




Article

An Exploratory Study of Software Sustainability at Early Stages of Software Development

Hira Noman ¹, Naeem Ahmed Mahoto ¹, Sania Bhatti ¹, Hamad Ali Abosaq ^{2,*} , Mana Saleh Al Reshan ²  and Asadullah Shaikh ² 

¹ Department of Software Engineering, Mehran University of Engineering & Technology, Jamshoro 76062, Pakistan; hira.noman@faculty.muett.edu.pk (H.N.); naeem.mahoto@faculty.muett.edu.pk (N.A.M.); sania.bhatti@faculty.muett.edu.pk (S.B.)

² College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia; msalreshan@nu.edu.sa (M.S.A.R.); asshaikh@nu.edu.sa (A.S.)

* Correspondence: haabosaq@nu.edu.sa

Abstract: Sustainability incorporation within the field of Software Engineering is an emerging research area. Sustainability, from an academic perspective, has been addressed to a large extent. However, when it comes to the software industry, the topic has not received much-needed attention. Software, being designed and developed in the industry, can benefit society at large, if sustainability is taken into account by the software professionals during the software design and development process. To develop a sustainable software application, knowledge and awareness about sustainability by professional software developers is one of the key elements. This study is an attempt to examine sustainability knowledge, importance, and support from the perspective of South Asian software professionals. Additionally, this study also proposes sustainability guidelines for certain software applications and also a catalog for the identification of sustainability requirements for different software applications. The queries such as ‘What does sustainability mean to a professional software developer?’, ‘How does the software industry identify sustainability requirements?’, ‘How do software developers incorporate the sustainability parameters within software during software development?’, and many other such queries are addressed in this study. To achieve this goal, a survey was carried out among 221 industry practitioners involved in software projects in various application domains such as banking, finance, and management applications. The results pinpoint that even though sustainability is deemed important by 91% of practitioners, still there is a lack of understanding regarding sustainability incorporation in software development. A total of 48% of professionals often misunderstand “Green software” as “sustainable software”. The technical aspect of sustainability is considered most important by professionals (67%) as well as companies (77%). One of the key findings of this study is that 92% of software practitioners are not able to identify sustainability requirements for software applications. The outcomes of the study may be regarded as an initial attempt towards how sustainability is comprehended in software by the South Asian software industry.

Keywords: sustainability; software sustainability; software practitioners; software design; sustainability guidelines; sustainability catalog



Citation: Noman, H.; Mahoto, N.A.; Bhatti, S.; Abosaq, H.A.; Al Reshan, M.S.; Shaikh, A. An Exploratory Study of Software Sustainability at Early Stages of Software Development. *Sustainability* **2022**, *14*, 8596. <https://doi.org/10.3390/su14148596>

Academic Editors: Alok Mishra, Coral Calero, M^a Ángeles Moraga, Jari Porras and Deepti Mishra

Received: 2 June 2022

Accepted: 12 July 2022

Published: 14 July 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sustainability has emerged as a grand challenge for societies around the globe because of its pervasive nature [1]. It stands on five pillars, namely: Environmental, Social, Economic, Technical, and Individual [2,3]. The exponential increase in the development of software applications and human dependencies for day-to-day routine tasks through automation have brought forward sustainability as one of the most persistent apprehensions in the field of Software Engineering (SE) [4,5].

The SE research community over the last decade has made efforts to incorporate sustainability in software design and development [6]. Although these efforts are a step in the right direction, they are insufficient because there is a lack of acceptance and support from the software industry [7]. The software industry is quite passive towards sustainability incorporation [8]. One of the fundamental reasons behind the passive approach of the software industry towards sustainability could be its lack of clear understanding and availability of methods to incorporate sustainability during the course of software development. This insufficient understanding of sustainability and unavailability of detailed guidelines of its incorporation may prevent software professionals from effectively addressing sustainability during software application development. Given the extent to which software applications facilitate numerous aspects of our daily lives today, society may benefit at large from software applications when sustainability is taken into consideration during their design and development. Thus, industry professionals need concrete guidance for the effective integration of sustainability in software systems [9].

Additionally, identification of sustainability requirements for a software is also challenging for the practitioners as there are a plethora of software applications available in the market, having different features and configurations. Therefore, not all software applications can be made sustainable in a similar fashion. Different software may require different parameters for sustainability. For example, a software application that allows printing documents may provide its user notifications or recommendations to take prints in gray-scale whenever the application detects a monotone printer, thus reducing carbon emissions during printing. A location-based software application should consider fuel saving options. Thus, there is a need for further discussion on the explicit identification of sustainability requirements for different software applications [10,11]. Evidently, the current SE practices do not provide explicit support for all sustainability apprehensions as the support for quality, safety, and security is available [3]. Even though sustainability has been reckoned as a nonfunctional requirement [8,12,13], still the standard software quality models do not offer sufficient provision for sustainability requirements and apprehensions. The scientific literature suggests the need for further discussion on the understanding and identification of sustainability requirements [10,14].

Because sustainability is a global issue, it must be tackled at the same level. If we wish to effectively address sustainability in the software development process, we must first understand how the software industry in different parts of the world views sustainability. Researchers have undertaken surveys in various parts of the world, including developed countries, such as the United Kingdom and Austria, to investigate software professionals' awareness and knowledge of sustainability. However, such exploratory investigations are still needed in the developing countries, specially in South Asian countries. To the best of our knowledge, no study has been performed in the South Asian region that examined professionals' ability for identifying sustainability requirements or their understanding of sustainability. Therefore, with this viewpoint in this exploratory study, we identified the understanding of software professionals about sustainability by conducting a qualitative survey. The participants of this survey study were from the software industry with a total of 221 professionals. The geo-location of the software professionals was from the South Asia region with a major portion being Pakistani nationals. We hope to expand this study in two ways in the future: First by increasing the number of survey participants and adding additional data sources such as interviews to better understand experts' perspectives on sustainability and its requirements. Second, by extending the number of case studies in order to develop a software prototype capable of suggesting sustainability requirements or guidelines for different kinds of software applications.

The key research question addressed in this study is: How is sustainability comprehended by software professionals? The study also attempted to examine software professionals' ability in identifying sustainability requirements. Additionally, a catalog for sustainability has also been proposed in this study that can be used for the identification of sustainability requirements for different software applications. The usage of the catalog has

been exemplified using two case studies. Consequently, the proposed catalog can provide a foundation for developing a prototype that can assist professionals in the identification of sustainability requirements for different types of software applications. The survey results corroborate findings from prior studies such as [14–16] discussed in Table 1. Sustainability in the context of SE is found to be a relatively new topic. Professionals may not fully comprehend sustainability, but they recognize its significance. Experts’ frequently confuse sustainable software with Green Software. The experts’ priority is technical sustainability. According to one of the study’s distinctive findings, professionals are unable to appropriately identify sustainability requirements for software applications at hand.

Table 1. The literature Summary of existing studies about software sustainability.

Ref	Research Study	Research Description	Geo-Location
[14]	Research Type: Qualitative Research Instrument: Questionnaire Sample Category Industry Sample Size: 97	Purpose: Industry awareness about sustainable Software Engineering. Target Audience: Brazilian software practitioners. Dimension: Technical, Social, Environmental, and Economic. Sustainability Guidelines/Requirements: N/A	Brazil
[17]	Research Type: Quantitative Research Instrument: Questionnaire Sample Category Industry and Academia Sample Size: 108	Purpose: Identification of challenges in Green and sustainable software development. Target Audience: Not solely affiliated with the IT industry. Dimension: Environmental. Sustainability Guidelines/Requirements: N/A	Multiple countries including Pakistan
[18]	Research Type: Quantitative and Qualitative Research Instrument: Questionnaire and Document Analysis Sample Category Academia Sample Size: 144	Purpose: Sustainability awareness in Higher Education Institute (HEI) students. Target Audience: University students affiliated with IT (Information Technology), Computer Science, and Software Engineering. Dimension: Environmental, Social, and Economic Sustainability Guidelines/Requirements: N/A	Pakistan

Table 1. Cont.

