally, Section 6.5 summarizes this Chapter.

6.1 METHODOLOGY

To answer the research question: "How is it characterized the sustainable software development?", we employed a GT method to understand the sustainability concerns in the sustainable software development. To accomplish such goal, we analyzed recently published literature to understand the impacts these concerns on individuals, environment, society, economy and ICT.

Figure 6.1 shows the three-pillar-based GT method, according to the work of Glaser et al. (1968). The following sections explain each step in more detail.

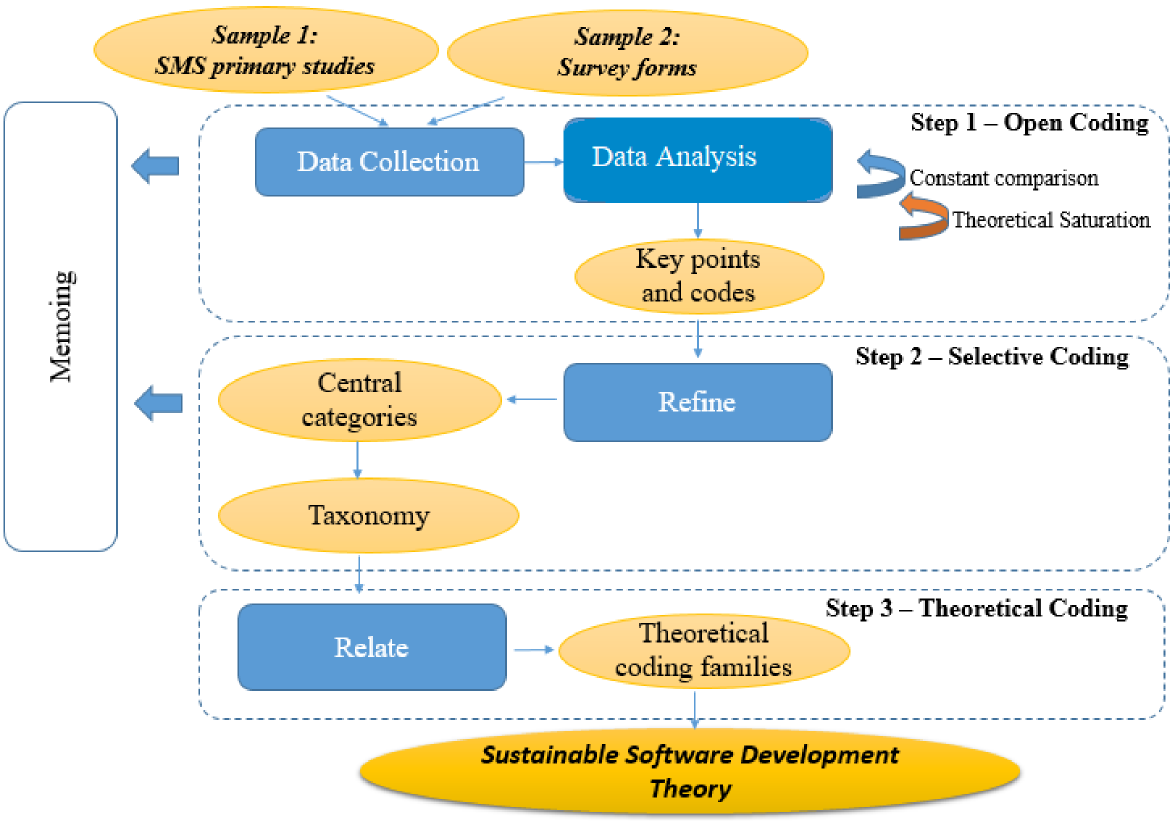

Figure 6.1 GT Process

6.1.1 Data Sources

This study integrates two data sources, that we called **Sample 1** and **Sample 2**. These are next explained.

- **Sample 1:** From the SMS (MOURÃO et al., 2018) described in Chapter 4, we observed the importance of acquiring the knowledge about the concerns that occur throughout the software development to achieve sustainability. We used 48 of the 75 primary studies retrieved and categorized in one of the following SDLC phases proposed in the SWEBOK:

- **Software requirements:** Papers that collected and specified requirements related to sustainability;
- **Software Design:** Papers that addressed the energy usage patterns of high and low level design decisions;
- **Construction:** Papers that discussed energy-efficient or energy-aware software and programming libraries;
- **Software Testing:** Papers that sought to locate and correct software power issues;
- **Software Maintenance:** Papers that discussed the role and impact of energy use during the software maintenance phase.

- **Sample 2:** A set of sustainability concerns were recovered from the results of the survey study (KARITA et al., 2019) presented in Chapter 5. This sample included 11 codes retrieved from the open coding carried out to the 25 responses to open questions applied in the survey. The codes retrieved helped us to identify sustainability concerns not found in the Sample 1.

6.1.2 Step 1 - Open Coding

The open coding involved a complete analysis of the Sample 1 to capture as many key points as possible. The objective was to identify relevant information into sentences to understand the basic characteristics and terminologies applied in the sustainability context.

The line-by-line data analysis was applied in the primary studies. According to the literature, this mechanism is more effective and useful than word-by-word data analysis, since these could become tedious and potentially error-prone (HODA; NOBLE, 2017). As suggested by Glaser (2002), the reading activity was performed with great attention, observation and curiosity to compare and capture the codes. In order to gain a deeper understanding of this sustainability concerns encountered, we carried out a second round of reading in the studies. Simultaneously with the identification of the codes, excerpts from the text considered relevant as sustainability aspects, concepts and terms related to sustainable SE and promising aspects of the theme that could be relevant to the process were extracted to an intermediary artefact, an spreadsheet [1].

Figures 6.2 shows the actions performed in aleatory records of the Sample 1 to obtain the codes. In the example, we have a primary study categorized as belonging to the requirements phase. Once the key points were identified from raw data, we assigned one or more codes to the key point. Key points are the summarized points from the text and can lead to several codes. A code is a word that summarizes the key point. In the end, we got the codes: "decision making", "investment incentives" and "sustainability stakeholders".

[1] Online at: ⟨http://bit.ly/2TiWa6W⟩

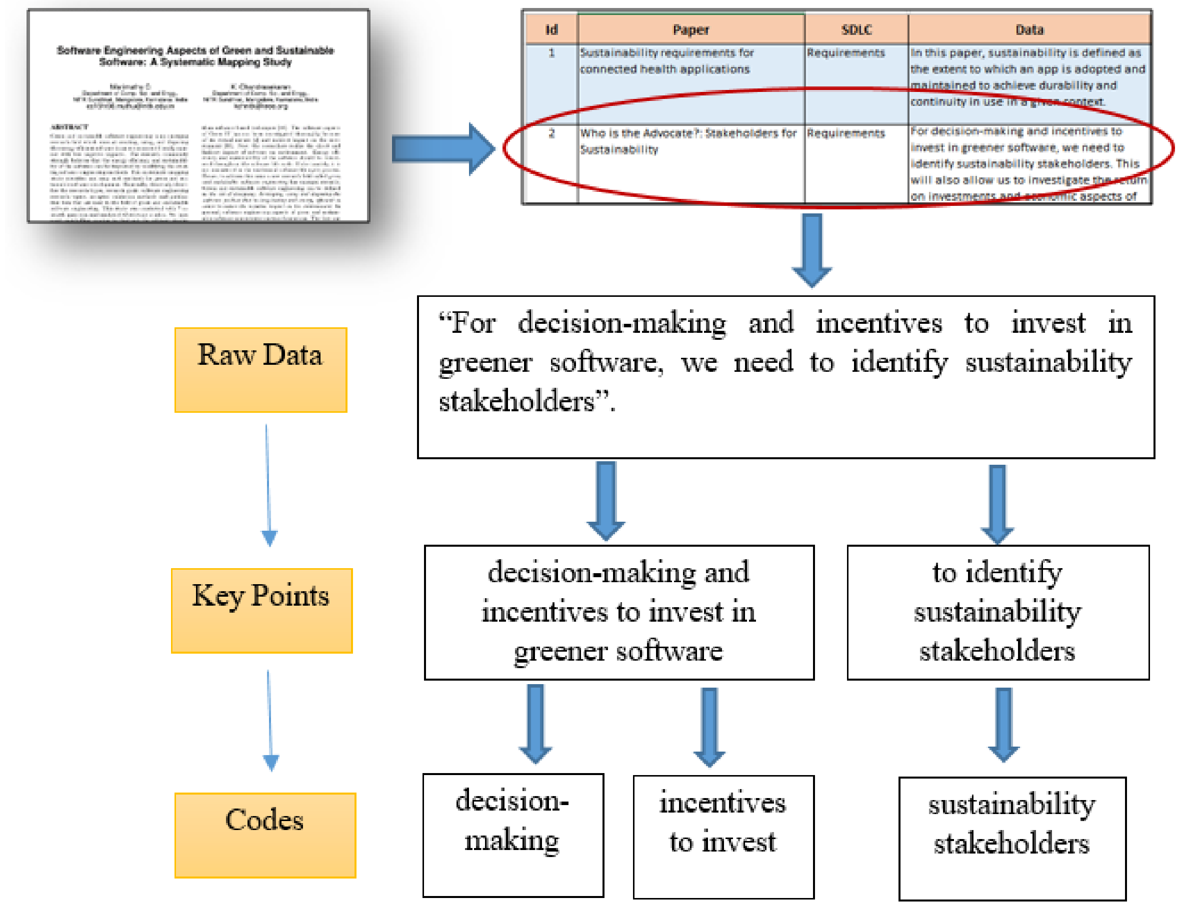

Figure 6.2 Codes identification

When the text under analysis generated the same codes as the code set previously analyzed, they were discarded, otherwise, a new code was inserted in the set. At the end of this step, the Sample 1 was analyzed and coded.

Next, we describe three activities of the GT method that were also carried out to support open coding step:

- **Writing Memoing:** This activity is a powerful way to allow all the ideas and thoughts about a certain code, concept, or category, to pour out. With further data collection and analysis, memos can be modified to reflect new ideas (HODA; NOBLE, 2017). We adopted the same spreadsheet as artifact to record the preliminary reflections and questions.

- **Constant Comparison:** In this activity the codes arising out of each study were constantly compared against the codes from the same study, and those from other studies (GLASER et al., 1968; GLASS; VESSEY, 1995; GLASER, 2002).

- **Theoretical saturation:** Open coding was over when theoretical saturation was

Table 6.1 Open Coding result

Category	Codes generated	Amount of Codes
Requirements	Business, Individual treatment, Comunity relationship, Continuity, Decision-making, Design for non-obsolescence, Durability, Ecological constraints, Economic aspects, Energy consumption, Energy efficiency , Enforce humane practices, Scalability, Sustainability stakeholders, Sustainability requirements, Financial costs, Good practice, Green efficiency, Incentives to invest, Local economy, Maintainability, Production sustainable environmentally, Quality requirement, Resource consumption, Carbon footprint, Resource efficiency, Return on investments, Social well-being, Sustainability, Sustainability culture and responsability, Sustainability policies and standards, Sustainable lifestyle, System evolution, Optimization, Time efficiency , Open source licensing.	36
Design	Awareness Best practices, Energy consumption, Life-cycle costs, Energy efficiency, Performance, Energy savings, Carbon footprints, Evolution, Improving architecture, Maintainability, Technical debt, Power consumption, Software optimization, Reusability, Software sustainability, Sustainable development, System quality.	18
Construction	Awareness, Energy efficiency, Energy consumption, Power behavior, Design decisions, Usability, Implementation choices, Carbon footprint, Decision-making, Power consumption, Energy savings.	11
Testing	Waste Battery power, Awareness, Energy savings, Energy hotspots/bugs, Energy consumption, Performance goals, Energy Efficient, Power consumption.	8
Maintenance	Carbon footprints, Awareness, Energy consumption, Energy efficiency, Legacy system modernizations, Decision-making, Refactoring, Energy reduction, Execution time, Social responsibility, Power consumption.	11

reached, that is, no new data added new insights to the process of analysis and categorization (GOULDING, 2002).

At the end of this phase, 84 exclusive codes were generated. Table 6.1 shows the result of the open coding step by considering each SDLC phase. A significant amount of codes was recurrent between the SDLC phases. Therefore, it is possible to infer that sustainability concern remains throughout the SDLC. For example, the codes “Energy Efficiency” and “Energy Consumption” were captured in all the SDLC phases. On the other hand, the code “ Stakeholders in sustainability ” was identified only in the requirements phase.

Figure 6.3 shows the coverage amount of these codes for each SDLC phase. The largest amount of codes were obtained in the requirements phase (36; 43%), followed by design (18; 21%), construction (11; 13%), maintenance (11; 13%) and testing (8; 10%), respectively.

Figures 6.4, 6.5, 6.6, 6.7 and 6.8 show the amount of codes identified in each SDLC phase, respectively.

6.1.3 Step 2 - Selective Coding

Selective coding refines the entire coding process carried out, identifying the central category of the theory to which all others are related to. Poorly formulated categories

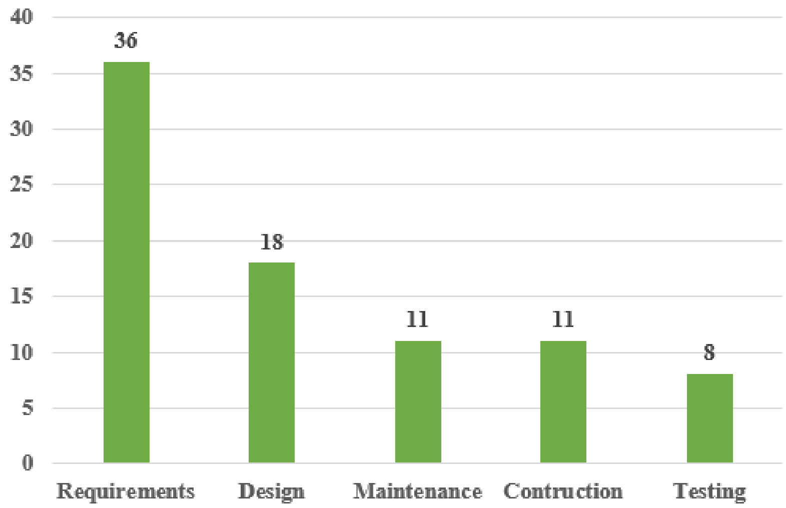

Figure 6.3 Amount of codes by SDLC Phase

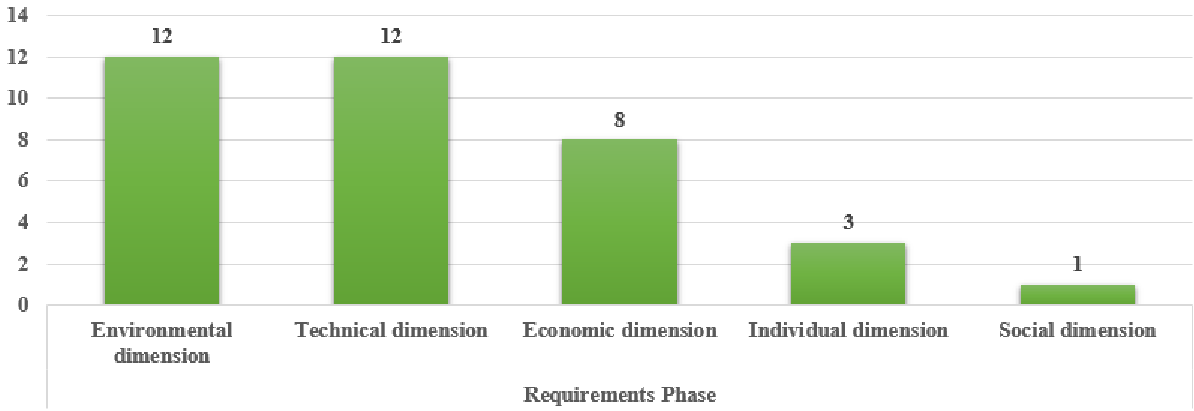

Figure 6.4 Amount of Sustainability Dimensions codes in the Requirements Phase

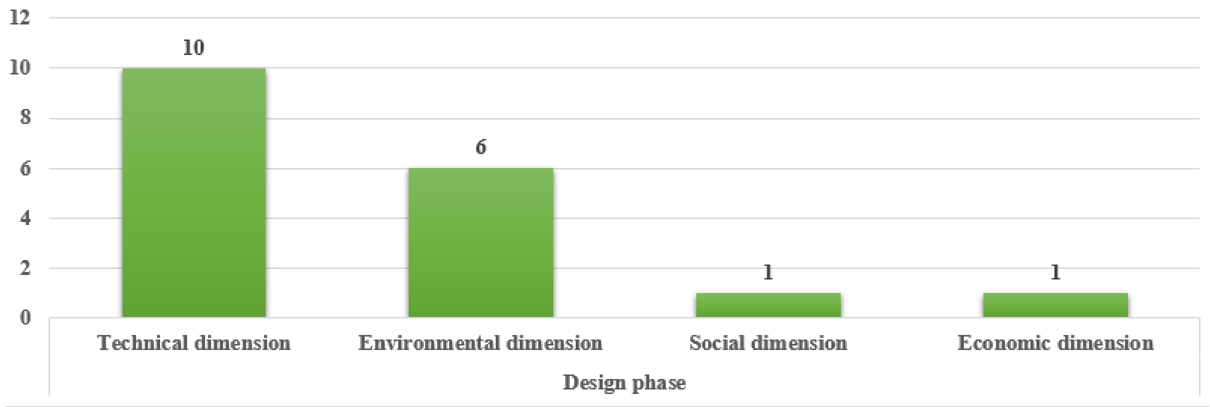

Figure 6.5 Amount of Sustainability Dimensions codes in the Design phase

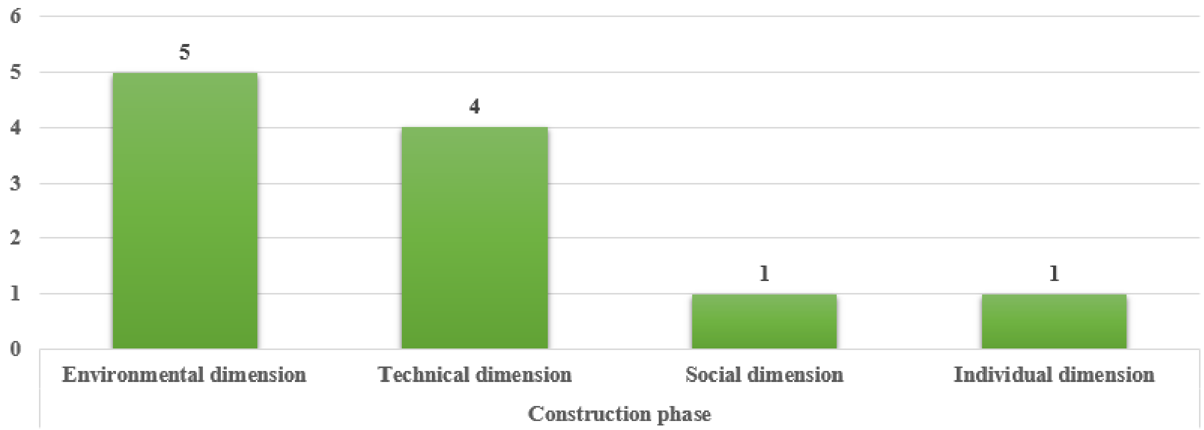

Figure 6.6 Amount of Sustainability Dimensions codes in the Construction phase

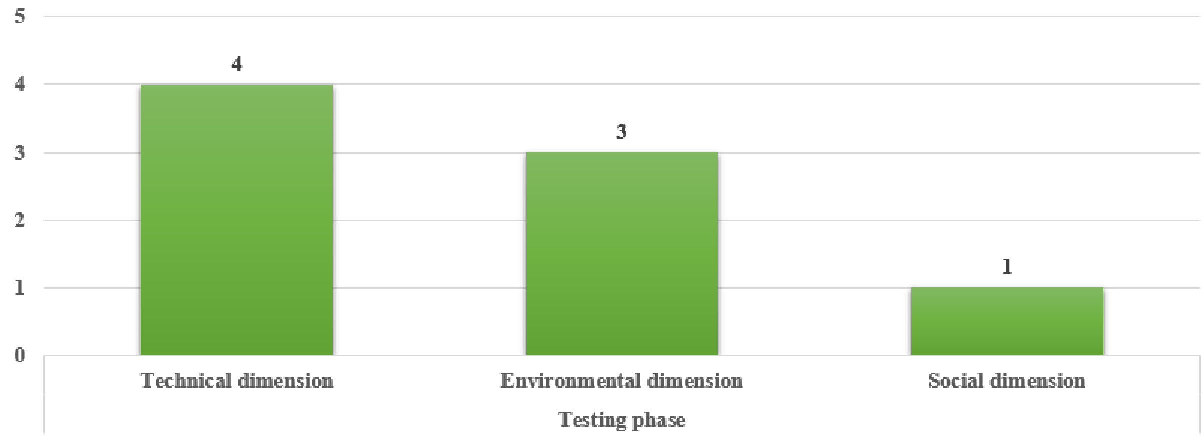

Figure 6.7 Amount of Sustainability Dimensions codes in the Testing phase

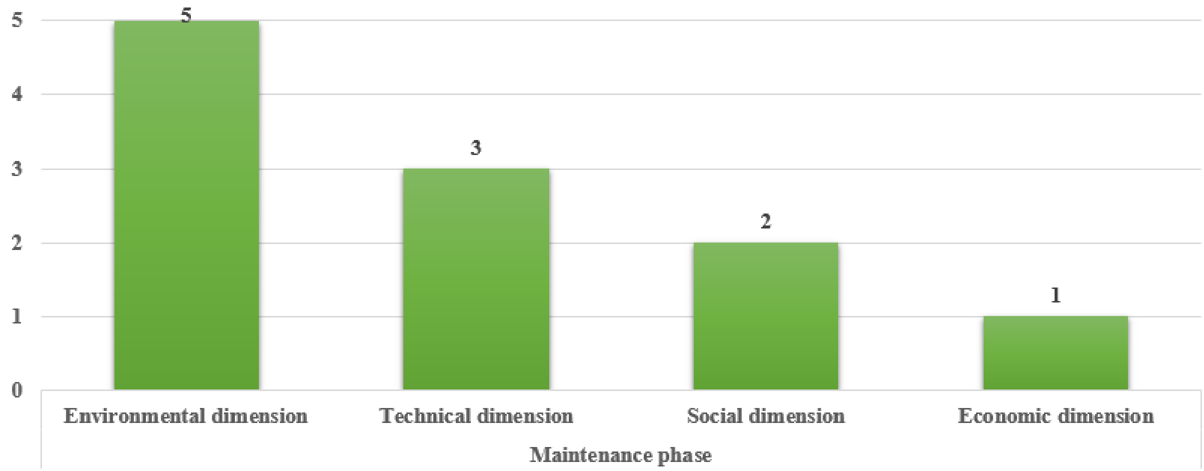

Figure 6.8 Amount of Sustainability Dimensions codes in the maintenance phase

must be revised and flaws in the logic of the theory resolved (GLASER et al., 1968). This step ends when "theoretical saturation" activity is reached, which occurs when the new data contains more evidence and examples than is already developed, but contains no new concepts or categories.

According to the literature, the central category can emerge from a list of existing categories or the researcher can study the categories and determine that, although each of them tells part of the story, none of them captures the entire story at all (GLASER, 2002). In this study, a central category was not sought because the conceptual idea was to explore codes in the light of the sustainability dimensions. Then, the codes found were grouped into four categories. Each of them represented a sustainability dimension: social, economic, environmental, technical and individual.

The selective coding step was based on the work of (LAGO; PENZENSTADLER, 2017). The authors analyzed a sample of six articles and based on the Software Sustainability Assessment Method (SoSA)[2] notation, they relate the software engineering contribution of each work with sustainability concerns. These concerns were addressed in two ways: directly (placed in the immediate impact area) and indirectly enabled (enabling impact area). The Lagos's study was considered an important source because it meets the execution proposal of this research, making a relationship between the sustainability concerns present in the studies with the sustainability dimensions. Figure 6.9 illustrates the result of the mapping carried out by the study that we used as a parameter to develop the initial theory.

The relationship between sustainable concerns and sustainability dimensions was also found in the studies of Raturi et al. (2014) and Penzenstadler et al. (2014a). Table 6.5 compares the two approaches. Both perspectives are similar and were used in this study as a complementary way.

After identifying the concerns mapped by the authors, and their relationship with sustainability dimensions, we performed the categorization of our research. We analyzed each code mapped in the open coding step and classified them in one of the sustainability dimension. Figures 6.11, 6.12, 6.13, 6.14 and 6.15 show the result of this classification. Each of them concerns a SDLC phase.

6.1.4 Step 3 - Theoretical Coding

The purpose of this step is to build a theory that explains how concepts, categories and relationships fit into a conceptual unit (GLASER et al., 1968). The theory proposed in this dissertation is an integrated set of concepts that explains the context of sustainable software development for the SE research community and software practitioners.

According to Glaser (2002), categories can represent families of common structures, also known as "Theoretical coding families". This grouping is done taking into account six Cs (Context, Cause, Consequence, Contingency, Conditions, and Covariance). For this research, the theoretical coding family adopted was the "context family", as it allows us to represent the findings in the sustainability dimensions context.

Figure 6.15 shows the theory representation. We used the notation proposed by

[2]Online at: ⟨https://www.slideshare.net/patricia_lago/sosa-a-software-sustainability-assessment-method⟩

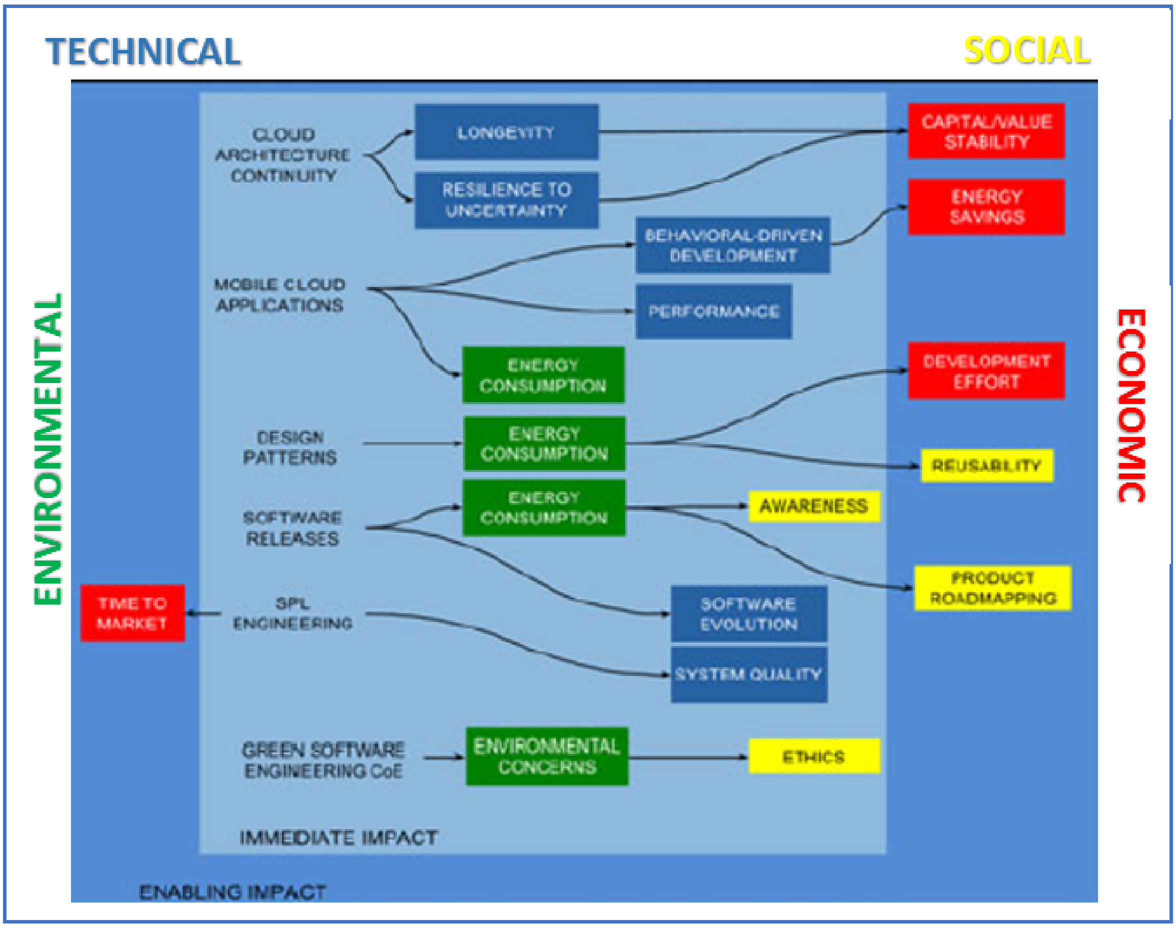

Figure 6.9 Lago's approach (LAGO; PENZENSTADLER, 2017)

Sjøberg et al. (2008) to SE studies, their components (actor, technology, activity and software system) and their relationships (propositions represented by the "P" letter followed by an incremental number). The typical situation is: an ***actor*** applies ***technologies*** to perform certain ***activities*** on an (exiting or planned) ***software system*** (SJØBERG et al., 2008). We next explain these components.