Ref	Research Study	Research Description	Geo-Location
[19]	Research Type: Quantitative Research Instrument: Questionnaire Sample Category: General Public Sample Size: 906	Purpose: Awareness of software sustainability dimensions and their characteristics. Target Audience: Saudi Arabian Software Users. Dimension: Technical, Social, Environmental, Economic, and Political. Sustainability Guidelines/Requirements: N/A	Saudi Arabia
[20]	Research Type: Quantitative Research Instrument: Questionnaire Sample Category: General Public. Sample Size: 906	Purpose: Understanding user perspective of software sustainability. Target Audience: Citizens of Saudi Arabia. Dimension: Technical, Social, Environmental, Economic, and Political. Sustainability Guidelines/Requirements: N/A	Saudi Arabia
[21]	Research Type: Quantitative Research Instrument: Questionnaire Sample Category: General public and Organizations Sample Size: 500	Purpose: Awareness relating to sustainable development goals. Target Audience: General public and employees. Dimension: Sustainable Development Goals Sustainability Guidelines/Requirements: N/A	Pakistan
[22]	Research Type: Quantitative and qualitative Research Instrument: Questionnaire and interviews Sample Category: Industry Sample Size: 464 + 18	Purpose: Professionals' perception of energy consumption during SDLC Target Audience: Professionals (ABB, Google, IBM, and Microsoft). Dimension: Environmental Sustainability Guidelines/Requirements: N/A	Unknown
[23]	Research Type: Quantitative and Qualitative Research Instrument: Questionnaire and Interviews Sample Category: Industry Sample Size: 122	Purpose: Programmers' awareness of software energy consumption. Target Audience: Developers Dimension: Environmental Sustainability Guidelines/Requirements: N/A	Unknown

Table 1. Cont.

Ref	Research Study	Research Description	Geo-Location
[15]	Research Type: Qualitative Research Instrument: Interviews Sample Category: Industry Sample Size: 10	Purpose: Understanding software sustainability adaptation in practice. Target Audience: Software Project Team Leads (9 companies). Dimension: Environmental, Social, Economic, Technical, and Individual. Sustainability Guidelines/Requirements: N/A	Austria
[24]	Research Type: Quantitative Research Instrument: Questionnaire Sample Category: Industry Sample Size: 53	Purpose: Energy-related impact of quality to develop environmentally sustainable software. Target Audience: Software Professionals (7 separate companies). Dimension: Environmental. Sustainability Guidelines/Requirements: N/A	Turkey
[25]	Research Type: Mixed-Method Research Instrument: Questionnaire, Interview, and Contextual Inquiry Sample Category: Canteen Sector (End-users of a decision support system (DSS)) Sample Size: 60 + 18 + 1	Purpose: Sustainability requirements and positive enabling effects of software on environmental sustainability. Target Audience: Meal Planners Dimension: Environmental Sustainability Guidelines/Requirements: Sustainability Guidelines/Requirements: Elicitation and comparison between traditional and sustainability requirements in the context of positive enabling effects for a canteen DSS.	Unknown
[16]	Research Type: Qualitative Research Instrument: Questionnaire Sample Category: Industry Sample Size: 25	Purpose: Practitioners' views of sustainable practices. Target Audience: Software Engineers Dimension: Environmental, Economic, Social, and Technical Sustainability Guidelines/Requirements: N/A	Brazil

Table 1. Cont.

Ref	Research Study	Research Description	Geo-Location
[26]	Research Type: Qualitative Research Instrument: Interviews Sample Category: Industry Sample Size: 98	Purpose: Recommendations for improving sustainability from developers' view. Target Audience: Research Software Engineers Dimension: Not stated/given Sustainability Guidelines/Requirements: Developer-defined recommendations are provided for research software only.	United Kingdom

Figure 1 reports the conceptual framework adopted for this exploratory research study. Initially, the literature was reviewed in-depth to discover norms that were followed and gaps in previous studies, followed by focus group discussions with domain experts and software professionals. This contributed to the discovery of a gap in the literature. Following that, a questionnaire was designed based on the observed need and refined through group discussions. Then, before the official survey, a pilot test was undertaken to fine-tune the questionnaire. Next, the official survey was conducted, after which the results of the survey were examined, and the most important findings were noted. Survey findings confirmed the gap identified in the literature. One of the survey's noteworthy findings was the difficulty in determining sustainability requirements; to combat this, a catalog was proposed and demonstrated through two case studies.

The rest of the paper's organization is as follows. The existing studies about software sustainability are discussed in Section 2. The survey conducted to explore sustainability understanding, importance, and support among practitioners is discussed in Section 3. The key observations of this study are presented in Section 4. Furthermore, the catalog and guidelines for sustainability-related requirements are discussed in Section 5. Challenges and threats to the validity of the study are discussed in Section 6. Finally, conclusions are drawn in Section 7.

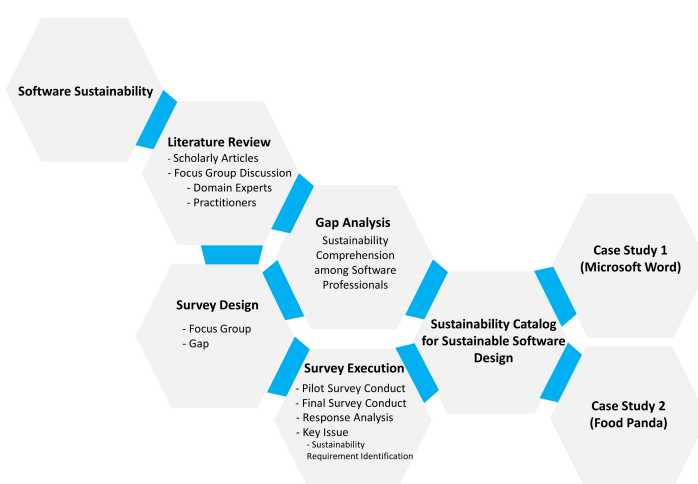


Figure 1. Conceptual framework for exploratory research study.

2. The Existing Studies

Several empirical studies have been conducted to understand and address sustainability and its related issues in software development. The existing studies available in the literature are summarized in Table 1.

The Existing Studies Summary and Gap Identification

The researchers in [14] ran a study that focused on software industry comprehension of sustainability. The study's data set was limited to only Brazil. The authors did not address the issue of sustainability requirements or guidelines as part of the study. The purpose of the industry survey conducted in [17] was to identify problems and challenges in the development of Green and sustainable software. The survey's target demographic was broad (not solely affiliated with the software industry). The survey sample replies came from several different offshore countries. The research did not include any sustainability recommendations or guidelines. A quantitative study conducted in [21] observed the public and organizational understanding of sustainable development goals. The focus of the study was not the IT industry. This study also did not include any sustainability recommendations or guidelines. The study in [25] focused on changes that occur in system requirements when they are viewed through the sustainability lens. The authors elicited system requirements from meal planners about positive enabling effects concerning environmental sustainability. The study did not provide any specific sustainability criteria; rather, it presented a comparison of a system's traditional requirements and sustainability requirements. An interview study in [26] explored the ways to improve software system's sustainability from a developer's perspective. The study did include recommendations for improving sustainability, but these were exclusively for research software. An exploratory empirical study presented in [20] focused on four scales—beliefs, intention, attitude, and perceptions—about adopting sustainable software. The study examined software sustainability from the perspective of users in the Kingdom of Saudi Arabia (KSA). It also addressed important obstacles to the development of sustainable software, such as a lack of knowledge and the difficulty in identifying sustainable software. The audience of the survey conducted as part of the study were all end-users from KSA, no sustainability guidelines were proposed in this study. Another empirical research carried out in [22] examined how professionals approached energy when they defined requirements, developed the software, tested it, and managed it. The participants of the study were from Google, IBM, Microsoft, and ABB.

Apart from empirical studies, systematic mapping studies (SMS) and systematic literature reviews (SLR) exploring sustainability in the context of software engineering have also been published. An SMS conducted by [27] on Green and sustainable SE discusses the need for better alignment between the research and practice within the SE domain. Another study presented by [10] suggests a lack of understanding of software sustainability among academia and industry. Additionally, the study conducted in [28] identifies the need of conducting research in developing countries about software sustainability.

Given the existing studies and to the best of our knowledge, currently, no qualitative study concerning sustainability comprehension and sustainability requirements has been conducted that focuses on the emerging South Asian countries, for instance, the Pakistani software industry. In addition, this study also recommends guidelines for sustainability incorporation in software during its design through a catalog. That is to say, sustainability requirements are suggested for two software applications as case studies for this research study. In addition to sustainability, there has been much research carried out in the SE domain related to prioritization techniques in software requirements [29], attack detection in software defined networks [30], and software defect prediction [31].

3. Sustainability Practitioners' Perspective

The research conducted a qualitative survey and collected the viewpoints of industry practitioners' about sustainability concerning software design and development and how it is being addressed in practice.

3.1. Study Aim and Research Objectives

The aim of this study was to identify how sustainability is perceived among software professionals, particularly in the South Asian region. This aim led to the following

three Research Objectives: (RO) 1. Investigate Knowledge. 2. Investigate Importance. 3. Investigate support of sustainability. To achieve each of the three research objectives, six research questions (RQ) were designed which were further transformed in the form of a questionnaire. Figure 2 shows the overall survey methodology and the mapping of six research questions with three research objectives.