- **Actor:** The actors are the roles someone would play in the software development process to provide sustainability;
- **Technology:** This is a central component. It presents the emergent codes of the GT. Each code is presented with a color that represents one of the categories or sustainability dimensions: individual, social, economic, environmental and technical;
- **Activity:** The activities identified in the GT are the SDLC phases;
- **Software System:** It could encompass either a new development or maintenance projects.

Table 6.2 Comparison between the two approaches.

Dimension	**Requirements by Raturi et al. (2014)**	**Requirements by Penzenstadler et al. (2014a)**
Environmental	Requirements with regard to resource flow, including waste management, can be elicited and analyzed by Life Cycle Analysis. Furthermore, impact effects can be analyzed by environmental impact assessment (EIA). The challenge is that usually only first order impacts by a system are considered, whereas second and third order impacts are not yet accounted for.	Seeks to improve human welfare by protecting natural resources, such as water, land, air, minerals, and ecosystem. Any system applied in a real-world context is situated within a natural environment which means that it has an impact on the environment. Environmental sustainability can be managed by controlling resource flow: waste management, life cycle analysis, and environment impact assessment.
Individual	Are covered by privacy, safety, security, HCI and usability as well as personal health and well-being, which still needs to be made explicit in requirements. An example for this could be that an application suggests to take a break after a specific amount of working time.	Refers to the maintenance of the individual human capital, eg, health, education, skills, and access to services. Individual sustainability can be covered by privacy, safety, security, human-computer interaction, usability, personal health, and well-being.
Social	Can be treated via computer supported collaborative work (CSCW) requirements, which reflect the interaction within user groups, via ICT for development requirements, and via political, organizational, or constitutional requirements, as in laws, policies, etc. Still missing are, for example, explicit requirements for strengthening community building.	Aims to preserve the social capital and preserve services and solidarity of social communities. Social sustainability can be handled via computer-supported collaborative work that aims to strengthening community building and improve community interaction.
Economic	Is taken care of in terms of budget constraints and costs as well as market requirements and long-term business objectives that get translated or broken down into requirements for the system under consideration. The economic concern lies at the core of most industrial undertakings.	Aims to maintain capital assets and added-value (interest) assets. Economic sustainability can be taken care of in terms of costs, budget constraints, long-term business objectives, and market requirements among other economic requirements.
Technical	Include non-obsolescence requirements as well as the traditional quality characteristics of maintainability, supportability, reliability, and portability, which all lead to the longevity of a system. Furthermore efficiency, especially energy-efficiency and (hardware-) sufficiency.	Refers to software systems longevity and their adequate evolution with changing surrounding conditions and respective requirements. Technical sustainability requirements include all requirements which lead to the longevity of a system such as non-obsolescence requirements and the ISO/IEC 2501029 quality characteristics (eg, maintainability, reliability, and transferability). Moreover, energy efficiency is also part of technical sustainability requirements.

We analyzed emerging relationships between the aforementioned items, i.e., the propositions. According to the literature, a proposition represents the specific values that an unit have in relation to the value of another (SJØBERG et al., 2008). Table 6.3 lists the propositions of the theory. They would be regarded as initial propositions. We discussed the theory findings in the next section.

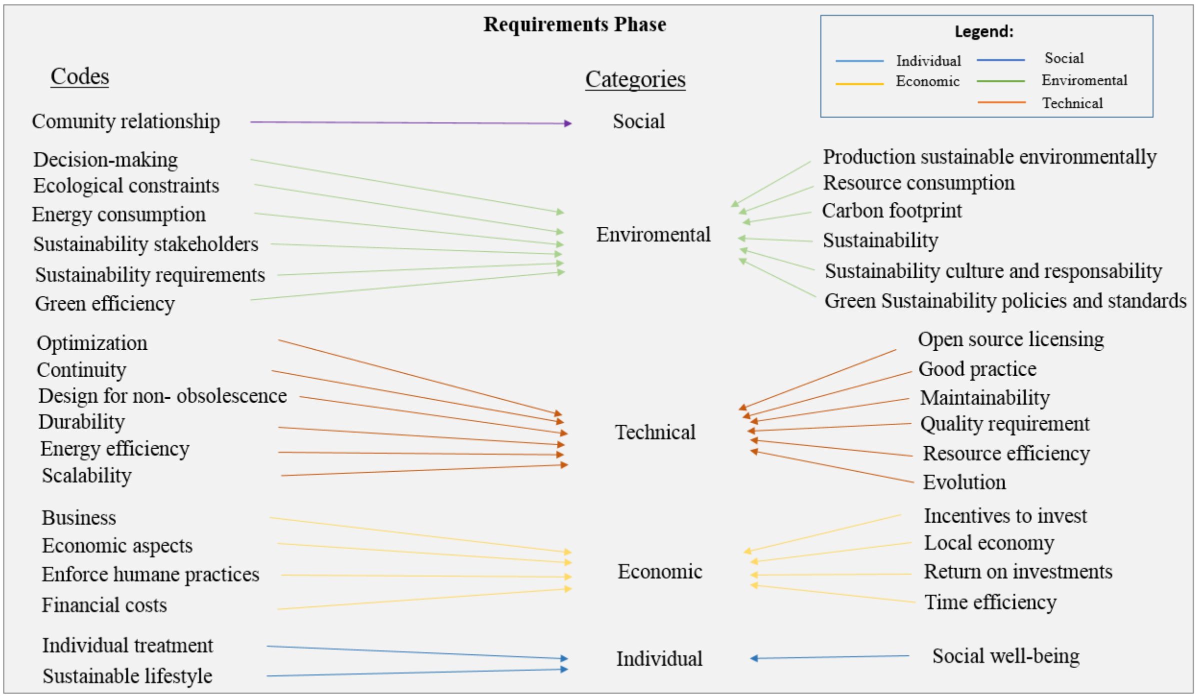

Figure 6.10 Code categorization in the requirements phase

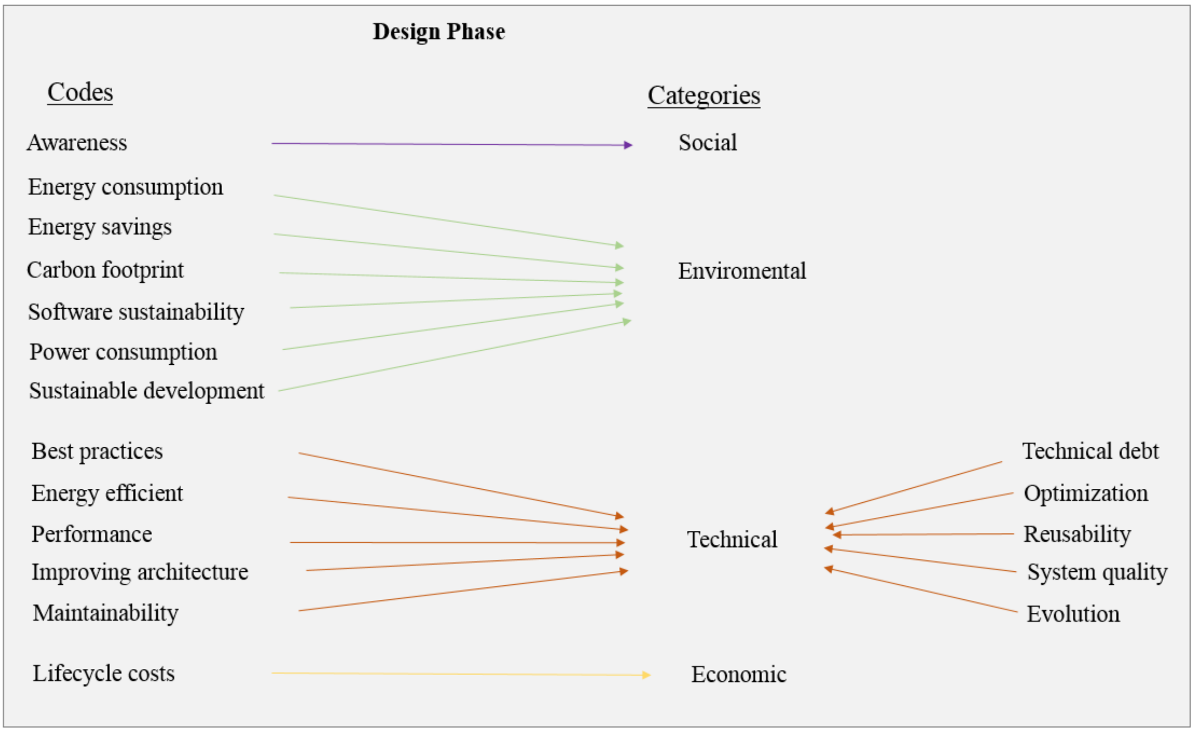

Figure 6.11 Code categorization in the design phase

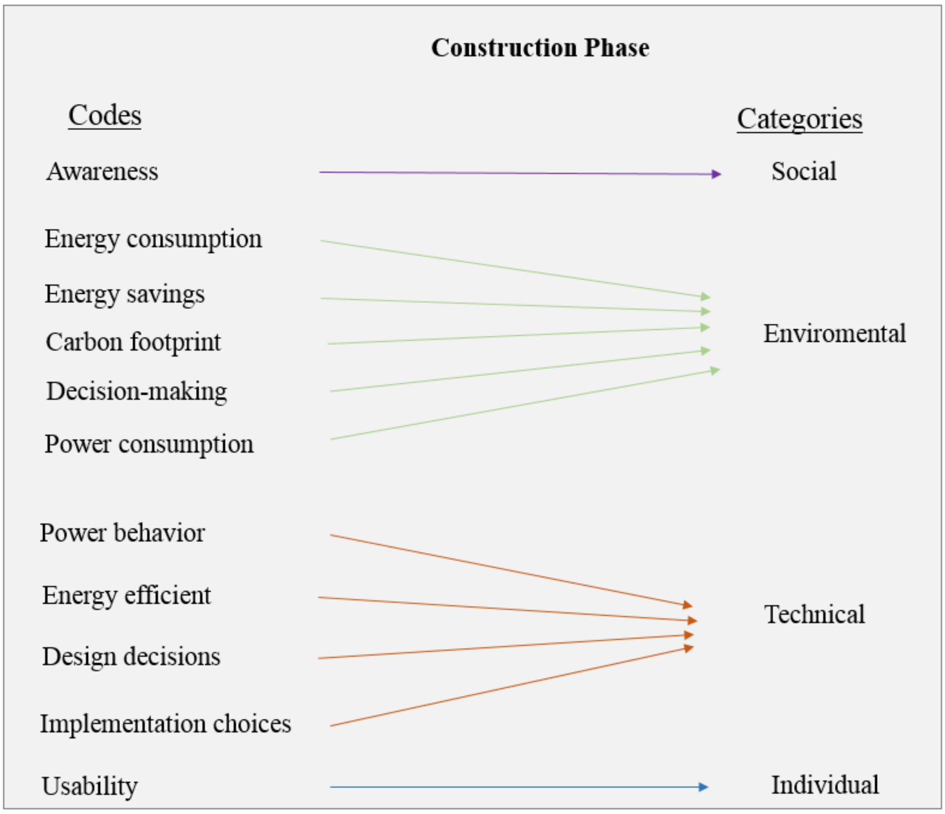

Figure 6.12 Code categorization in the construction phase

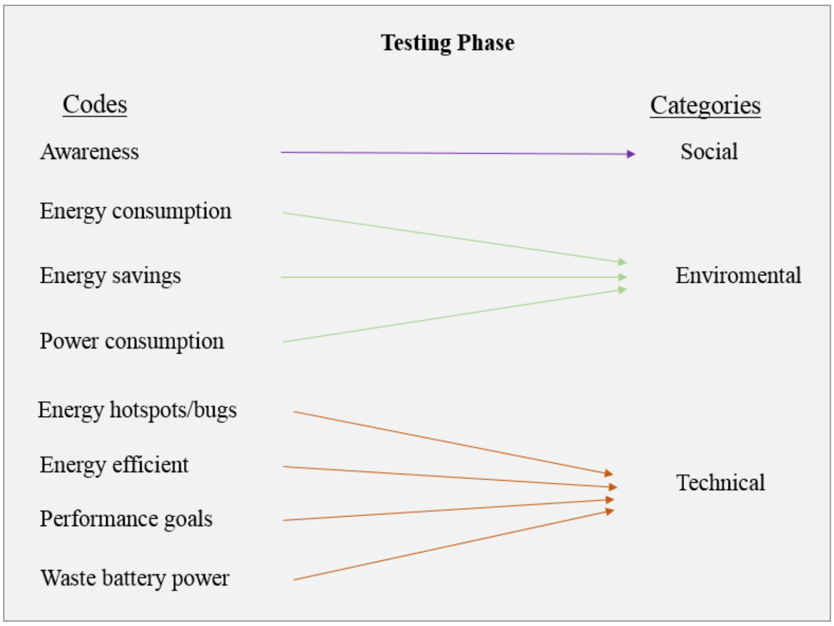

Figure 6.13 Code categorization in the testing phase

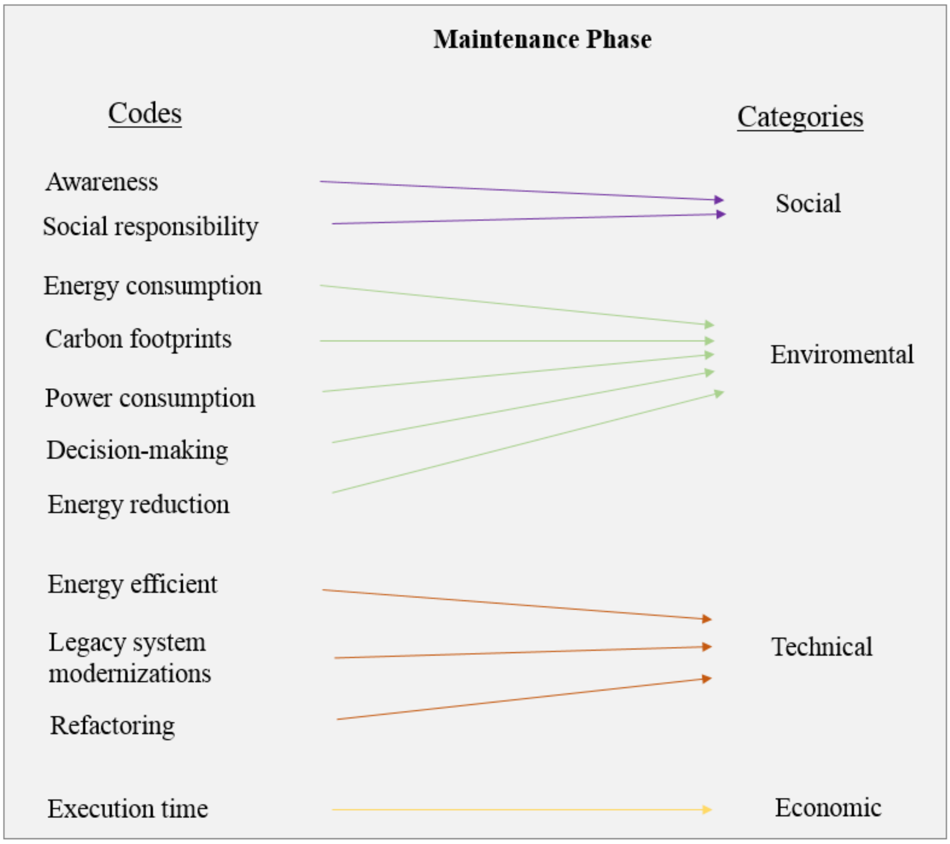

Figure 6.14 Code categorization in the maintenance phase

Table 6.3 Propositions of the theory

Id	Description
P1	Requirements engineers need to identify the sustainability stakeholders.
P2	Some requirements can be associated to more than one sustainability dimension.
P3	Sustainability stakeholders allow to investigate the return on investments and economic aspects of green software development.
P4	Requirements engineers need to analyse the sustainability of the context, eliciting sustainability objectives, goals and constraints.
P5	Requirements engineers could explicitly include a section on sustainability in the software requirements specification document template.
P6	Sustainability concerns can significantly impact the long-term effects of the systems design.
P7	Sustainable software development should trade off between environmental sustainability criteria and traditional quality requirements.

Continued on next column

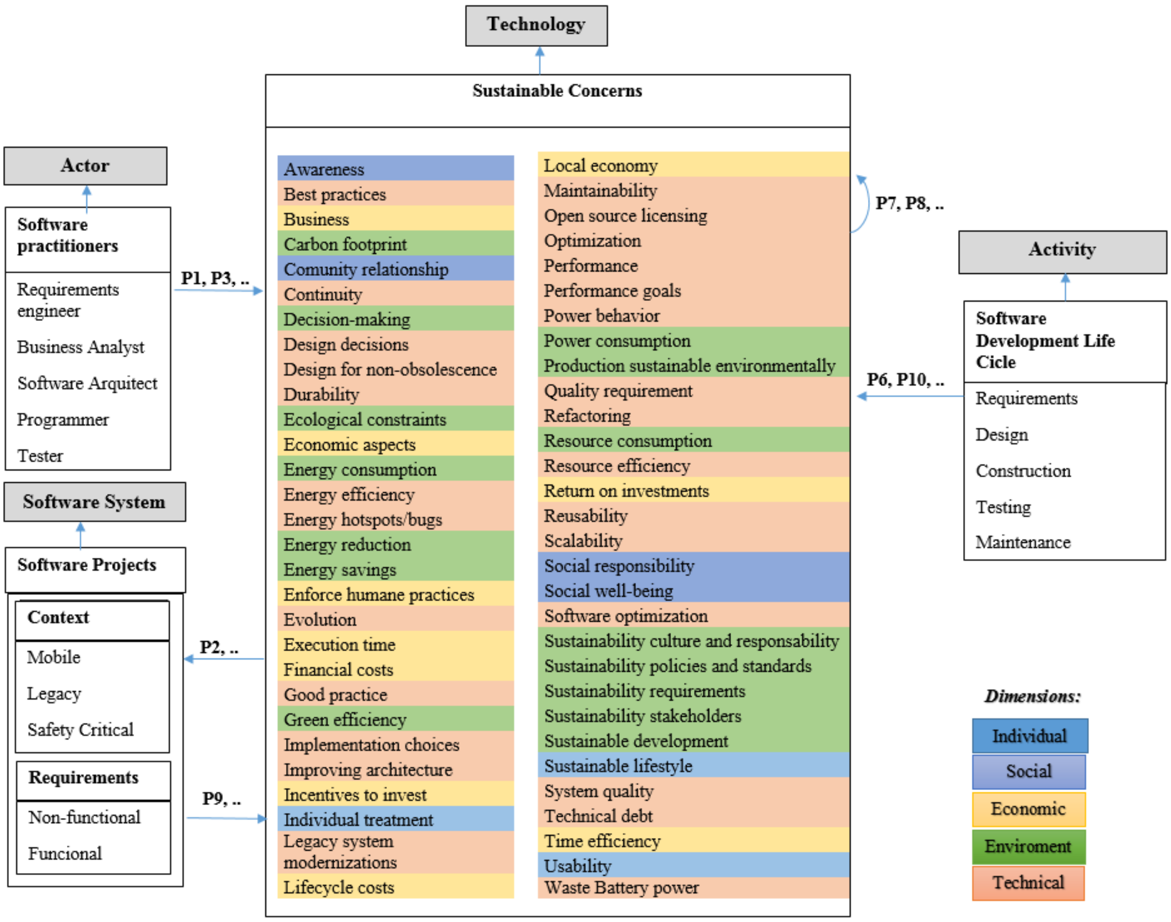

Figure 6.15 Theory representation

Table 6.3 – Continued from previous column

Id	Description
P8	The software development industry needs making certain trade-offs between the demands of end-users and the requirements for sustainability.
P9	There is no consensus in the literature on how to relate sustainability in terms of software quality.
P10	Sustainable requirements elicited will must be designed into the system.
P11	Requirements elicitation can be accomplished through interviews, observation, participatory workshops, as well as through goal elaboration, etc.
P12	Sustainability often is seen as a trade-off between the present and the future.
P13	Some aspects of the sustainability dimensions overlap with other non-functional requirements.

Continued on next column

Table 6.3 – Continued from previous column

Id	Description
P14	Energy requirements allows optimising an application's energy footprint.
P15	Sustainable requirements allows increases the awareness for major energy consumers.
P16	The design of software is critically important for sustainability.
P17	Software engineers are responsibly by the creation of sustainable software.
P18	Requirements Engineering is a key area where systems level thinking can be applied to identify sustainability concerns.
P19	Green tactics help architects extend their design reasoning towards energy efficiency.
P20	Energy-efficiency is defined as an important quality attribute for mobile and pervasive applications.
P21	Different designs can have a significant impact on the energy efficiency of software applications.
P22	Software development environments used efficiently can consume less energy.
P23	Software with more algorithmically efficient is also more energy efficient.
P24	Energy efficiency and performance represent separate and possibly conflicting issues.
P25	A better understanding of high-level design can play an important role in reducing the power consumption.
P26	A better understanding of implementation choices can play an important role in reducing the power consumption.
P27	Conscious software developers can influence design and implementation decisions on the final product's energy consumption.
P28	Energy consumption is critical to provide sustained service within mobile and wireless environments.
P29	Applying best practices can significant improve software energy efficiency.
P30	Software architects and developers need to think about energy efficiency.
P31	A solid knowledge base is needed to provide guidance in building energy-efficient software.
P32	Energy-efficiency is a key concern in continuously-running mobile applications.
P33	Green design can help minimize the system energy consumption.
P34	Energy usage can directly affect the usability of a mobile device.
P35	The number of fixed defects is negatively related with the power consumption.
P36	Many defects reflect a poor design and cause more power to consume.
P37	Reducing energy consumption should be raised to first-class status among performance goals when software is being designed.

Continued on next column

Table 6.3 – Continued from previous column

Id	Description
P38	The awareness of energy consumption can later enable the testers to generate energy-efficient techniques.
Legend: P#: Id	

6.2 UNDERSTANDING THE SUSTAINABLE SOFTWARE DEVELOPMENT

The understanding about the sustainable software development is explained through a network of propositions that relate the exploited sustainability concerns. We analyzed the propositions and identified common characteristics. Then, we obtained five main groups or explanations as suggested by Sjøberg et al. (2008). They are:

E1 Technical, environmental and social concerns are present at all the sustainable software development phases;

E2 Sustainability requirements should be considered in the early software development phase;

E3 Need for stakeholder engagement focused on sustainability;

E4 Software quality requirements help to develop sustainable software;

E5 Some sustainability concerns generate trade-offs.

Next, we explain each of them.

E1: Technical, environmental and social concerns are present at all the sustainable software development phases

The theory has shown that although the literature states that sustainability in the context of software is related to technical issues, it is clear that researchers are concerned to consider simultaneously the environmental resources, social welfare, individual, economic prosperity and longevity of the built systems. According to Lago et al. (2015), supporting sustainability from the business strategy, through the requirements, design and implementation of the software can provide a potentially positive impact both on the environment and on people's lifestyles.

From the perspective of Sustainable Development, there must really be a balance between the dimensions of sustainability, since sustainability is essentially a multidisciplinary subject. In this sense, the integrated vision of a series of disciplines becomes crucial for the development of a sustainable software project.

Technical and environmental concerns were expected to be identified in all explored studies.

One of the observed categories that caught our attention was "Awareness". This concern is present in all stages of the sustainable software development process, which leads us to realize that, more important than applying techniques and tools, is the team being aware of its use. Awareness permeates the need to understand the context of sustainable development. It is important to know the application field and, especially, to understand the needs of the stakeholders of the produced software. Thus, the environmental needs that would promote the development of sustainable product would be absorbed.

Sahin et al. (2012) states that a better understanding of implementation choices can play an important role in reducing the power consumption. To Munoz et al. (2017) software developers aware can influence the design and implementation decisions on the energy consumption of the final product. According to Lago and Jansen (2010), software architects and developers need to think about energy efficiency.

To achieve sustainability awareness, it is necessary to achieve highly interdisciplinary collaboration. Therefore, software engineers must be trained and equipped with technology and tools that enable them to design such software with individual, social, economic, technical and environmental sustainability dimensions in mind.

E2: Sustainability requirements should be considered in the early software development phase

Becker (2014) states that the impact a software system will have on its environment is often determined by how the software engineers understand its requirements.

In the context of traditional development, it is well known that this phase is historically responsible for the major problems caused during software development and the failure of many projects.

Although incipient, the research community studying sustainability in SE has frequently addressed this issue in order to analyze what and how Requirements Engineering (RE) could contribute to improving environmental sustainability. Becker et al. (2015) states that RE is a key area where systems level thinking can be applied to identify sustainability concerns, as it translates the domain-dependent goals and concerns into technical requirements that can be realized in the implementation of a software system. Recognized as a key topic, sustainability could be integrated into the requirements that define what the software does.

RE for Sustainability (RE4S) proposes to describe the RE process while taking steps to make the system more sustainable. Requirements engineers can specify sustainable requirements through interviews, observation, participatory workshops, etc. The important thing is to treat sustainability explicitly as a relevant topic. This can be facilitated by identifying the sustainability requirements in the inappropriate use of the software in relation to sustainable practices or by following general good practices (CHITCHYAN et al., 2015).

The application of the theory showed that the largest number of sustainable concerns were identified in the requirements phase. Additionally, this is the only phase that contains requirements related to all sustainable dimensions. One possible argument is that requirements analysis is the basis of a software project. Through it, the software engineer could identify, quantify and prioritize functional and quality software requirements.

In order to meet the objectives of sustainable development, the RE process should encompass activities that allow analyzing the sustainability context in the software project in question, identifying sustainability stakeholders and achieving sustainability goals and objectives. Penzenstadler et al. (2015) state that early-stage sustainability goal modeling gives us a clear picture of the potential of the system if it is implemented and used as intended.

To Penzenstadler et al. (2014a), RE could explicitly include a section on sustainability in the software requirements specification document template.

Another interesting aspect observed was the capture of requirements related to social, individual and economic dimensions in the requirements phase. The following economic concerns were mentioned: ***investment incentive, local economy, return on investment*** and ***time efficiency***. Similarly, individual and social concerns: ***Sustainable lifestyle, community relationship*** and ***social welfare*** were considered relevant to be explored in the context of the corresponding application domain. In the sustainable dimension scenario, the following stand out: ***Sustainability culture and responsibility*** and ***Sustainability policies and standards, Sustainability stakeholders, Ecological constraint***s and ***Decision-making***. In the technical dimension we found some quality requirements: ***Continuity, Durability, Energy efficiency, Scalability, Maintainability, Good practice, System evolution, Optimization and Open source licensing.***

We could observe the existence of three groups to allocate the concerns present at this stage. A further effort to be explored would be to classify these sustainable concerns into functional requirements, software quality requirements or software development assumptions / constraints.