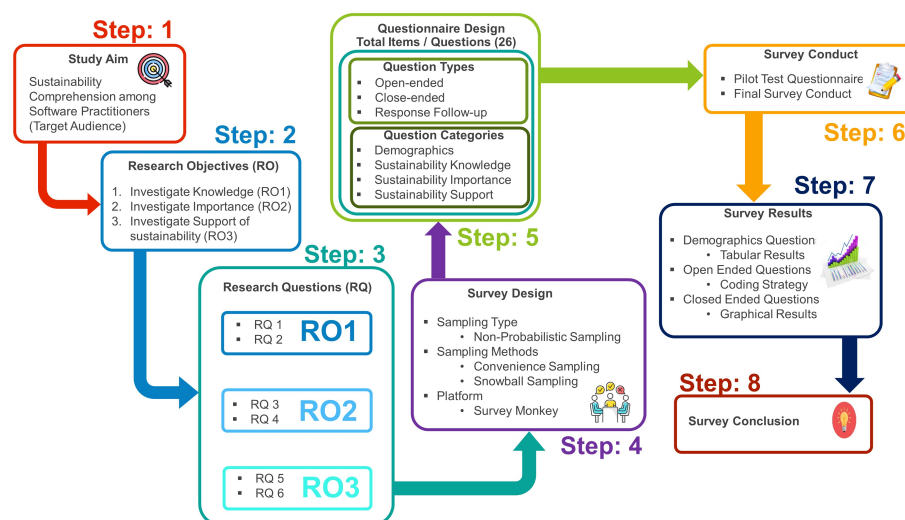


Figure 2. Survey Methodology.

3.2. Research Questions

A total of six research questions (RQ) were designed in this study, each of which was mapped according to the research objectives (RO).

3.2.1. RO 1: Investigate Knowledge

RQ 1: What does sustainability and software sustainability mean to professionals?

The question examines the software practitioners' overall knowledge regarding the concept of sustainability and software sustainability along with its dimensions.

RQ 2: Can professionals identify sustainability requirements for software applications?

The goal was to investigate whether the industry practitioners could identify sustainability requirements within a certain software application provided to them.

3.2.2. RO 2: Investigate Importance

RQ 3: How important is software sustainability to the professionals?

This RQ aimed at exploring the level of extent to which the practitioners considered software sustainability as an important aspect.

RQ 4: Is the trade-off between software cost and software sustainability justified?

The goal was to discover justifications and viewpoints on whether software professionals favor the increase in software cost concerning sustainability incorporation in the software systems.

3.2.3. RO 3: Investigate Support/Organizational Support

RQ 5: Is sustainability supported by software organizations? Which aspects of sustainability are being addressed in practice?

These questions identify whether or not the organizations provide support for software sustainability and which aspect of sustainability is considered by organizations during software development.

RQ 6: How is software sustainability supported during the design of software applications?

This research question identifies if there are any models/tools/guidelines used in practice to engineer software sustainability during the software development process.

3.3. Survey Design

This section contains details concerning the planning, execution, and reporting of the conducted survey.

3.3.1. Protocol

The survey was planned, executed, and reported using the methodology proposed in Kasunic [32] and the principles of survey research by Kitchenham and Pfleeger [33–38]. This qualitative exploratory study applied nonprobabilistic sampling. Two of the nonprobabilistic sampling methods, namely convenience and snowball sampling [37], were utilized to collect a sizable sample and to overcome the weaknesses of each type when used alone. The initial group of participants was selected based on convenience sampling. Because the goal was to learn about as many perspectives as possible, the initial set of participants was progressively expanded by adding people based on suggestions from current participants until the theoretical saturation was reached. This means that additional responses are only marginally (if at all) useful once no new information is obtained from them. Snowball sampling made it possible for us to find practitioners from various channels, having diverse positions, and ranging in seniority. The target population of this survey was software professionals working in the South Asia region of the world, the majority of the respondents belonged to Pakistan.

3.3.2. Questionnaire

For the self-administered questionnaire, we designed it to understand the industry professionals' standpoint on sustainability. We identified five information groups (categories). The categories are: participants categorization, organization categorization, research objective 1 (sustainability knowledge), research objective 2 (sustainability importance), and research objective 3 (sustainability support). Each category is described below.

1. **Participants Categorization:** The goal of this category was to investigate the participant profile, which included information such as gender, name, age, level of education, and professional experience.
2. **Organization Categorization:** This category looked into the companies' demographics including sector, type of organization, certifications, and the role played by participants inside the company.
3. **Research Objective 1 (sustainability knowledge):** The purpose of this category was to examine the respondent's understanding of sustainability along its dimensions in general as well as of sustainability in relation to software. We also examined respondents' ability to identify sustainability requirements. Moreover, how the respondent felt about adopting sustainability as a quality feature was also examined.
4. **Research Objective 2 (sustainability importance):** This category was intended to learn about how significant sustainability is in the eyes of professionals when it comes to software design and development.
5. **Research Objective 3 (sustainability support):** This category was intended to look into how the respondent's employer develops software and to determine which and to what extent sustainability aspects are considered during software development.

The questionnaire contained a combination of open-ended and closed-ended questions. Though the majority of the questions were close-ended, however, there were some open questions that inquired about sustainability support, sustainability requirements, certification types, and sustainability models/techniques/tools used in current software development practices. In addition, some follow-up questions are part of the questionnaire, which were prompted to the participants based on their responses to previously answered questions. For instance, if the respondent signaled that they could identify sustainability requirements for a software application, then they were asked to state such

requirements. The questionnaire was built using SurveyMonkey (Survey Questionnaire Link: <https://www.surveymonkey.com/r/D9RNB3L>, accessed on 7 January 2022). The survey remained active from 15th August 2021 to 14th September 2021 in collecting responses. The URL to access it was sent through email, social media, and WhatsApp to the participants.

The Inclusion-Exclusion Criteria: A total of 221 responses were collected. The completion rate of the questionnaire was 83% because some questions involved skip logic and some unfinished responses were collected. To maintain the quality of the survey, the received responses were validated. The validation process was scrutinized, a final of 201 responses were analyzed, and 10 responses were discarded due to nonconformity with the quality criteria set by the authors.

Quality Criteria (Inclusion):

- Questionnaire must be completely filled.
- The respondent must be working as a software professional in the South Asian region.

Quality Criteria (Exclusion):

- Incomplete questionnaire.
- For instance, individuals working as human resource managers and those working in the educational field as research assistants, students, and teachers.

The received responses reflect a similar success rate in response collection for this study as has been observed in the study [39]. Table 2 presents the key statistics of the conducted survey.

Table 2. Survey responses' statistics.

Total Responses	Incomplete Responses	Completed Responses	Valid Responses	Average Response Time in Minutes	Responses Completion Rate
221	10	211	201	15	83%

To extract unbiased opinions, the participants were not given any prior knowledge of sustainability or software sustainability concepts. Furthermore, to avoid evaluation stress, participants were notified regarding the purpose of the study and were assured about the privacy and confidentiality of the data.

3.3.3. Survey Response Analysis

The strategy used to analyze the responses to the question is discussed below.

1. For close-ended questions that allowed multiple responses, the aggregate of percentages was above 100%;
2. For closed questions that accepted a single outcome, the percentages were calculated based on the response chosen;
3. Coding strategy was applied to open questions that inquired about sustainability measures in practice, software tools supporting sustainability, and organization certification types. Two of the authors coded the replies and compared their results to come up with a single coding scheme;
4. For the open-ended question that explored sustainability requirements regarding particular software applications, we included extracts from the qualitative replies. Each extract is tagged with a unique identifier to distinguish it from the rest. For instance, [*1] signifies the first extract.

3.4. Results of Survey

The received responses for the considered study are described in the following section.

3.4.1. Participants' Demographics

The participants' responses were examined based on age, level of education, and professional experience to draw their profiles. Table 3 gives an overview of the demographics of the survey participants. Participants in the study ranged in age from 18 to 44, with men making up the majority. The survey's target audience's work history was also noted. The work experience of the participants shown in Table 3 represents their professional experience in Pakistan. The professional experience of participants shows that they worked in several different countries. In total, 99 professionals who had solely worked in Pakistan participated in the study, compared with 102 individuals who had both domestic and international job experience. For instance, there were 40 professionals who had fewer than a year experience of working abroad, while 36 professionals had one to three years of experience. A total of 13 participants had foreign experience of between four and six years and 7 had seven to ten years of experience. Finally, 6 professionals had more than 10 years of international work experience.

Table 3. Independent variables of software professionals (participants).