E3: Need for stakeholder engagement focused on sustainability

In addition to the proposition P2, the theory exposes the "sustainability stakeholders" concern, which brings up a discussion about the importance of identifying stakeholders who are experts in this field, since it is the responsibility of those involved in software creation carefully consider sustainable impact" (BECKER et al., 2015). For Becker et al. (2015), software engineers are responsible for the long-term impacts of the systems we design.

The role of this "sustainability expert" is to document how specific services and products can affect the different dimensions of sustainability.

To integrate the right sustainability requirements into a given system, software engineers need to identify the appropriate sources for those requirements, i.e., their stakeholders (CHITCHYAN et al., 2015). Penzenstadler et al. (2013) state that the identification of these sustainability stakeholders by software engineers would allow us to investigate the return on investment and economic aspects of sustainable software development. Penzenstadler et al. (2015) complement that this sustainability profile may also introduce new challenges to the system to either maintain or improve sustainability.

Already Huber et al. (2015) bring the definition of "indirect stakeholders", i.e., the people who are affected by the use of the implemented system. For the authors, these stakeholders should be involved in extracting sustainability requirements.

To meet these concerns, the requirement engineers should include an activity to identify the users affected by each of the identified requirements. Consequently, they could be aware of which dimensions would be affected during the production of the software. According to Huber et al. (2015), the typical roles that play a role in social sustainability are managers, economic sustainability, responsible budgeting, the technical dimension, managers and environmental sustainability, there is also the role of designated Corporate Social Responsibility (CSR).

Therefore, what we can see from the theory is that, to create sustainable software, software companies need to promote the experience of their software engineers, apply principles of sustainability in the RE discipline and, mainly, apply continuous improvement to constantly evolve in the sustainability field.

E4: Software quality requirements help to develop sustainable software

One of the key challenges presented by the sustainable SE research community is whether sustainability should be treated as a software product quality attribute. Although current discussions state that sustainability should be considered in software development process, actual models and quality standards, such as ISO/IEC 9126 (International Standards Organisation (ISO), 1991) and ISO/IEC 25010 (International Standards Organisation (ISO), 2011), do not consider sustainability as a quality attribute. However, there are studies (CALERO; BERTOA, 2013; VENTERS et al., 2014b; BECKER, 2014; PENZENSTADLER et al., 2014a) discussing the relationship between software quality and sustainability.

In software development, product quality is directly related to the quality of the development process, so it becomes obvious that the search for sustainable software necessarily requires an adaptation in the software development process to incorporate sustainability concerns. Therefore, in order to characterize sustainable software development we need to identify the quality aspects for the project that is in focus. Prior to projecting this quality, software engineers must map the characteristics that will meet the quality and the terms that will describe those characteristics.

What the theory allowed to observe was that some studies consider sustainability as a new factor affecting software quality by stating that sustainable software requirements need to be considered beyond traditional quality requirements in software development (CALIENES, 2013). On the other hand, Huber et al. (2015) argue that treating sustainability requirements as a subcategory of specific quality requirements turns out to be inadequate. Indeed, discussions on this subject are still incipient and it is not possible to identify a consensus in the literature on how to relate sustainability in terms of software quality yet (KOÇAK et al., 2015).

The quality attributes explored are present at all stages of the sustainable software development process and have been mapped as technical dimension requirements. They are: Continuity, Durability, Energy efficiency, Scalability, Good practice, Maintainability, Resource efficiency, Evolution, Optimization, Performance and Reusability. Therefore, we can confirm that identifying quality attributes is not something that should be thought of at the end of the project, but rather at the beginning instead. This activity would help the software engineer treat each of them with the appropriate tools in the corresponding phase. The fact that it is detected early in development brings benefits to the project because the later an attribute is identified, the more expensive it can be to implement.

For many researchers, the key factors that determine sustainable aspects of software as a product are related to energy consumption. For them, this interest is the main resource used by the hardware to run the software. Therefore, when the concern is to represent the reduction of energy consumption, Beghoura et al. (2017) proposes to incorporate energy efficiency as an additional feature of the software quality model. In the mobile application domain, it is precisely the "energy efficiency" interest that is increasingly being defined as an important quality attribute and is seen as a key concern (PROCACCIANTI et al., 2014; NIKZAD et al., 2014).

E5: Some sustainable concerns generate trade-offs

García-Mireles et al. (2017) state that the software impacts can be studied in three distinct scope levels: direct (e.g., energy consumption), indirect (e.g., reducing energy consumption when supporting a business process), or rebound effect (e.g., optimizing energy efficiency of a product could have the effect of increasing its demand and therefore the overall energy consumption due to such product). The third level can also be interpreted as interactions, conflicts or constraints.

The theory shows that some quality features showed conflicting interactions with sustainability aspects:

- **Energy efficiency X Performance:** They represent separate and possibly conflicting issues. While a substantial body of literature has focused on time performance, to the best of our knowledge, there are no previous studies relating performance to application software's energy efficiency (CAPRA et al., 2012);
- **Legacy system modernization X Energy consumption:** Considering environmental sustainability legacy system modernization regarding increasing the system functionality has a mixed effect on energy consumption (KOÇAK et al., 2015). The authors related the use of software modernization efforts and software energy consumption suggesting that there may be a trade-off between legacy system modernizations and energy consumption. This is due to the fact that increased system functionality has a mixed effect on power consumption;
- **Refactoring X Power consumption:** Although research indicates that the refactoring field is now sufficiently mature to improve system maintainability,

most refactoring techniques decrease sustainability. In particular, the excessive message traffic derived from refactoring god classes increases a system's power consumption (PÉREZ-CASTILLO; PIATTINI, 2014);

- **Software defects X Power consumption:** The investigation yields that the number of fixed software defects is negatively related with the power consumption. Defect removal is not just a coding error that is fixed, sometimes it is a logical error which have an impact on the design of the software. If software has many defects, it has poor design and thus causing more power to consume (LI et al., 2013; AHMED et al., 2014).

To minimize these conflicts it is important to study the contextual factors that may impact interactions between two conflicting requirements.

6.3 THEORY EVALUATION

Corbin and Strauss (1990) suggest that researchers using GT procedures should discuss their procedural operations, even if briefly. For the authors, if a GT researcher provides the pertinent information, they enable readers to assess the adequacy of a complex coding procedure.

The evaluation of the initial theory proposed in this study is based on the four factors defined by Strauss and Corbin (1990) to evaluate a GT: Fit, Understanding, Generality, and Control. There are discussed next.

6.3.0.1 Fit. The theory must have the substantive area and match the data. When researchers develop a theory of the phenomena studied, there is a risk that the theory incorporates the ideals and perceptions of the researcher. Coleman and O'Connor (2007) states that "When these theories subsequently do not quite reach the developed categories, the consequences are often to force data to do so and to reject data that is not or cannot be forced to do so". Therefore, it becomes a basic premise that GT has as its starting point a diverse sample of collected data. The theory we have developed fully addresses this factor, as it presents in Sub-section 6.1.1 which data were collected in the sample.

6.3.0.2 Understanding. Theory makes sense for practitioners. Although there is an expectation that researchers will have a good grounding on the researched domain, their expertise was used during the process of exploration and collection of sustainable interests and also to achieve theoretical sensitivity. This knowledge was also used to drive study completion. To validity to this factor, the research applied the constant comparison method as predicted by the GT methodology. This step was important because the constant reevaluation of the texts ensured that the researcher's bias was minimized and the theory maintained.

Coleman and O'Connor (2007) states that "A theory that closely represents the realities of an area will make sense and will be understandable to practitioners in that area". The authors' statement becomes important because it stimulates the researcher's and author's awareness of the theory about the proper use of a GT methodology. In this research, we sought to identify codes carefully to support understanding by anyone interested in the SE literature or the software industry. In sample 2, the codes were extracted from phrases cited by the target audience. This allowed the proposed theory to be aligned with the perceptions of software development practitioners.

Another important aspect related to this factor refers to the fact that the data collection from the second sample, from the survey responses applied in the industry, occurred in a relatively shorter time. This is justified by the fact that there was a learning of process execution during the first sample. Therefore, only the sentences that related to the research question were retrieved. This allowed the researcher to be sure that the codes previously collected were indeed saturated, meaning that no new code was perceived through additional survey coding.

6.3.0.3 Generality. Theory must be summarily abstract to be a general guide without losing its relevance.

Glaser et al. (1968) state that "A theory with sufficiently comprehensible controllable concepts, which is understandable and comprehensible, gives anyone wishing to apply these concepts to bring change into a controllable theoretical position in various situations". In summary, the theory must ensure that the person using it has sufficient control in the situations it encounters to make the application of the theory considered. That is, the theory must provide an understanding for anyone who wishes to analyze it, revise it, or propose changes. For this, the theory must have a sufficient number of codes and relationships that explain the interaction between them.

The proposed theory meets this premise as it presents a comprehensive set of categorized and related codes when possible. All details are presented in Section 6.1.

6.3.0.4 Control. Theory serves as a general guide and allows the person to fully understand the situation.

During the writing of this dissertation, we tried to provide all the details of the path followed until the theory was created. Explanations were given on how the initial sampling of the data collection was selected and each new phase and how the decisions on the choice of the theoretical sampling originated.

Comments on the analyzes that were performed on the data for the creation of codes, as seen in section 6.1.2. Discovery of categories in selective coding are also described in section 6.1.3.

For each reading conducted and analyzed, the textual parts that underlie the creation of the concepts were extracted from the research and inserted in appropriate places so that the researcher's thinking could be followed during the development of this study. Thus, the pattern adopted by this study is expected to be clear enough to follow up on the developments and developments in this research.

According to GT's assumptions, another researcher may come up with a different theory from the same data, but anyone should be able to follow the researcher's path and agree or disagree with what was done.

6.4 THREATS TO VALIDITY

This section presents and discusses threats that might affect the validity of this study.

Construct Validity: More codes could have been derived from the data. This threat may have influenced the study, but it is impossible to know when data analysis is sufficient. Another additional threat is the adoption of sustainability dimensions as categories. However, we provide this data considering the objective of this work, which was to identify sustainability concerns according to the dimensions of sustainability. Other threat to this study is a possible inadequate code capture without losing relevant details. To mitigate the risk, we adopted memos as the GT method proposes.

Internal Validity: An internal limitation may be the selection of primary studies and survey responses for the sample. We understand that both the number of studies and the number of responses obtained may not adequately represent the entire population of sustainable SE researchers and software professionals, characterizing a threat to internal validity. However, as the selection was the result of systematic studies, we believe that this set can be representative.

External Validity: The limited number of studies contained in the data source could not represent the state-of-the-art in sustainable SE, once the systematic mapping study was carried out in 2018 (MOURÃO et al., 2018). In addition, even though the state-of-the-art is represented, the mapping study only presents an outline, which has the bias that it elaborated the search and selection criteria. Therefore, several relevant studies may have been left out. Likewise, the survey open responses number (KARITA et al., 2019) could not be representative in the GT data source. However, we understand that the extension of the both studies could be a research opportunity.

Reliability: Although GT offers rigorous procedures for data collection and analysis, qualitative research is generally subject to researcher bias. Certainly other researchers could form a different interpretation and theory after analyzing the same data, but it is believed that at least the main insights would be preserved.

6.5 CHAPTER SUMMARY

In this Chapter, we presented a GT aimed to understand the sustainable software development. It was built on the perception of researchers in sustainable SE and practitioners of the software industry, aiming to identify the main sustainability concerns in light of the concepts discussed in the literature.

In order to achieve our goal, we analyzed the collected data and derived a initial theory using GT procedures (GLASER et al., 1968). The proposed understanding emerged from the identification of sustainable concerns present in 48 study texts retrieved in Chapter 4 and survey responses applied on software industry presented in Chapter 5. Based on these concerns, propositions were established based on the present relationships between them. Finally, the elements that are part of the process were explained by defining six large groupings of concept categories.

Overall, this study showed that all stages of the development process are impacted by the dimensions of sustainability. It is also evident that software engineers interested in sustainable software development should focus on identifying sustainable requirements at an early stage of the project. To this end, they must also identify key stakeholders in this new context, sustainability experts. Given the elicitation of sustainable requirements, it is already possible to list the quality requirements that will affect the final product. Most of these attributes will be the categories provided by ISO/IEC 9126 (International Standards Organisation (ISO), 1991), such as reusability, maintainability, performance, etc. Still under discussion in the literature, there are two quite cited concerns that permeate all stages of the sustainable software development process: energy efficiency and energy consumption. It was also noted that some sustainable concerns can generate trade-offs.

Finally, the evaluation of the theory proposed in this study was performed based on the factors defined by Strauss and Corbin (1990).

In the next Chapter, we present the final remarks of this dissertation, as well as the contributions, limitations and future works.

Chapter 7

CONCLUSION

The sustainable software development theme is relatively new to the SE research community and has grown considerably. Much of this is due to the fact that the scientific community is increasingly warning society about the negative impacts that our actions are having on the environment. However, when we bring this issue into the realm of software development, we run into gaps, as this issue is still nebulous for software engineers and developers.

Although several researches have been conducted in the SE area search of solutions and strategies that mainly promote the reduction of energy consumption by software applications, the knowledge that has been developed is still incipient in the literature. The first effort in this investigation was to leverage state-of-the-art approaches discussing sustainability in the SE field (MOURÃO et al., 2018). There, we observed that there is still a gap about what, in fact, is the development of sustainable software. The lack of such understanding could prevent industry from building software with sustainability awareness.

This dissertation presented a set of qualitative studies in a multi-method approach, in order to present a characterization of the sustainable SE field. For this, empirical evidence was produced from a survey (Chapter 5) and a GT (Chapter 6).

The research concluded that the sustainable software development is characterized through the following explanations: (a) technical, environmental and social concerns must be present in all phases of sustainable software development. This means that researchers and software engineers are concerning about considering the longevity of the produced software and environmental resources, in addition to social welfare; (b) the identification of sustainable requirements must occur in the initial phase of the project (c) with the support of experts engaged with sustainability, who must be part of the group of stakeholders; and (d) the use of sustainable concerns can generate trade-offs in the project, for example: performance x energy efficiency and defects x power consumption.