S. No	Parameter	Classification	Sample Size
1	Age in years	18 to 24	39
		25 to 34	137
		35 to 44	25
2	Gender	Male	157
		Female	44
3	Experience in years	Less than 1	23
		1 to 3	48
		4 to 6	64
		7 to 10	45
		Above 10	21
4	Education	Graduate	149
		Postgraduate	52

Figure 3 reports the respondents' roles (job responsibilities) in the organization. It is observed in Figure 3 that 7.46% of the participants work as project managers, 4.47% are employed as software designers, 59.20% as a software developers, 10.94% as consultants, 1.49% as a system analysts, 0.99% as database administrators, 0.99% as business analysts, and 1.99% as requirement engineers. The remaining 14.92% of respondents specified their job title as one of the following: requirement engineer, trainee software engineer, senior QA engineer, web designer, solution architect, graphics designer, software architect, DevOps engineer, software engineer, web developer, technical lead, and user interface developer.

3.4.2. Organizations' Demographics

The organization type was also analyzed to obtain in-depth knowledge about the participants in this qualitative survey. The type of organizations to which the participants belonged are reported in Table 4. It is observed that the majority of the survey participants were employed in purely software-related companies, while other participants were working in companies operating in domains such as banking, telecommunication, and e-commerce. In terms of official software-related certifications, 37.81% of the population reported that the company does not have any such certification. A total of 45.7% were not sure if their employer company had some certification policy. However, 16.4% responded that their company does have some of the following certifications: ISO 9001:2015, ISO/IEC 27001:2013, CMMI-DEV V1.3 Level 3, PCI DSS, and CMMI level 3 and level 5.

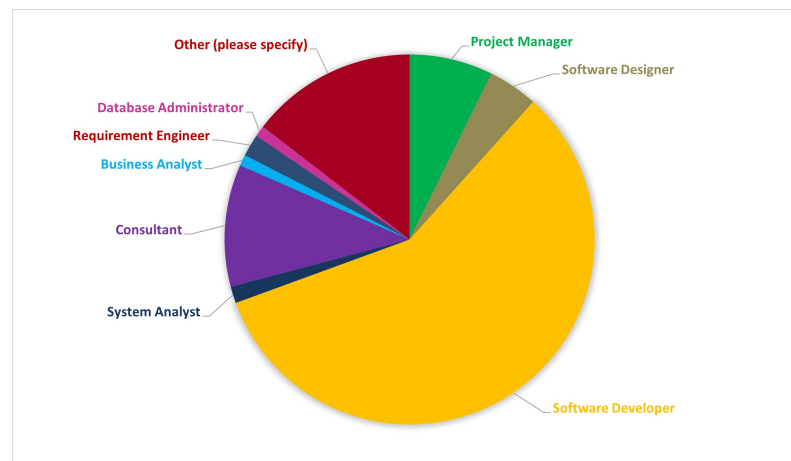


Figure 3. Roles of Software Professionals (Participants).

Table 4. Independent variables (organizations’).

S. No	Parameter	Classification	Sample Size
1	Organization Sector	Public	30
		Private	171
2	Organization Type	Startup	24
		Consulting Company	21
		Software House	131
		Educational Institution	5
		Others	20
3	Certifications	None	76
		Not Known	92
		Certified	33

3.4.3. Research Questions Responses

The collected responses according to the research questions (RQ) laid at the beginning of the survey mapped with research objectives (RO) are explained in the subsequent sections.

RO 1: Investigate Knowledge

RQ1: What does sustainability and software sustainability mean to professionals?

The objective of this question was to investigate respondents’ awareness and level of understanding of sustainability in general and in the context of software engineering. Initially, to examine the respondents’ awareness, we simply asked them if they were familiar with the term ‘sustainability’. Results show that 81% were acquainted with it while 19% never heard of the term sustainability. Secondly, they were queried regarding their awareness of sustainability dimensions. Respondents were allowed to select multiple options among the five (Environmental, Economic, Social, Technical, and Individual) which they considered to be sustainability dimensions. Responses to this question reported that 63% were aware of the Technical dimension, followed by 61% Environmental, 58% Economic, 42% Social, and 17% Individual.

Respondents were also inquired to self-examine their knowledge regarding software sustainability in terms of software design and development. Figure 4 shows that 6% claimed high knowledge of software sustainability, 29% considered their knowledge at a medium level, and 48% assessed themselves as having low knowledge, while 16% considered that they had no knowledge of it.

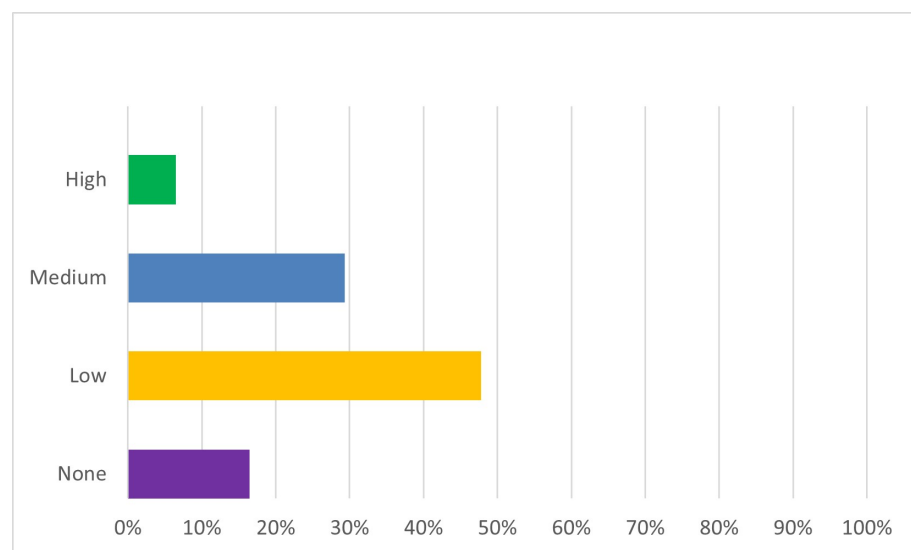


Figure 4. Participants' self-examination of sustainability (N = 201).

Additionally, a survey questionnaire queried about the comprehension of sustainable and Green software. A total of 48% of respondents considered Green and sustainable software to be the same. However, 24% were opposed to this consideration and 28% could not comprehend whether being Green and sustainable is the same or not.

Furthermore, the respondents were queried about whether they consider sustainability a quality attribute as a nonfunctional requirement. It was observed that 72% agreed to sustainability being a quality attribute (nonfunctional requirement) in software systems, and 14% were against this opinion. The remaining 14% could not decide whether or not it should be considered a nonfunctional requirement.

RQ 2: Can professionals identify sustainability requirements for software applications?

Aligned with the previous RQ, the goal of this question was to discover whether or not software practitioners can identify sustainability requirements in software applications. Therefore, respondents were questioned initially if they could identify the sustainability requirements. Consequently, those who responded positively were asked to jot down sustainability requirements for Microsoft Word. In total, 48% refuted that they could not identify sustainability requirements, and 44% were dubious in this regard. The remaining 8% replied positively. Figure 5 reports the responses collected.

As a follow-up question, 8% and 44% were requested to set down the sustainability requirements for Microsoft Word. However, the response was quite discouraging as only 1% came up with the answer. The excerpts from their answers are stated as follows.

[*1] "CLOUD STORAGE: Microsoft Word offers you to store your files on One Drive, this is a very important aspect in terms of sustainability of the software because that is the future of how we will be creating, reading, editing, and saving the documents. PORTABILITY: Microsoft Word offers you to export the document to other formats like PDF." [*2] "Optimize builds for speed—Optimize unit tests and integration tests for speed—Enforce policies to cycle down development machines to power saving state when idle—Get some percentage of energy iced from renewable energy to offset environmental pact—Encourage online downloads for distribution—Continue supporting older hardware as long as possible to keep those machines usable and prevent those from ending up in landfills". According to [*1], one of the pioneering advancements in the direction of sustainability is cloud storage. Definitely, cloud storage capability paves the way for new methods of storing and retrieving data. At the same time, software that supports cloud functionalities consumes more energy than its traditional counterparts. It is implied from the excerpt that there's a misunderstanding about what 'sustainable' implies in software and what impact cloud support has on software's energy consumption. Conferring to [*2], the emphasis is on

energy-aware practices that should be followed during software development (coding) techniques. This sample confirms that practitioners are more interested in the Green side of sustainability, implying that they regard Green software to be sustainable software. However, sustainability entails more than just being environmentally friendly.

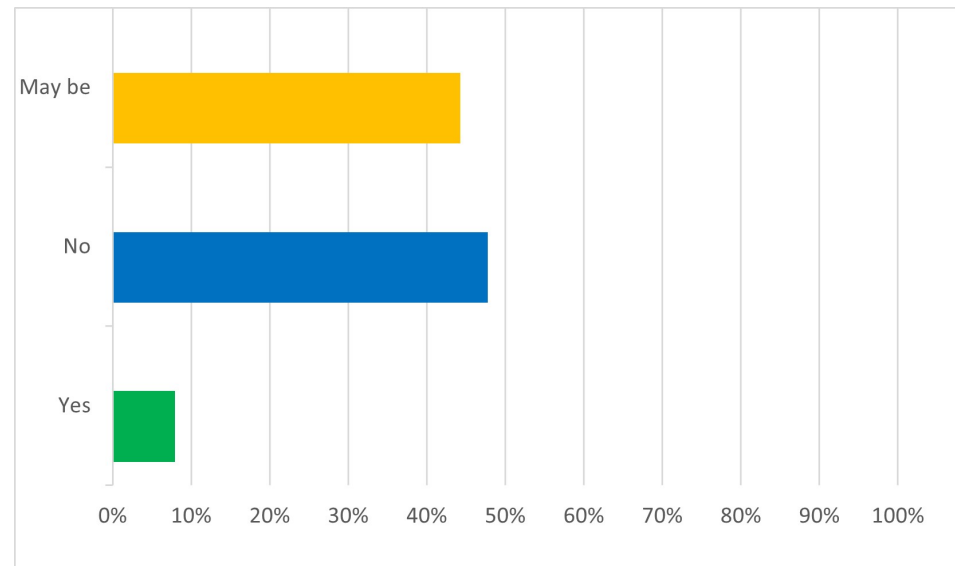


Figure 5. Participants' identification of sustainability requirements (N = 201).

RO 2: Investigate Importance

RQ 3: How important is software sustainability to professionals?

The goal was to know how significant sustainability was to software practitioners. A total of 91% agreed that sustainability should be taken into account during software production, 4% ruled against this view, while 5% could not decide. Furthermore, they were asked to prioritize sustainability dimensions based on their importance with respect to the design and development of software systems. The feedback obtained (multiple selections) is shown in Figure 6.

RQ 4: Is the trade-off between software cost and software sustainability justified?

Given RQ4, respondents were queried if the trade-off between software cost and sustainability support was justified in their opinion. A total of 83% responded in support, 12% answered against it, and the remaining 5% could not evaluate it.

RO 3: Investigate Support

RQ 5: Is sustainability supported by software organizations and which aspects of sustainability are being addressed in practice?

To investigate organizational support for sustainability, survey participants were initially queried if their employing organization provided explicit support for sustainability considerations during the software development process. A total of 51.58% of participants responded as 'Yes', 18.42% negated it, and 30% were doubtful. The respondents who replied 'Yes' were further queried regarding the dimensions based on priority by their employing organization in practice as shown in Figure 7. That is, which aspect of sustainability is provided support for by the organizations during software production. Participants were permitted to choose multiple dimensions in this question due to which the sum of percentages could exceed 100%.

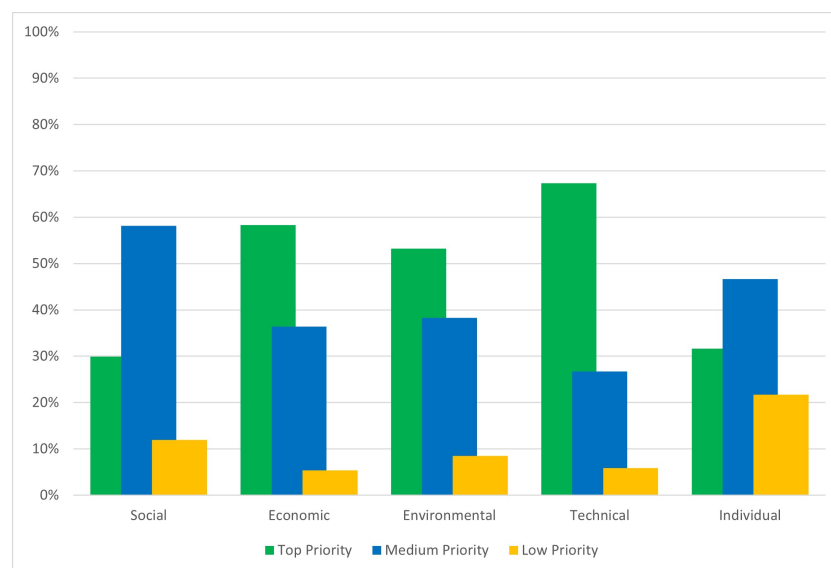


Figure 6. Participants' viewpoint about sustainability importance (N = 192).

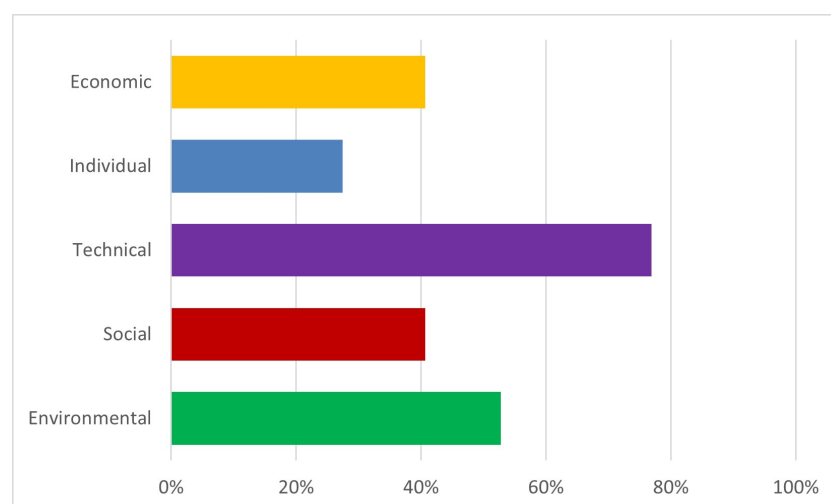


Figure 7. Participants' organizational level for sustainability dimensions (N = 91) (N = 201).

RQ 6: How is software sustainability supported during the design of software applications?

To explore how software sustainability is supported in the software industry, the professionals were queried about the tools/methods/techniques available in their respective organizations to incorporate sustainability. A total of 59 participants specified tools/methods/techniques used in their respective organizations. Some professionals responded their companies use LTS (long-term Support) for extending maintainability and promoting the dependability of the software. One respondent indicated that his/her company utilizes GRI (Global Reporting Initiative) Sustainability Reporting Guidelines to understand and report the impacts of software development on people, the environment, and the economy. Some respondents stated the use of agile methodologies such as SAFe (The Scaled Agile Framework) to develop and deliver software systems in the shortest and most sustainable lead time. Others asserted that they employed the SPRING framework for developing software applications in a secure and fast-paced manner. While some respondents indicated that their companies do contribute towards sustainability through cost management and encouraging remote-based work, they did not explicitly state the tools/methods/techniques being used.

Moreover, survey participants were further questioned if they were aware of any software capable of proposing sustainability requirements for different kinds of software applications. In all, 30% stated unavailability of such software, and 18% responded in favor of software availability. The remaining 52% were doubtful if there was some software system available that could recommend sustainability requirements. Additionally, we asked the respondents who were positive about software availability to name such software. This was an open-ended question and responses reported were: JIRA, Benchmarks Software (such as 3D mark, AIDA64 Extreme, Everest), Quentic, Cisco CEMS software, Cority (Integrated Sustainability Software), Confluence, Clubhouse, D365 ERP Solution, and IntelliJ IDEA. The responses indicated that practitioners relate CRM applications, integrated software management solutions, and project management and collaboration tools with sustainability. However, the results also point to the fact that practitioners are unaware of specific tools that support sustainability in software systems. The participants were probed if it would be beneficial to have a software system that suggests sustainability requirements for different kinds of software applications. A total of 61% voted in favor, 36% were doubtful, and the remaining 4% were against.

4. Survey Observations

The following are the research implications that surfaced from this exploratory study.

- **Sustainable SE field is still in its immaturity.**
The Sustainable SE field is still emerging. The software industry continues to see sustainability as a secondary concern. Although the practitioner community is aware of the concept of sustainability and its dimensions in general, they lack the understanding of sustainability in relation to software. Therefore, the software industry needs to be familiarized with it so that software designers and developers take sustainability into account during the software design and development process.
- **Sustainability is important.**
Even though professionals possess a limited understanding of sustainability in the context of software design and development, they still consider it important and beneficial.
- **Sustainability is not just Green.**
Software practitioners misunderstand sustainable software with Green software. Being Green is one aspect of being sustainable but sustainability is beyond Green. Practitioners need to consider the impacts of software systems on all sustainability dimensions rather than just focusing on environmental effects.
- **Technical sustainability takes priority.**
The concept of sustainability stands on five pillars, namely: Environmental, Social, Economic, Technical, and Individual. All of these dimensions are equally significant, but in industry the technical aspect of sustainability is given the most weight.
- **Sustainability requirements are challenging.**
Because professionals lack the understanding and expertise to integrate sustainability in relation to the software, they are unable to accurately identify the sustainability requirements for software systems.
- **Trade-off between sustainability and software cost is justified.**
Practitioners believe that increasing software costs is justified in order to incorporate sustainability into software systems.
- **Sustainability incorporation tools would be beneficial.**
The implementation of sustainability is not generally known among software professionals. They link traditional software development frameworks, processes, and tools to sustainability. However, the majority of experts agree that it would be extremely beneficial if there were software tools that might help with sustainability incorporation, particularly sustainability requirement identification.