The yielded results represent an academic contribution to the SE community, since: (1) no other study that investigated sustainable concerns using GT methodology was identified; (2) the number of primary studies read was relevant to produce insights into sustainable concerns; and (3) explanations as to how these sustainable concerns relate in the software development process constitute an important step towards the maturing of sustainable SE.

7.1 CONTRIBUTIONS

7.1.1 Research Contributions

The main contributions of this dissertation are:

- **SE empirical research:** This research contributes significantly to the expansion of knowledge in Sustainable SE, as it is an exploratory and systematic approach, which has drawn from the social actors involved, researchers and industry, the way to reach empirical evidence that can generate future actions;

- **Sustainable SE domain characterization:** The results obtained in this study are a first step towards helping the sustainable SE research community and industry better understand the characteristics that will best define sustainable software development;

- **Provide conceptual support for the software industry:** These findings can become a starting point for companies to obtain an in-depth understanding of sustainable software development. Since the proposal brings evidence of "what" are the sustainable concerns, it is up to new studies to highlight the "how" should be done. This evidence will also serve as a basis for the research community to move towards identifying techniques and tools that support sustainable software development.

7.2 FUTURE PERSPECTIVES

There are still many important research gaps to be explored and investigated. In this section, we list some future research opportunities that emerged from the multi-method approach applied in this dissertation. Based on the results obtained in the studies, in addition to findings related to the existing literature, the following directions emerge as perspectives for future work:

- We expect to expand the systematic mapping study reported in Chapter 4 to add new journals and conference proceedings as we had restricted the search to studies published up to December 2017. Therefore, we believe that new studies have emerged since then, which may promote new insights;

- It is suggested to expand the survey described in Chapter 5 with a larger number of participants, from different locations, considering the evolutions identified in the previous study. This replication would allow the results to be more generalizable and more specific analyzes, such as the adoption of sustainable practices by region;

- Another necessary extension will be to conduct interviews with industry practitioners in order to deepen the qualitative results of the survey reported in Chapter 5;
- We understand that it is still necessary to conduct a study in industry to evaluate the theory produced in Chapter 6 and its applicability. With this study, individuals are expected to provide feedback to improve the proposal;
- Additionally, the theory proposed is not absolute or final, since studies using grounded theory in general are not intended to be definitive. Therefore, we understand that extensions of it based on unnoticed aspects, refined details of categories and concepts, or discovery of new concepts and relationships between them are welcome;
- Other research opportunity would be to expand the SE domain ontologies available in the literature to incorporate sustainable concepts and reduce the lack of definitions.

REFERENCES

AHMED, F.; MAHMOOD, H.; ASLAM, A. Green computing and software defects in open source software: An empirical study. In: IEEE. *International Conference on Open Source Systems & Technologies*. Lahore, Pakistan, 2014. p. 65–69.

ALAVI, M.; CARLSON, P. A review of mis research and disciplinary development. *Journal of management information systems*, Taylor & Francis, v. 8, n. 4, p. 45–62, 1992.

AMSEL, N.; IBRAHIM, Z.; MALIK, A.; TOMLINSON, B. Toward sustainable software engineering. In: ACM. *Proceedings of the 33rd International Conference on Software Engineering*. Honolulu, USA, 2011. p. 976–979.

ANWAR, H.; PFAHL, D. Towards greener software engineering using software analytics: A systematic mapping. In: IEEE. *Software Engineering and Advanced Applications (SEAA), 2017 43rd Euromicro Conference on*. Vienna, Austria, 2017. p. 157–166.

BAJPAI, N. *Business research methods*. India: Pearson Education India, 2011.

BARBARA, K.; STUART, C. Guidelines for performing systematic literature reviews in software engineering (version 2.3.). *Staffordshire, UK: University of Keele*, 2007.

BECKER, C. Sustainability and longevity: Two sides of the same quality? *mental*, Citeseer, v. 20, p. 21, 2014.

BECKER, C.; CHITCHYAN, R.; DUBOC, L.; EASTERBROOK, S.; PENZENSTADLER, B.; SEYFF, N.; VENTERS, C. C. Sustainability design and software: The karlskrona manifesto. In: IEEE PRESS. *Proceedings of the 37th International Conference on Software Engineering-Volume 2*. Florence, Italy, 2015. p. 467–476.

BEGHOURA, M. A.; BOUBETRA, A.; BOUKERRAM, A. Green software requirements and measurement: random decision forests-based software energy consumption profiling. *Requirements Engineering*, Springer, Berlin, Heidelberg, v. 22, n. 1, p. 27–40, 2017.

BELL, E.; BRYMAN, A.; HARLEY, B. *Business research methods*. New York, U.S.A: Oxford university press, 2018.

BERNTSEN, K. R.; OLSEN, M. R.; LIMBU, N.; TRAN, A. T.; COLOMO-PALACIOS, R. Sustainability in software engineering-a systematic mapping. In: SPRINGER. *International Conference on Software Process Improvement*. Cham, 2016. p. 23–32.

CALERO, C.; BERTOA, M. 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) co-located with ACM Conference on Aspect-Oriented software Development (AOSD)*, 2013.

CALERO, C.; MORAGA, M.; BERTOA, M. F. Towards a software product sustainability model. *arXiv preprint arXiv:1309.1640*, 2013.

CALERO, C.; PIATTINI, M. *Green in software engineering*. New York, U.S.A: Springer, 2015.

CALERO, C.; PIATTINI, M. Puzzling out software sustainability. *Sustainable Computing: Informatics and Systems*, Elsevier, v. 16, p. 117–124, 2017.

CALIENES, G. G. *Requirements prioritization framework for developing green and sustainable software using ANP-based decision making*. Hamburg: EnviroInfo, 2013.

CAPRA, E.; FRANCALANCI, C.; SLAUGHTER, S. A. Measuring application software energy efficiency. *IT Professional*, IEEE, v. 14, n. 2, p. 54–61, 2012.

CASTELLAN, C. M. Quantitative and qualitative research: A view for clarity. *International journal of education*, Macrothink Institute Inc., v. 2, n. 2, p. 1, 2010.

CHITCHYAN, R.; BETZ, S.; DUBOC, L.; PENZENSTADLER, B.; EASTERBROOK, S.; PONSARD, C.; VENTERS, C. Evidencing sustainability design through examples. Ottawa, Canada, 2015.

COLEMAN, G.; O'CONNOR, R. Using grounded theory to understand software process improvement: A study of irish software product companies. *Information and Software Technology*, Elsevier, v. 49, n. 6, p. 654–667, 2007.

COLLINS, H. *Creative research: the theory and practice of research for the creative industries*. London: Bloomsbury Publishing, 2018.

COOK, J.; ORESKES, N.; DORAN, P. T.; ANDEREGG, W. R.; VERHEGGEN, B.; MAIBACH, E. W.; CARLTON, J. S.; LEWANDOWSKY, S.; SKUCE, A. G.; GREEN, S. A. et al. Consensus on consensus: a synthesis of consensus estimates on human-caused global warming. *Environmental Research Letters*, IOP Publishing, v. 11, n. 4, 2016.

CORBIN, J. M.; STRAUSS, A. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, Springer, v. 13, n. 1, p. 3–21, 1990.

CRESWELL, J. W.; CLARK, V. L. P. *Designing and conducting mixed methods research*. London: Sage publications, 2017.

CROWTHER, D.; LANCASTER, G. *Research methods*. United Kingdom: Routledge, 2012.

DICK, M.; NAUMANN, S. Enhancing software engineering processes towards sustainable software product design. Aachen, p. 706–715, 2010.

DICK, M.; NAUMANN, S.; KUHN, N. A model and selected instances of green and sustainable software. In: *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*. Berlin, Heidelberg: Springer, 2010. p. 248–259.

DYBA, T.; DINGSOYR, T.; HANSSEN, G. K. Applying systematic reviews to diverse study types: An experience report. In: IEEE. *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. DC United States, 2007. p. 225–234.

EASTERBROOK, S.; SINGER, J.; STOREY, M.-A.; DAMIAN, D. Selecting empirical methods for software engineering research. In: *Guide to advanced empirical software engineering*. London: Springer, 2008. p. 285–311.

ERDELYI, K. Special factors of development of green software supporting eco sustainability. In: IEEE. *2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY)*. Subotica, Serbia, 2013. p. 337–340.

GALLIERS, R. D.; LAND, F. F. Choosing appropriate information systems research methodologies. *Communications of the ACM*, ACM, v. 30, n. 11, p. 901–902, 1987.

GARCÍA-MIRELES, G. A. Environmental sustainability in software process improvement: a systematic mapping study. In: SPRINGER. *International Conference on Software Process Improvement*. Springer, Cham, 2016. p. 69–78.

GARCÍA-MIRELES, G. A.; MORAGA, M. Á.; GARCÍA, F.; CALERO, C.; PIATTINI, M. Interactions between environmental sustainability goals and software product quality: a mapping study. *Information and Software Technology*, Elsevier, 2017.

GERRING, J.; THOMAS, C. W. Quantitative and qualitative: A question of comparability. *International Encyclopedia of political science: Sage*, London, v. 7, p. 2189–2196, 2011.

GLASER, B. G. Constructivist grounded theory? Forum qualitative sozialforschung/forum: Qualitative social research, Berlin, v. 3, n. 3, 2002.

GLASER, B. G.; STRAUSS, A. L.; STRUTZEL, E. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, LWW, v. 17, n. 4, p. 364, 1968.

GLASS, R. L.; VESSEY, I. Contemporary application-domain taxonomies. *IEEE Software*, IEEE, v. 12, n. 4, p. 63–76, 1995.

GOULDING, C. *Grounded theory: A practical guide for management, business and market researchers*. London: Sage, 2002.

GREEN, I. H. Principles and practices. *San Murugesan, GR*, 2012.

GROHER, I.; WEINREICH, R. An interview study on sustainability concerns in software development projects. In: IEEE. *43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. Vienna, Austria, 2017. p. 350–358.

HAMMARBERG, K.; KIRKMAN, M.; LACEY, S. de. Qualitative research methods: when to use them and how to judge them. *Human reproduction*, Oxford University Press, v. 31, n. 3, p. 498–501, 2016.

HILTY, L. M.; ARNFALK, P.; ERDMANN, L.; GOODMAN, J.; LEHMANN, M.; WÄGER, P. A. The relevance of information and communication technologies for environmental sustainability–a prospective simulation study. *Environmental Modelling & Software*, Elsevier, v. 21, n. 11, p. 1618–1629, 2006.

HODA, R.; NOBLE, J. Becoming agile: a grounded theory of agile transitions in practice. In: IEEE PRESS. *Proceedings of the 39th International Conference on Software Engineering*. Buenos Aires, Argentina, 2017. p. 141–151.

HUBER, M. Z.; HILTY, L. M.; GLINZ, M. Uncovering sustainability requirements: An exploratory case study in canteens. Ottawa, Canada, p. 35–44, 2015.

International Standards Organisation (ISO). *International Standard ISO/IEC 9126. Information technology: Software product evaluation: Quality characteristics and guidelines for their use*. 1991.

International Standards Organisation (ISO). *International Standard ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. 2011.

JAGROEP, E.; BROEKMAN, J.; WERF, J. M. E. V. D.; LAGO, P.; BRINKKEMPER, S.; BLOM, L.; VLIET, R. V. Awakening awareness on energy consumption in software engineering. In: IEEE. *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*. Buenos Aires, Argentina, 2017. p. 76–85.

JOHANN, T.; DICK, M.; KERN, E.; NAUMANN, S. Sustainable development, sustainable software, and sustainable software engineering: an integrated approach. In: IEEE. *Humanities, Science & Engineering Research (SHUSER), International Symposium on*. Kuala Lumpur, Malaysia, 2011. p. 34–39.

KARITA, L.; MOURÃO, B. C.; MACHADO, I. Software industry awareness on green and sustainable software engineering: a state-of-the-practice survey. Proceedings of the XXXIII Brazilian Symposium on Software Engineering, Salvador, Brasil, p. 501–510, 2019.

KASUNIC, M. *Designing an effective survey*. Pittsburgh, PA, 2005.

KHANDELWAL, B.; KHAN, S.; PARVEEN, S. Cohesive analysis of sustainability of green computing in software engineering. *International Journal of Emerging Trends Technology in Computer Science (IJETTCS)*, v. 6, n. 4, p. 011–016, 2017.

KITCHENHAM, B. A.; PFLEEGER, S. L. Principles of survey research part 2: designing a survey. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 27, n. 1, p. 18–20, 2002.

KOÇAK, S. A.; ALPTEKIN, G. I.; BENER, A. B. Integrating environmental sustainability in software product quality. Ottawa, Canada, p. 17–24, 2015.

LAGO, P.; JANSEN, T. Creating environmental awareness in service oriented software engineering. In: SPRINGER. *International conference on service-oriented computing*. Berlin, Heidelberg, 2010. p. 181–186.

LAGO, P.; KOCAK, S. A.; CRNKOVIC, I.; PENZENSRADLER, B. et al. Framing sustainability as a property of software quality. v. 58, n. 10, p. 70–78, 2015.

LAGO, P.; PENZENSTADLER, B. Reality check for software engineering for sustainability—pragmatism required. *Journal of Software: Evolution and process*, Wiley Online Library, v. 29, n. 2, p. e1856, 2017.

LI, D.; SAHIN, C.; CLAUSE, J.; HALFOND, W. G. Energy-directed test suite optimization. In: IEEE. *2nd International Workshop on Green and Sustainable Software (GREENS)*. San Francisco, CA, USA, 2013. p. 62–69.

MANOTAS, I.; BIRD, C.; ZHANG, R.; SHEPHERD, D.; JASPAN, C.; SADOWSKI, C.; POLLOCK, L.; CLAUSE, J. An empirical study of practitioners' perspectives on green software engineering. In: IEEE. *IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. Austin, TX, USA, 2016. p. 237–248.

MARIMUTHU, C.; CHANDRASEKARAN, K. Software engineering aspects of green and sustainable software: A systematic mapping study. In: ACM. *Proceedings of the 10th Innovations in Software Engineering Conference*. Jaipur India, 2017. p. 34–44.

MOURÃO, B. C.; KARITA, L.; MACHADO, I. do C. Green and sustainable software engineering-a systematic mapping study. In: ACM. *Proceedings of the 17th Brazilian Symposium on Software Quality*. Salvador, Brasil, 2018. p. 121–130.