5. Sustainability Incorporation during Software Design

The SE research community has long recognized the significance of sustainability and has put great attention into including it in the software development life cycle. The industry, on the other hand, lacks both an understanding and awareness of sustainability, as well as the ability to put it into practice. Consequently, sustainability is still immature in the software industry. There are several other factors behind this immaturity of sustainability in the software industry. Firstly, academic initiatives are not efficiently transferred to the industry. Secondly, industry professionals focus primarily on software release to the market, thus lacking an effective grasp of and identification of requirements related to software system sustainability [40]. Thirdly, the research community has offered frameworks, catalogs, or methodologies for integrating sustainability into SE, but most of these are at a high degree of abstraction or limited in scope.

Another stumbling block to effective adoption is the lack of practical guidelines for incorporating sustainability into software design and development [41]. As a result, it continues to lead sustainability down an ambiguous path. Only if practitioners are aware and understand how sustainability is injected into software systems during design and development can they engineer sustainability effectively.

The study in [42] presented a design manifesto that contains a set of nine principles for the sustainable design of software systems. Based on the Karlskrona design manifesto, the research in [9] offered a sustainability catalog and framework for software system design. The study exemplified the use of the catalog through four case studies and suggested sustainability guidelines for software systems under consideration. The catalog in the study was split into two sections. The first part concentrated on the Karlskrona design manifesto based derivation and application of sustainability goals, while the second part focused on the direct, indirect, and systemic effects of software on the five dimensions of sustainability. By definition, '[Green and Sustainable Software] is software, whose direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development' [43]. Therefore, designing sustainable software necessitates taking into account the direct, indirect, and systemic implications that software has on all the sustainability dimensions. This implies that software effects can be used to effectively drive and advance sustainability goals.

Another research study [44] conducted a case study examination of web-based personal health record systems to look at environmental sustainability recommendations related to energy efficiency for the development of software systems. The focus of this research was solely on the environment, with sustainability recommendations mostly centered on the power consumption of health record systems. The research in [45] describes a study that resulted in a set of software architecture principles intended to help with sustainability-driven design and monitoring. The framework given focuses on the aviation industry in particular. Because there are so many software applications in the market with different functionality and features, and since software applications serve different application domains and hence have varied architectures, not all of them can be made sustainable in the same way. Therefore, more research is needed to propose goals and guidelines or requirements for making sustainable software.

During the survey, when the practitioners were required to jot down sustainability-related requirements for Microsoft Word, the majority of them could not identify the requirements. The respondents' responses thus confirm the fact that sustainability requirement identification is a challenge for software professionals. This study analyzed the participant's responses as well as investigated the scientific literature. The study also proposed a catalog for incorporation of sustainability within software during their design.

5.1. Catalog for Sustainability Guidelines

This section describes the sustainability catalog for sustainable software design (SC-SSD). This catalog can be used to incorporate sustainability into the design of software sys-

tems. SCSSD will assist software practitioners including requirement engineers, designers, and developers in determining how software systems can be designed with sustainability at the forefront. Figure 8 presents SCSSD and the following is the description of how SCSSD works:

1. Select any software system that needs to be analyzed with respect to sustainability.
2. Analyze selected software system based on each of the five sustainability dimensions.
3. For each dimension, identify the direct, indirect, and systemic effects of the software.
4. Based on the identified effects, drive sustainability goals for their effective mitigation.
5. Next, identify the available software features that can be improved to achieve the derived goals. If none of the available features can be used, then recommend new features for the achievement of the driven goals.
6. Provide guidelines for the effective application of the sustainability goals using the software features.

The functioning of SCSSD is exemplified through two case studies: Microsoft Word and Foodpanda. Table 5 reports sustainability guidelines for Microsoft Word and Table 6 catalogs the guidelines for Foodpanda, which is a location-based food delivery service operating in Pakistan.

- **[MS WORD REQ-01]** Sustainability requirement 01 recommends launching the application in different energy modes. It may introduce three modes: (1) economic (energy-friendly mode), (2) standard, and (3) advanced mode. The economic mode allows only basic features and removes all additional software features that are costly in terms of energy consumption. For example: disable features such as auto save, spell check, and design. The standard mode should allow only some of the energy costly features, while the advanced mode has full flash word features. A user may switch between these modes;
- **[MS WORD REQ-02]** The application shall show the print preview of documents based on the current amount of ink in the toner. The user shall decide whether to take a print or not under the current ink conditions. This feature may reduce the amount of reprinting;
- **[MS WORD REQ-03]** Educate the users regarding resource savings such as toner ink and saving over time due to avoidance of reprinting documents.

Table 5. Sustainability requirements of Microsoft Word.

S. No	Proposed Sustainability Goal	Goal-Feature Mapping	Sustainability Guideline	Sustainability Dimension
1	[Ms. WORD REQ-01] Energy Saving	Energy Profiles/modes	Introduce energy profiles in Microsoft Office that allow the software to be run in an energy-saving mode.	Environmental
2	[Ms. WORD REQ-02] True Print	Printing	Show the user the actual print preview of documents based on toner conditions.	Environmental
3	[Ms. WORD REQ-03] Educate User	Notification	Let the users know how much ink they saved over time.	Social

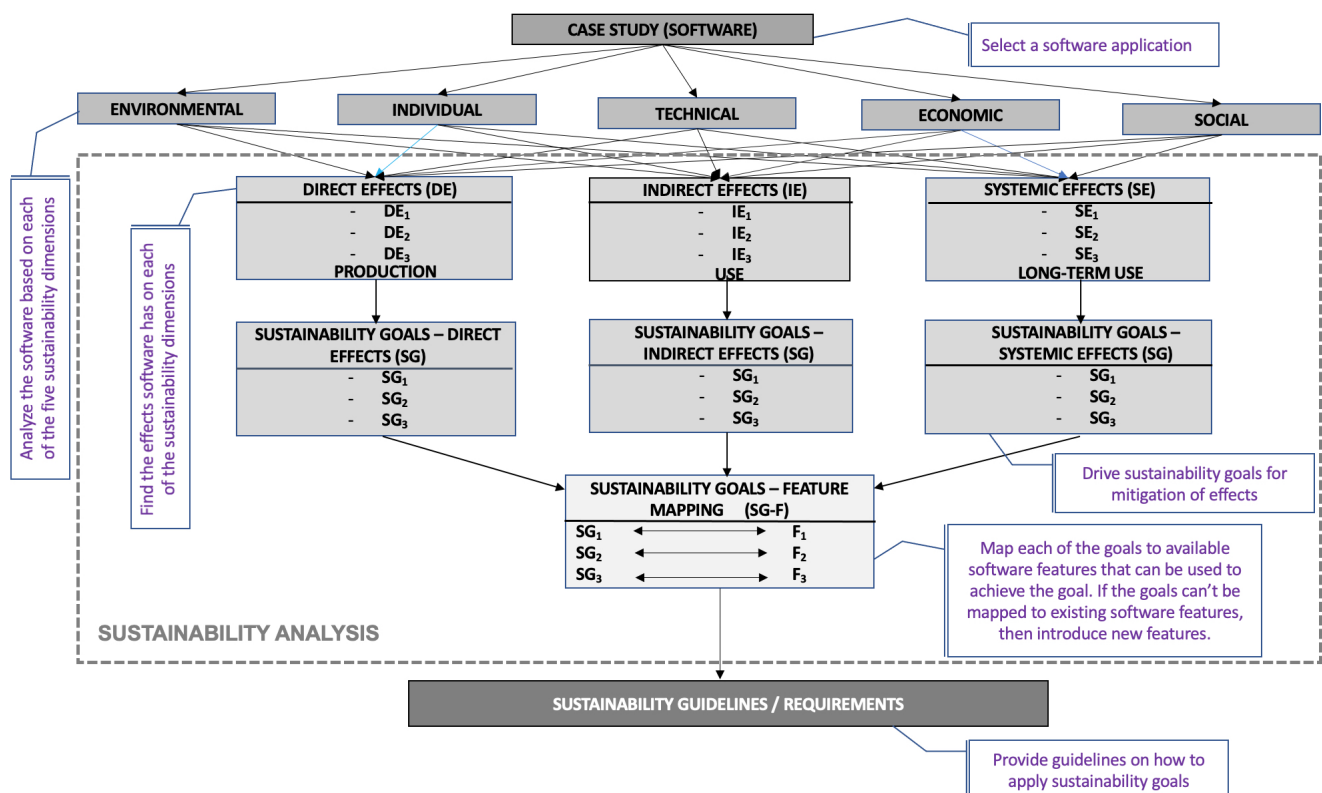


Figure 8. Sustainability catalog for sustainable software design.