MUNOZ, D.-J.; PINTO, M.; FUENTES, L. Hadas and web services: Eco-efficiency assistant and repository use case evaluation. In: IEEE. *International Conference in Energy and Sustainability in Small Developing Economies (ES2DE)*. Funchal, Portugal, 2017. p. 1–6.

MURUGESAN, S. Harnessing green it: Principles and practices. *IT professional*, IEEE, v. 10, n. 1, p. 24–33, 2008.

MYERS, M. D. *Qualitative research in business and management*. London: Sage, 2013.

NAKAGAWA, E. Y.; SCANNAVINO, K. R. F.; FABBRI, S. C. P. F.; FERRARI, F. C. *Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática*. Brasil: Elsevier Brasil, 2017.

NAUMANN, S.; DICK, M.; KERN, E.; JOHANN, T. The greensoft model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems*, Elsevier, v. 1, n. 4, p. 294–304, 2011.

NETO, P. A. M. S.; MACHADO, I. C.; MCGREGOR, J. D.; ALMEIDA, E. S.; MEIRA, S. R. L. A systematic mapping study of software product lines testing. *Information and Software Technology*, Elsevier, v. 53, n. 5, p. 407–423, 2011.

NIKZAD, N.; CHIPARA, O.; GRISWOLD, W. G. Ape: an annotation language and middleware for energy-efficient mobile application development. In: ACM. *Proceedings of the 36th International Conference on Software Engineering*. New York, NY, United States, 2014. p. 515–526.

OYEDEJI, A. S. et al. Early investigation towards defining and measuring sustainability as a quality attribute in software systems. 2016.

PENZENSTADLER, B. Infusing green: Requirements engineering for green in and through software systems. Karlskrona, Sweden, p. 44–53, 2014.

PENZENSTADLER, B.; FEMMER, H. A generic model for sustainability with process- and product-specific instances. In: ACM. *Proceedings of the workshop on Green in/by software engineering*. New York, NY, United States, 2013. p. 3–8.

PENZENSTADLER, B.; FEMMER, H.; RICHARDSON, D. Who is the advocate? stakeholders for sustainability. In: IEEE. *2013 2nd International workshop on green and sustainable software (GREENS)*. San Francisco, CA, USA, 2013. p. 70–77.

PENZENSTADLER, B.; MEHRABI, J.; RICHARDSON, D. J. Supporting physicians by re4s: Evaluating requirements engineering for sustainability in the medical domain. In: IEEE PRESS. *Proceedings of the Fourth International Workshop on Green and Sustainable Software*. Florence, Italy, 2015. p. 36–42.

PENZENSTADLER, B.; RATURI, A.; RICHARDSON, D.; TOMLINSON, B. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. *IEEE software*, IEEE, v. 31, n. 3, p. 40–47, 2014.

PENZENSTADLER, B.; RATURI, A.; RICHARDSON, D.; CALERO, C.; FEMMER, H.; FRANCH, X. Systematic mapping study on software engineering for sustainability (SE4S). In: ACM. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, United States, 2014. p. 14.

PÉREZ-CASTILLO, R.; PIATTINI, M. Analyzing the harmful effect of god class refactoring on power consumption. *IEEE software*, IEEE, v. 31, n. 3, p. 48–54, 2014.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. Swindon, United Kingdom, v. 8, p. 68–77, 2008.

PROCACCIANTI, G.; LAGO, P.; LEWIS, G. A. A catalogue of green architectural tactics for the cloud. In: IEEE. *IEEE 8th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems*. Victoria, BC, Canada, 2014. p. 29–36.

RATURI, A.; PENZENSTADLER, B.; TOMLINSON, B.; RICHARDSON, D. Developing a sustainability non-functional requirements framework. In: ACM. *Proceedings of the 3rd International Workshop on Green and Sustainable Software*. New York, NYU, United States, 2014. p. 1–8.

SAHIN, C.; CAYCI, F.; GUTIÉRREZ, I. L. M.; CLAUSE, J.; KIAMILEV, F.; POLLOCK, L.; WINBLADH, K. Initial explorations on design pattern energy usage. In: IEEE. *First International Workshop on Green and Sustainable Software (GREENS)*. Zurich, Switzerland, 2012. p. 55–61.

SAPUTRI, T. R. D.; LEE, S.-W. Incorporating sustainability design in requirements engineering process: A preliminary study. In: *Asia Pacific Requirements Engineering Conference*. Singapore: Springer, 2016. p. 53–67.

SAUNDERS, M.; LEWIS, P.; THORNHILL, A. Research methods. *Business Students*, 2007.

SEAMAN, C. B.; MENDONÇA, M. G.; BASILI, V. R.; KIM, Y.-M. User interface evaluation and empirically-based evolution of a prototype experience management tool. *IEEE Transactions on Software Engineering*, IEEE, v. 29, n. 9, p. 838–850, 2003.

SHULL, F.; SINGER, J.; SJØBERG, D. I. *Guide to advanced empirical software engineering*. London: Springer, 2007.

SJØBERG, D. I.; DYBÅ, T.; ANDA, B. C.; HANNAY, J. E. Building theories in software engineering. In: *Guide to advanced empirical software engineering*. London: Springer, 2008. p. 312–336.

STRAUSS, A.; CORBIN, J. *Basics of qualitative research*. London: Sage publications, 1990.

TAINTER, J. A. Social complexity and sustainability. *ecological complexity*, Elsevier, v. 3, n. 2, p. 91–103, 2006.

TATE, K. *Sustainable software development: an agile perspective*. New York, U.S.A: Addison-Wesley Professional, 2005.

VENTERS, C.; JAY, C.; LAU, L.; GRIFFITHS, M. K.; HOLMES, V.; WARD, R.; AUSTIN, J.; DIBSDALE, C. E.; XU, J. Software sustainability: The modern tower of babel. In: RWTH AACHEN UNIVERSITY. *Proceedings of the Third International*

Workshop on Requirements Engineering for Sustainable Systems co-located with 22nd International Conference on Requirements Engineering (RE 2014). Germany, 2014. v. 1216.

VENTERS, C.; LAU, L.; GRIFFITHS, M.; HOLMES, V.; WARD, R.; JAY, C.; DIBSDALE, C.; XU, J. The blind men and the elephant: Towards an empirical evaluation framework for software sustainability. *Journal of Open Research Software*, Ubiquity Press, London, v. 2, n. 1, 2014.

WIERINGA, R.; MAIDEN, N.; MEAD, N.; ROLLAND, C. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer, Berlin, Heidelberg, v. 11, n. 1, p. 102–107, 2006.

WILSON, J. *Essentials of business research: A guide to doing your research project*. London: Sage, 2014.

WOLFRAM, N.; LAGO, P.; OSBORNE, F. Sustainability in software engineering. *Sustainable Internet and ICT for Sustainability (SustainIT)*, Funchal, Portugal, 2017.

Appendix

A

SMS - PRIMARY STUDIES

This Appendix lists the primary studies selected in SMS presented in Chapter 4.

A.1 QUALITY STUDIES SCORES

Table A.1 Selected primary studies

P#	Primary Study	Year	QS
P1	A Catalogue of Green Architectural Tactics for the Cloud	2014	10
P2	A framework for estimating the energy consumption induced by a distributed system's architectural style	2009	10.5
P3	A generic model for sustainability with process and product-specific instances	2013	11
P4	A Green Software Development Life Cycle for Cloud Computing	2013	9
P5	A model and selected instances of green and sustainable software	2010	8
P6	A practical model for evaluating the energy efficiency of software applications	2014	8
P7	A preliminary study of the impact of software engineering on Green IT	2012	9.5
P8	A programming model for sustainable software	2015	9
P9	A software engineer's energy-optimization decision support framework	2014	11
P10	An Automated Analysis of the Branch Coverage and Energy Consumption Using Concolic Testing	2017	9
P11	An Empirical Study of Practitioners' Perspectives on Green Software Engineering	2016	10
P12	An Energy Consumption Perspective on Software Architecture	2015	9
P13	An interview study on sustainability concerns in software development projects	2017	9
P14	An investigation into energy-saving programming practices for Android smartphone app development	2014	10
P15	An ISO/IEC 33000-compliant measurement framework for software process sustainability assessment	2014	8
P16	Analyzing the harmful effect of god class refactoring on power consumption	2014	10
P17	APE: an annotation language and middleware for energy-efficient mobile application development	2014	11

Continued on next column

Table A.1 – Continued from previous column

P#	Primary Study	Year	QS
P18	Awakening awareness on energy consumption in software engineering	2017	9
P19	Characterizing sustainability requirements. A new species, red herring, or just an odd fish?	2017	9
P20	Creating environmental awareness in service oriented software engineering	2011	7
P21	Derivation of Green Metrics for Software	2013	7
P22	Detecting energy bugs and hotspots in mobile apps	2014	11
P23	Developing a sustainability non-functional requirements framework	2014	8.5
P24	Empirical evaluation of two best practices for energy-efficient software development	2016	9
P25	Energy efficiency embedded service lifecycle: Towards an energy efficient cloud computing architecture	2014	10
P26	Energy-aware software: Challenges, opportunities and strategies	2013	9
P27	Energy-directed test suite optimization	2013	11
P28	Enhancing Software Engineering Processes towards Sustainable Software Product Design	2010	7
P29	Enhancing Sustainability of the Software Life Cycle via a Generic Knowledge Base	2010	8
P30	Estimating mobile application energy consumption using program analysis	2013	9.5
P31	Evaluating energy efficiency of Internet of Things software architecture based on reusable software components	2017	11
P32	Evidencing sustainability design through examples	2015	10
P33	Green computing and Software Defects in open source software: An Empirical study	2014	9
P34	Green software development model: An approach towards sustainable software development	2011	7.5
P35	Green Software Engineering with Agile Methods	2013	8
P36	Green software requirements and measurement: random decision forests-based software energy consumption profiling	2015	10.5
P37	Green software services: From requirements to business models	2013	8
P38	Green tracker: a tool for estimating the energy consumption of software	2010	10
P39	Greenadvisor: A tool for analyzing the impact of software evolution on energy consumption	2015	10
P40	GreenC5: An adaptive, energy-aware collection for green software development	2017	10
P41	Green-J Model: a novel approach to measure energy consumption of modified condition/decision coverage using concolic testing	2017	9
P42	HADAS and web services: Eco-efficiency assistant and repository use case evaluation	2017	10
P43	How do code refactorings affect energy usage?	2014	10
P44	Impacts of software and its engineering on the carbon footprint of ICT	2015	9
P45	Incorporating Sustainability Design in Requirements Engineering Process: A Preliminary Study	2016	10
P46	Infusing green: Requirements engineering for green in and through software systems	2014	9.5
P47	Initial explorations on design pattern energy usage	2012	10
P48	Integrating Aspects of Carbon Footprints and Continuous Energy Efficiency Measurements into Green and Sustainable Software Engineering	2013	9

Continued on next column

Table A.1 – Continued from previous column

P#	Primary Study	Year	QS
P49	Integrating environmental sustainability in software product quality	2015	9.5
P50	"Is software green"? Application development environments and energy efficiency in open source applications"	2012	9
P51	Measuring application software energy efficiency	2012	10
P52	Measuring Software Sustainability	2003	8
P53	Measuring the Sustainability Performance of Software Projects	2010	8
P54	Requirements Prioritization Framework for Developing Green and Sustainable Software using ANP -based Decision Making	2013	8.5
P55	Requirements: the key to sustainability	2015	10
P56	Safety, security, now sustainability: the nonfunctional requirement for the 21st century	2014	10
P57	Self-optimization of the energy footprint in service-oriented architectures	2010	9
P58	Software Sustainability from a Process-Centric Perspective	2012	9
P59	Supporting Physicians by RE4S: Evaluating Requirements Engineering for Sustainability in the Medical Domain	2015	8.5
P60	Sustainability design and software: The karlskrona manifesto	2015	8
P61	Sustainability design in requirements engineering: state of practice	2016	10
P62	Sustainability guidelines for long-living software systems	2012	9
P63	Sustainable development, sustainable software, and sustainable software engineering: An integrated approach	2011	8
P64	Sustainable software engineering: process and quality models, life cycle, and social aspects	2015	8
P65	The Contexto Framework: Leveraging Energy Awareness in the Development of Context-Aware Applications	2016	11
P66	The greensoft model: A reference model for green and sustainable software and its engineering Sustainable Computing	2011	8
P67	The Impact of Improving Software Functionality on Environmental Sustainability	2013	10
P68	The impact of user choice on energy consumption	2014	10
P69	Towards power reduction through improved software design	2012	10
P70	Towards sustainable software criteria: Rescue operation and disaster management system model	2013	8
P71	Uncovering sustainability requirements: An exploratory case study in canteens	2015	10
P72	Understanding Green Software Development: A Conceptual Framework	2015	9.5
P73	Who is the Advocate?: Stakeholders for Sustainability	2013	10
P74	Sustainability requirements for connected health applications	2017	10
P75	Puzzling out Software Sustainability	2017	9

Legend: P#: Primary Study ID, QS: Quality Score

Appendix B

SURVEY INSTRUMENT

This Appendix shows the instrument applied in survey study presented in Chapter 5. The instrument was written in Portuguese because it is the language of the target audience.

B.1 PRESENTATION FORM

Survey - Desenvolvimento de Software Verde e Sustentável

Olá, participante!

O objetivo deste survey consiste em investigar e identificar a sua percepção e conscientização a respeito do tema "Desenvolvimento de Software Verde e Sustentável" e identificar "se" e "de que forma" a empresa em que você trabalha tem aplicado práticas sustentáveis no seu processo de desenvolvimento de software.

Você foi convidado(a) a participar desta investigação porque atua, profissionalmente, no processo de desenvolvimento de software. Esta pesquisa é parte integrante da dissertação de mestrado das estudantes Brunna Caroline e Leila Karita, do Programa de Pós-Graduação em Ciência da Computação da UFBA. Obrigada, a sua participação é muito importante para nossa pesquisa sobre "Engenharia de Software Verde e Sustentável"!