5.2. Sustainability Guidelines for Design of Software Systems

This section summarizes the sustainability guidelines derived from the case studies mentioned in the previous section.

1. Just as there is a provision of energy profiles in hardware systems, this consideration should also be taken into account while designing software systems. Support for energy profiles would allow users to utilize the software based on their work needs i.e., the software can be launched in basic or energy-efficient mode with only the required features, thus lowering the hardware's energy requirement.
2. Applications supporting printing features should also contain features that encourage the saving of printing resources (i.e., paper and toner).
3. Software applications can be exploited from the perspective of imparting sustainable thinking in their users. Applications can be used to change user mindsets, and to make it happen applications should provide feedback to users based on their choices. For instance, if a user's choice saved a computing resource, the applications should provide feedback in the form of notifications.
4. Location-based applications should provide incentives for fuel savings, packaging reduction, and so on.

Table 6. Sustainability Requirements of FoodPanda.

S. No	Proposed Sustainability Goal	Goal-Feature Mapping	Sustainability Guideline	Sustainability Dimension
1	[FOODPANDA REQ-01] Fuel Saving	Notification	During application usage, notify the user if someone else from the same vicinity has placed an order on the application. If the user opts for delivery from the same restaurant the delivery charges can be split in half.	Environmental
2	[FOODPANDA REQ-02]	Cart	Encourage restaurants to use the least amount of food packaging as possible. Allow users to opt for tiffin services instead of disposable packaging.	Environmental
3	[FOODPANDA REQ-03] Reduce use of plastic	Cart	Give users the choice of skipping delivery of disposable cutlery with their placed orders.	Environmental

6. Threats to Validity

This section discusses the possible threats to validating the findings of the study. This study is subjected to the following validity threats [46].

6.1. Construct Validity

Construct validity focuses on the theoretical constructs used being interpreted and measured correctly. To tone down this threat, a presurvey pilot study was conducted. Firstly, a focus group was formed which comprised of five members from the target population and two domain experts in the software engineering discipline. The group was instructed to review the survey instrument in terms of clarity and completeness. The questionnaire adopted in this study was improved before its distribution based on the suggestions given by the focus group. For instance, during the pilot study, it was found that the completion time of the survey was extensive. To mitigate the time duration, the number of items was reduced from 33 to 27 to keep the survey within the limited time duration of 15 min (generally). Some of the questions were re-written to compensate for the removed items to avoid any negative impact on the richness of the information collected through the survey.

6.2. Internal Validity

Internal validity examines if the results are truly based on the data. The sample selected for the survey may have resulted in an internal shortcoming. As snowballing is partly used to gather the sample for the survey, a potential limitation of this strategy is community bias. To tackle this internal shortcoming, an initial group of participants was farmed, keeping it as diverse as possible. It is understood that the number of companies and the software professionals surveyed might not sufficiently represent the entire population of software practitioners and companies. Nevertheless, the targeted audience has been selected as professionals working in the South Asia region of the world. It is believed that due to diverse samples, internal community bias is dealt with.

6.3. External Validity

External Validity focuses on the extent to which the results can be generalized. The participants of this survey may not adequately represent the entire practitioners' community. Nevertheless, to increase the transferability of results, the respondents were selected from

companies belonging to diverse domains. Therefore, it is believed that the responses analyzed offered a rich source of qualitative data that revealed valuable insights.

6.4. Reliability

Reliability centers on the reproducibility of identical outcomes. Even though the grounded theory presents a meticulous process for data collection and analysis, generally qualitative studies are subject to researcher bias. It is possible that other researchers might interpret the same data differently and make different deductions after an examination. However, it is believed that the primary insights would still uphold. Additionally, the deduction made from the outcomes of RQ2 could be a risk to the validity of this study. To mitigate this, the responses were carefully analyzed and discussed thoroughly in a joint venture of the four-eyes principle [47]. The disagreements were reviewed again until a consensus was reached.

7. Conclusions

Much research efforts are made by the SE community towards the realization of sustainability importance and its incorporation within the software engineering discipline. However, the software industry still lacks an understanding of sustainability, which eventually leads to a lack of implementation in the software systems during development.

This study surveyed software industry professionals within the South Asian region. Most of the participants were Pakistani nationals. The study aimed to collect evidence about sustainability among software professionals: How software professionals perceive sustainability and identify sustainability requirements. It has been found that software practitioners require clear and complete awareness about incorporating sustainability within software development. Practitioners also need to become familiar with identifying requirements related to sustainability. Additionally, the current SE practices do not provide explicit support for sustainability. Therefore, for practitioners to effectively address sustainability-related concerns during software development, there is a need to provide explicit guidelines for its incorporation. The study presented a catalog and sustainability guidelines for two case studies (1. MS Word and 2. Foodpanda). This work will be extended in the future to impart sustainability guidelines for more software. The case studies will then be used to develop a prototype software capable of suggesting sustainability requirements for different kinds of software applications. On the basis of this research, the following recommendations for the future are offered:

1. More studies should detail practical guidelines on making software systems sustainable during their design and development.
2. Sustainability is not explicitly supported by the present SE procedures. Therefore, it is necessary to establish precise standards for its implementation.
3. Future research should attempt to create software prototypes that could suggest sustainability requirements for different types of software systems.

Author Contributions: Conceptualization, methodology, H.N., N.A.M. and S.B.; formal analysis, H.A.A., M.S.A.R. and A.S.; software, validation, writing—original draft, H.N., N.A.M. and S.B.; writing—review and editing, data curation H.A.A., M.S.A.R., S.B. and A.S.; supervision, funding acquisition, N.A.M. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Collaboration Funding program grant code (NU/RC/SERC/11/7).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Mahaux, M.; Heymans, P.; Saval, G. Discovering sustainability requirements: An experience report. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 19–33.
2. Goodland, R. Sustainability: Human, social, economic and environmental. *Encycl. Glob. Environ. Chang.* **2002**, *5*, 174–196.
3. Penzenstadler, B.; Raturi, A.; Richardson, D.; Tomlinson, B. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. *IEEE Softw.* **2014**, *31*, 40–47. [\[CrossRef\]](#)
4. Penzenstadler, B.; Bauer, V.; Calero, C.; Franch, X. Sustainability in Software Engineering: A Systematic Literature Review. In Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), Ciudad Real, Spain, 14–15 May 2012.
5. Silveira, C.; Reis, L. Sustainability in Information and Communication Technologies. In *Research Anthology on Measuring and Achieving Sustainable Development Goals*; IGI Global: Hershey, PA, USA, 2022; pp. 771–792.
6. Penzenstadler, B.; Duboc, L. Software Engineering for Sustainability. Ph.D. Thesis, Technische Universität München, Munich, Germany, 2015.
7. Riemenschneider, C.K.; Hardgrave, B.C.; Davis, F.D. Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Trans. Softw. Eng.* **2002**, *28*, 1135–1145. [\[CrossRef\]](#)
8. Chitchyan, R.; Becker, C.; Betz, S.; Duboc, L.; Penzenstadler, B.; Seyff, N.; Venters, C.C. Sustainability design in requirements engineering: State of practice. In Proceedings of the 38th International Conference on Software Engineering Companion, Austin, TX, USA, 14–22 May 2016; pp. 533–542.
9. Oyediji, S.; Seffah, A.; Penzenstadler, B. A catalogue supporting software sustainability design. *Sustainability* **2018**, *10*, 2296. [\[CrossRef\]](#)
10. Venters, C.C.; Seyff, N.; Becker, C.; Betz, S.; Chitchyan, R.; Duboc, L.; McIntyre, D.; Penzenstadler, B. Characterising sustainability requirements: A new species red herring or just an odd fish? In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS), Buenos Aires, Argentina, 20–28 May 2017; IEEE: Hoboken, NJ, USA, 2017; pp. 3–12.
11. Oyediji, A.S. Early Investigation towards Defining and Measuring Sustainability as a Quality Attribute in Software Systems. Master's Thesis, Lappeenranta University of Technology, Lappeenranta, Finland, 2016.
12. Calero, C.; Bertoa, M.F.; Moraga, M.Á. A systematic literature review for software sustainability measures. In Proceedings of the 2013 2nd International Workshop on Green and Sustainable Software (GREENS), San Francisco, CA, USA, 20 May 2013; IEEE: Hoboken, NJ, USA, 2013; pp. 46–53.
13. Venters, C.; Lau, L.; Griffiths, M.; Holmes, V.; Ward, R.; Jay, C.; Dibsedale, C.; Xu, J. The blind men and the elephant: Towards an empirical evaluation framework for software sustainability. *J. Open Res. Softw.* **2014**, *2*, e8. [\[CrossRef\]](#)
14. Karita, L.; Mourão, B.C.; Martins, L.A.; Soares, L.R.; Machado, I. Software industry awareness on sustainable software engineering: A Brazilian perspective. *J. Softw. Eng. Res. Dev.* **2021**, *9*, 2:1–2:15. [\[CrossRef\]](#)
15. Groher, I.; Weinreich, R. An interview study on sustainability concerns in software development projects. In Proceedings of the 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, 30 August–1 September; IEEE: Hoboken, NJ, USA, 2017; pp. 350–358.
16. Karita, L.; Mourão, B.C.; Machado, I. Software industry awareness on green and sustainable software engineering: A state-of-the-practice survey. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering, Salvador, Brazil, 23–27 September 2019; pp. 501–510.
17. Salam, M.; Khan, S.U. Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey. *J. Softw. Evol. Process* **2018**, *30*, e1939. [\[CrossRef\]](#)
18. Malik, M.N.; Khan, H.H.; Chofreh, A.G.; Goni, F.A.; Klemesš, J.J.; Alotaibi, Y. Investigating students' sustainability awareness and the curriculum of technology education in Pakistan. *Sustainability* **2019**, *11*, 2651. [\[CrossRef\]](#)
19. Aljarallah, S.; Lock, R. An exploratory study of software sustainability dimensions and characteristics: End user perspectives in the Kingdom of Saudi Arabia (KSA). In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Oulu, Finland, 11–12 October 2018; pp. 1–10.
20. Aljarallah, S.; Lock, R. Software sustainability from a user perspective a case study of a developing country (Kingdom of Saudi Arabia). In Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 16–17 August 2018; IEEE: Hoboken, NJ, USA, 2018; pp. 1–6.
21. Javeed, A.; Khan, M.Y.; Rehman, M.; Khurshid, A. Tracking sustainable development goals—A case study of Pakistan. *J. Cult. Herit. Manag. Sustain. Dev.* **2021**. [\[CrossRef\]](#)
22. 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 Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), Austin, TX, USA, 14–22 May 2016; IEEE: Hoboken, NJ, USA, 2016; pp. 237–248.
23. Pang, C.; Hindle, A.; Adams, B.; Hassan, A.E. What do programmers know about software energy consumption? *IEEE Softw.* **2015**, *33*, 83–89. [\[CrossRef\]](#)
24. Koçak, S.A.; Alptekin, G.I.; Bener, A.B. Integrating Environmental Sustainability in Software Product Quality. RE4SuSy@ RE. 2015; pp. 17–24. Available online: <http://ceur-ws.org/Vol-1416/Session1Paper3.pdf> (accessed on 15 March 2022).