Contato:

Bruna Caroline	Leila Karita
brunna.caroline@ufba.br	leila.karita@ufba.br
(71) 99339-5026 (Whatsapp)	(71) 98795-1127 (Whatsapp)

Datas importantes para o respondente:
1° lembrete: 26/11/2018
2° lembrete: 03/12/2018
Prazo final: 04/01/2019

Atenciosamente,
Brunna Caroline e Leila Karita

Universidade Federal da Bahia (UFBA) - Instituto de Matemática e Estatística - Departamento de Ciência da Computação - Grupo de Pesquisa RISE
Av. Adhemar de Barros, s/n, sala 280, Ondina, 40170-110, Salvador – BA, Tel.: (71) 3283-6266/6293

B.2 CONSENT FORM

Declaração:
Declaro que fui informado(a) que os dados aqui fornecidos serão utilizados, única e exclusivamente, para os fins desta pesquisa. Que as pesquisadoras comprometem-se em não repassar informações que identifiquem o meu nome ou da empresa em que trabalho na participação da presente pesquisa. Que a minha participação não envolverá nenhum risco físico e nem custo ou ganho financeiro. Que sou livre para deixar de participar da pesquisa a qualquer momento, sem nenhum prejuízo. Que este questionário utiliza o pacote de aplicativo Google Docs, portanto, a coleta e o uso de informações do Google estão sujeitos à Política de privacidade do Google (https://www.google.co.uk/policies/privacy/).

Declaro que eu fui informado(a) dos objetivos e políticas de privacidade do presente estudo de maneira clara e detalhada. *

- Concordo
- Não concordo

B.3 SURVEY QUESTIONS

2. Sobre você....

Esta seção possui perguntas referentes à caracterização do perfil do profissional entrevistado.

2.1. Nome completo:

Your answer

2.2. Idade: *

Choose

2.3. Sexo: *

- Masculino
- Feminino
- Outros

2.4. Titulação máxima: *

- Ensino Médio
- Graduação
- Especialização
- Mestrado
- Doutorado
- Pós Doutorado
- Other: ______

2.5. Qual a sua experiência profissional no mercado de desenvolvimento de software? *

- até 1 ano
- 1 a 3 anos
- 4 a 6 anos
- 7 a 10 anos
- Acima de 10 anos

3. Sobre a empresa...

Esta seção possui perguntas referentes à caracterização da empresa onde o profissional entrevistado atua.

3.1. Estado (UF) da empresa em que você trabalha: *

Choose ▾

3.2. Qual é o seguimento da organização em que você trabalha? *

- ○ Fábrica de Software
- ○ Consultoria
- ○ Fábrica de Testes
- ○ Startup
- ○ Governo
- ○ Other: ____________

3.3. Qual é o porte da empresa em que você trabalha? *

- ○ Microempresa (Até 9 funcionários)
- ○ Pequeno porte (Até 49 funcionários)
- ○ Médio porte (Até 99 funcionários)
- ○ Grande porte (Acima de 99 funcionários)

3.4. Quanto tempo de mercado a empresa em que você trabalha possui? *

- ○ até 1 ano
- ○ até 5 anos
- ○ até 10 anos
- ○ Acima de 10 anos

3.5. Qual é a função que você exerce na empresa? *

- [ ] Analista de Sistemas
- [ ] Analista de Negócios
- [ ] Analista de Requisitos
- [ ] Arquiteto de Software
- [ ] Desenvolvedor
- [ ] Gerente de Projetos
- [ ] Gerente de Portfólio
- [ ] Líder de Projetos
- [ ] Product Owner
- [ ] Pesquisador
- [ ] Scrum Master
- [ ] Testador
- [ ] Consultor
- [ ] Other: ______

3.6. Quais certificações a empresa em que você trabalha possui? *

- [ ] CMMI
- [ ] MPS-BR
- [ ] ISO IEC 25000
- [ ] ISO IEC 29010
- [ ] TMMI
- [ ] Não sei informar.
- [ ] Nenhum
- [ ] Other: ______

3.7. Se a sua resposta na questão anterior foi CMMI, MPS.Br ou TMMI, por favor, informe o nível de maturidade:

Your answer ______

3.8. Na sua opinião, qual é o nível de consciência ambiental e uso de práticas sustentáveis (destinação adequada e reciclagem do lixo, pilhas, baterias e pneus; atendimento às legislações ambientais; economia de água, energia e papel, etc) da empresa em que você trabalha? *

- ◯ Iniciante (Atende poucas legislações e realiza algumas práticas)
- ◯ Intermediário (Atende diversas legislações e realiza diversas práticas)
- ◯ Expert (Atende todas as legislações, realiza e incentiva diversas práticas)
- ◯ Não atende às legislações
- ◯ Não sei informar

3.9. Em relação à questão anterior, informe qual(is) é(são) a(s) legislação(ões) atendidas pela empresa:

Your answer

4. Objeto da Pesquisa

O objetivo desta seção consiste em obter uma percepção do conhecimento do profissional entrevistado sobre os conceitos de sustentabilidade no contexto do software e sobre o uso de práticas sustentáveis na empresa em que trabalha.

4.1. A respeito do assunto Sustentabilidade no processo de desenvolvimento do Software, avalie o seu nível de conhecimento: *

- ◯ Nenhum (É o primeiro contato com esse tema.)
- ◯ Baixo (Tem pouco conhecimento sobre o tema.)
- ◯ Médio (Tem bom conhecimento sobre o tema.)
- ◯ Alto (Tem amplo conhecimento sobre o tema.)

4.2. Como você define "Sustentabilidade" no processo de desenvolvimento do software? *

Your answer

4.3. Qual das opções abaixo mais se aproxima do seu entendimento sobre "Software sustentável"? *

- Software desenvolvido e usado de tal forma que deixa um impacto negativo mínimo sobre os usuários, meio ambiente, economia e sociedade em geral.
- Código de software sendo sustentável, agnóstico de propósito.
- Software cujo propósito é apoiar metas de sustentabilidade, ou seja, melhorar a sustentabilidade da humanidade em nosso planeta.
- Software cujos impactos negativos diretos e indiretos sobre a economia, sociedade, seres humanos e meio ambiente que resultam do seu desenvolvimento, implantação e uso são mínimos e / ou que tem um efeito positivo sobre o desenvolvimento sustentável.
- Software amigável ao meio ambiente que ajuda a melhorá-lo.
- Uma aplicação que produz o mínimo de desperdício possível durante o seu desenvolvimento e operação.

4.4. Dê uma nota de 1 a 5 para o quanto você considera que as empresas deveriam dar importância à questão da sustentabilidade no processo de desenvolvimento do software. Sendo que a nota 1 significa "sem importância" e a nota 5 "muito importante". *

1	2	3	4	5
○	○	○	○	○

4.5. Em relação à questão anterior, justifique sua nota. *

Your answer

4.6. Na sua opinião, o que é necessário para que um processo de desenvolvimento de software seja considerado sustentável? *

Your answer

4.7. Dê uma nota de 1 a 5 para o quanto você considera que as empresas que têm um processo de desenvolvimento de software voltado para a sustentabilidade atraem mais clientes. Sendo que a nota 1 significa "não atrai" e a nota 5 "atrai muito". *

1	2	3	4	5
○	○	○	○	○

4.8. Em relação à questão anterior, justifique sua nota.

Your answer

4.9. Com base na sua percepção profissional, avalie os tópicos, abaixo, segundo o seu nível de importância no processo de desenvolvimento do software: *

	Irrelevante	Pouco importante	Tanto faz	Importante	Muito importante
Longevidade (Habilidade do software de ter vida longa, evoluindo junto com as necessidades do usuário, a plataforma ou tecnologia implementada)	○	○	○	○	○
Adaptação à mudanças	○	○	○	○	○
Desenvolvimento dirigido à features (Características)	○	○	○	○	○
Performance	○	○	○	○	○
Esforço de desenvolvimento	○	○	○	○	○
Reusabilidade	○	○	○	○	○

Roadmap de produtos ("Mapa" que visa organizar as metas de desenvolvimento de um software)	○	○	○	○	○
Consumo de energia	○	○	○	○	○
Conscientização sobre o uso de práticas sustentáveis	○	○	○	○	○
Evolução do software	○	○	○	○	○
Qualidade do sistema	○	○	○	○	○
Ética sustentável	○	○	○	○	○
Interesse ambiental	○	○	○	○	○
Time to Market (Período de tempo entre as primeiras idéias em torno de um produto e sua eventual disponibilidade nos mercados consumidores)	○	○	○	○	○

5. Sobre o processo de desenvolvimento da empresa...

Esta seção consiste em explorar o processo de desenvolvimento do software da empresa em que o profissional trabalha e identificar a adoção de práticas sustentáveis.

5.1. A sua empresa incentiva a adoção de práticas sustentáveis no processo de desenvolvimento do software? *

- ○ Sim
- ○ Não
- ○ Não sei informar.

5.2. Se a resposta da questão anterior foi "Sim", dê uma nota de 1 a 5 para o quanto sua empresa incentiva a adoção de práticas sustentáveis no processo de desenvolvimento do software. Dê uma nota de 1 a 5, sendo que 1 significa "não incentiva" e 5 significa "incentiva muito".

1	2	3	4	5
○	○	○	○	○

5.3. Em relação à questão anterior, justifique sua nota. *

Your answer

5.4. Na sua opinião, o uso da sustentabilidade no processo de desenvolvimento de software representa para as empresas: *

- ☐ Oportunidade de novos negócios
- ☐ Nem ganhos, nem despesas
- ☐ Custos e despesas
- ☐ Não sabe avaliar
- ☐ Other:

5.5. A empresa em que você trabalha se preocupa em minimizar os impactos negativos que as atividades do processo de desenvolvimento tradicional podem causar no meio ambiente? Dê uma nota de 1 a 5, sendo que 1 significa "não se preocupa" e 5 "se preocupa sempre". *

1	2	3	4	5
○	○	○	○	○

5.6. Em relação à questão anterior, justifique sua nota. *

Your answer

5.7. A empresa em que você trabalha faz: *

- ☐ Do ponto de vista de processo, uso de algum modelo de referência para desenvolvimento de software sustentável (Ex. ISO 50000)
- ☐ Aplicação de técnicas de elicitação de requisitos que apoiem a identificação dos requisitos voltados para a sustentabilidade.
- ☐ Uso de padrões de projeto visando a redução do consumo de energia das aplicações implementadas.
- ☐ Uso de ferramenta de medição do consumo de energia durante o desenvolvimento da aplicação.
- ☐ Testes para verificação de consumo de energia da aplicação.
- ☐ Nenhum
- ☐ Other:

5.8. Cite qual(is) ferramenta(s), técnica(s) ou método(s) de medição da sustentabilidade é (são) utilizado(s) no processo de desenvolvimento do software da empresa em que você trabalha?

Your answer

5.9. Cite qual(is) padrão(ões) de projeto sustentável(is) é(são) utilizado(s) no processo de desenvolvimento do software?

Your answer

5.10. Caso sua empresa aplique noções de sustentabilidade no processo de desenvolvimento do software, quais fases do ciclo de desenvolvimento de software são cobertas? *

- [ ] Requisitos
- [ ] Arquitetura (Análise e Projeto)
- [ ] Codificação
- [ ] Teste
- [ ] Manutenção
- [ ] Nenhuma das alternativas anteriores.

6. Dificuldades encontradas

Esta seção objetiva identificar as dificuldades e benefícios esperados quanto a aplicação da sustentabilidade no processo de desenvolvimento do software.

6.1. Na sua opinião, quais são as principais barreiras que dificultam a adoção de ações/práticas de sustentabilidade no processo de desenvolvimento de software do ambiente corporativo? *

- [ ] É muito caro
- [ ] Falta de conscientização
- [ ] Não possui pessoal qualificado
- [ ] Não considera o assunto relevante
- [ ] Dificuldade em mensurar os ganhos
- [ ] Burocracia
- [ ] Não sabe avaliar
- [ ] Other:

6.2. Em qual(is) fase(s) do processo de desenvolvimento de software da empresa na qual atua, você identifica deficiências relacionadas à sustentabilidade? *

- [ ] Requisitos
- [ ] Design/Arquitetura
- [ ] Codificação
- [ ] Testes
- [ ] Manutenção
- [ ] Nenhuma das opções acima.

6.3. Em relação à questão anterior, qual(is) deficiência(s) você identifica na(s) respectiva(s) fase(s)?

Por favor, informe as deficiências por fase. Exemplo: Requisitos - Deficiência A, Arquitetura - Deficiência B e C.

Your answer

6.4. O que você acha que pode ser feito para melhorar as deficiências apontadas na questão anterior?

Por favor, informe as sugestões por fase. Exemplo: Requisitos - Proposta A, Arquitetura - Proposta B e C.

Your answer

7. A sustentabilidade como um atributo de qualidade

Esta seção tem o objetivo de obter uma percepção do entrevistado em relação à importância do uso da sustentabilidade como um atributo de qualidade em seus projetos.

7.1. Na sua opinião, a sustentabilidade deveria ser considerada um requisito não-funcional (atributo de qualidade)? *

- ○ Sim
- ○ Não

7.2. Se sua resposta foi "SIM" na questão anterior, por quê?

Your answer

7.3. A empresa em que você trabalha considera quais dos atributos, abaixo, como requisito não-funcional de software verde? *

- [ ] Sustentabilidade
- [ ] Eficiência energética
- [ ] Consumo de energia
- [ ] Desempenho de energia
- [ ] Nenhuma das opções acima.
- [ ] Other: __________

7.4. Você tem conhecimento de algum modelo de qualidade do processo ou produto para apoiar a Engenharia de Software Sustentável? *

- ○ Sim
- ○ Não

7.5. Se sua resposta foi "SIM" na questão anterior, qual(is) modelo(s)?

Your answer

Agradecimentos

A equipe de pesquisa agradece imensamente por você ter chegado até aqui. Sua contribuição certamente será muito valiosa à nossa investigação.

Informe seu e-mail caso seja do seu interesse receber uma cópia dos resultados desta pesquisa:

Your answer

Caso queira nos deixar alguma sugestão/crítica/recomendação, por favor, use o espaço abaixo:
Sinta-se à vontade para não nos informar. Caso decida nos informar, saiba que não divulgaremos.

Your answer