25. Huber, M.Z.; Hilty, L.M.; Glinz, M.; Penzenstadler, B. Uncovering sustainability requirements: An exploratory case study in canteens. In *CEUR Workshop Proceedings*; Number 1416; CEUR: Kootenai County, ID, USA, 2015; pp. 35–44.
26. de Souza, M.R.; Haines, R.; Jay, C. Defining sustainability through developers' eyes: Recommendations from an interview study. In *Proceedings of the 2nd Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2)*, New Orleans, LA, USA, 16 November 2014.
27. Mourão, B.C.; Karita, L.; do Carmo Machado, I. Green and sustainable software engineering—a systematic mapping study. In *Proceedings of the 17th Brazilian Symposium on Software Quality*, Curitiba, Brazil, 17–19 October 2018; pp. 121–130.
28. Mohankumar, M.; Kumar, D.M.A. Empirical study on green and sustainable software engineering. *Adv. Softw. Eng. Syst.* 2015. Available online: https://www.researchgate.net/profile/Dr-M-Anand-Kumar/publication/274666881_An_Empirical_Study_on_Green_and_Sustainable_Software_Engineering/links/593a9e67a6fdcea3b65a9e0/An-Empirical-Study-on-Green-and-Sustainable-Software-Engineering.pdf (accessed on 18 March 2022).
29. Khari, M.; Kumar, N. Comparison of six prioritization techniques for software requirements. *J. Glob. Res. Comput. Sci.* **2013**, *4*, 38–43.
30. Sahoo, K.S.; Tripathy, B.K.; Naik, K.; Ramasubbareddy, S.; Balusamy, B.; Khari, M.; Burgos, D. An evolutionary SVM model for DDOS attack detection in software defined networks. *IEEE Access* **2020**, *8*, 132502–132513. [[CrossRef](#)]
31. Son, L.H.; Pritam, N.; Khari, M.; Kumar, R.; Phuong, P.T.M.; Thong, P.H. Empirical study of software defect prediction: A systematic mapping. *Symmetry* **2019**, *11*, 212. [[CrossRef](#)]
32. Kasunic, M. *Designing an Effective Survey*; Technical Report; Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst: Pittsburgh, PA, USA, 2005.
33. Pfleeger, S.L.; Kitchenham, B.A. Principles of survey research: Part 1: Turning lemons into lemonade. *ACM SIGSOFT Softw. Eng. Notes* **2001**, *26*, 16–18. [[CrossRef](#)]
34. Kitchenham, B.A.; Pfleeger, S.L. Principles of survey research part 2: Designing a survey. *ACM SIGSOFT Softw. Eng. Notes* **2002**, *27*, 18–20. [[CrossRef](#)]
35. Kitchenham, B.A.; Pfleeger, S.L. Principles of survey research: Part 3: Constructing a survey instrument. *ACM SIGSOFT Softw. Eng. Notes* **2002**, *27*, 20–24. [[CrossRef](#)]
36. Kitchenham, B.; Pfleeger, S.L. Principles of survey research part 4: Questionnaire evaluation. *ACM SIGSOFT Softw. Eng. Notes* **2002**, *27*, 20–23. [[CrossRef](#)]
37. Kitchenham, B.; Pfleeger, S.L. Principles of survey research: Part 5: Populations and samples. *ACM SIGSOFT Softw. Eng. Notes* **2002**, *27*, 17–20. [[CrossRef](#)]
38. Kitchenham, B.; Pfleeger, S.L. Principles of survey research part 6: Data analysis. *ACM SIGSOFT Softw. Eng. Notes* **2003**, *28*, 24–27. [[CrossRef](#)]
39. Punter, T.; Ciolkowski, M.; Freimut, B.; John, I. Conducting on-line surveys in software engineering. In *Proceedings of the 2003 International Symposium on Empirical Software Engineering, ISESE 2003*, Rome, Italy, 30 September–1 October 2003; IEEE: Hoboken, NJ, USA, 2003; pp. 80–88.
40. Saputri, T.R.D.; Lee, S.W. Software sustainability requirements: A unified method for improving requirements process for software development. In *Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE)*, Notre Dame, IN, USA, 20–24 September 2021; IEEE: Hoboken, NJ, USA, 2021; pp. 506–507.
41. Saputri, T.R.D.; Lee, S.W. Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. *Inf. Softw. Technol.* **2021**, *129*, 106407. [[CrossRef](#)]
42. Becker, C.; Chitchyan, R.; Duboc, L.; Easterbrook, S.; Penzenstadler, B.; Seyff, N.; Venters, C.C. Sustainability design and software: The karlskrona manifesto. In *Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, Italy, 16–24 May 2015; IEEE: Hoboken, NJ, USA, 2015; Volume 2, pp. 467–476.
43. Naumann, S.; Dick, M.; Kern, E.; Johann, T. The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustain. Comput. Informatics Syst.* **2011**, *1*, 294–304. [[CrossRef](#)]
44. Garcia-Berna, J.A.; Fernandez-Aleman, J.L.; de Gea, J.M.C.; Toval, A.; Mancebo, J.; Calero, C.; Garcia, F. Energy efficiency in software: A case study on sustainability in personal health records. *J. Clean. Prod.* **2021**, *282*, 124262. [[CrossRef](#)]
45. Gupta, S.; Lago, P.; Donker, R. A framework of software architecture principles for sustainability-driven design and measurement. In *Proceedings of the 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, Stuttgart, Germany, 22–26 March 2021; IEEE: Hoboken, NJ, USA, 2021; pp. 31–37.
46. Easterbrook, S.; Singer, J.; Storey, M.A.; Damian, D. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 285–311.
47. Nihei, Y.; Terashima, M.; Suzuki, I.; Morikawa, S. Why are four eyes better than two? Effects of collaboration on the detection of errors in proofreading. *Jpn. Psychol. Res.* **2002**, *44*, 173–179. [[CrossRef](#)]