Software Sustainability Perceptions in the Industry: A Questionnaire Study

Jennifer Gross[a], Temesgen Hagos Mengesha and Sofia Ouhbi[b]
Dept. Information Technology, Uppsala University, Uppsala, Sweden
fi

Keywords: Software Sustainability, Questionnaire Study, Industry, Practitioners, Perceptions.

Abstract: Despite growing awareness, many industry efforts regarding software sustainability often appear superficial and fail to address its complex, multifaceted nature. This paper examines software sustainability practices in the industry through the perspectives of IT practitioners. A questionnaire study involving 23 professionals based in Sweden revealed a significant gap in understanding, with 35% of participants unfamiliar with the term "software sustainability." Most definitions provided by participants focused on technical aspects, overlooking economic and social dimensions. The findings indicate that key barriers perceived by participants to integrating sustainability include a lack of awareness, time and budget constraints, and skepticism toward sustainability metrics. A majority of respondents recognized the link between sustainability and software quality. To promote sustainable software practices, respondents recommended embedding sustainability into industry practices and educational curricula, as well as developing clear metrics to measure its impact.

1 INTRODUCTION

Software plays a vital role in various sectors, including education, healthcare, and industry, with global software sales projected to reach $896.20 billion by 2029 (Statista, 2024). As Information and Communications Technology (ICT) systems now consume 10% of the world's electricity (Verdecchia et al., 2021), the urgency of incorporating sustainability into software development has never been greater. Understanding how sustainability principles are applied throughout the software development lifecycle is essential as our reliance on software continues to grow.

Despite increasing awareness, many industry efforts regarding sustainability are perceived as superficial, often overlooking the complexities of the issue (Kretschmer, 2022). Sustainability is a multifaceted concept that encompasses technical, social, environmental, and economic dimensions (McGuire et al., 2023). Sustainable software can be defined as software that minimizes environmental impact and maximizes resource efficiency across all stages of its lifecycle, while also considering economic and social factors (Calero et al., 2021). However, the software engineering research community lacks a consensus on what constitutes sustainable software (McGuire et al., 2023; Gross and Ouhbi, 2024a).

In the ICT industry, the complexity of software sustainability is often oversimplified, resulting in strategies that focus narrowly on aspects such as energy efficiency or software durability, while neglecting broader, interconnected sustainability elements (Venters et al., 2023). A recent tertiary study (Gross and Ouhbi, 2024a) identified several key challenges that impede progress in the field: ambiguity in defining software sustainability, uncertainty about when to integrate sustainability into software development, a lack of assessment metrics and tools, limited perspectives on sustainability in software systems, and insufficient awareness of sustainable practices among developers and end-users.

While various research studies have explored software sustainability, little is known about current industry practices. A recent review (Danushi et al., 2024) focusing on environmental sustainability found that the Technology Readiness Level (TRL) of existing environmentally sustainable software design solutions is low, with few studies reaching advanced stages—an essential requirement for advancing the field in practice.

This paper addresses the gap in research on how software sustainability is implemented in the industry by exploring the perspectives of IT practition-

[a] https://orcid.org/0009-0008-6397-4087
[b] https://orcid.org/0000-0001-7614-9731

Gross, J., Mengesha, T. H. and Ouhbi, S.
Software Sustainability Perceptions in the Industry: A Questionnaire Study.
DOI: 10.5220/0013405300003928
In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 197-207
ISBN: 978-989-758-742-9; ISSN: 2184-4895

ers. We conducted a questionnaire study to gather insights from industry professionals regarding their awareness and implementation of software sustainability. Specifically, this paper seeks to answer the following research questions (RQs):

RQ1: How do IT professionals perceive software sustainability?

RQ2: How do IT professionals perceive the relationship between software quality and software sustainability?

RQ3: How do IT professionals perceive the feasibility and obstacles of incorporating sustainability into practice?

2 RELATED WORK

The study by (Oyedeji et al., 2021) utilized grounded theory derived from a workshop with various software development practitioners to understand their definitions of software sustainability. They compared these practitioners' definitions with those from academia. Initially, when asked, 75% of early-career developers reported having no understanding of software sustainability, in contrast to 75% of mid-level and senior practitioners, who had some form of definition. The study revealed that practitioners prioritized economic and technical sustainability as the most important dimensions, while they perceived no connection between their work and social sustainability. The authors emphasized the need for greater collaboration between academia and industry to align their perceptions of software sustainability.

The study by (Karita et al., 2022) aimed to bridge this knowledge gap by enhancing professional competencies and establishing an industry-standard framework for software sustainability. Their research, which included a systematic mapping study and industry surveys, highlighted the importance of involving stakeholders and managing trade-offs when addressing sustainability concerns throughout the software development lifecycle. They connected these concerns to five key dimensions of sustainability: economic, environmental, technical, social, and individual.

The study by (Bambazek et al., 2022) examined practitioners' perspectives on integrating sustainability into agile development, focusing specifically on the Scrum framework. They found that practitioners often struggled to incorporate sustainability into agile practices, underscoring the need for expert input to effectively assess the impacts of software systems. While Scrum was viewed as a potential solution, not all of its artifacts aligned with sustainability principles.

The study by (Heldal et al., 2024) conducted interviews across 27 organizations in nine countries to investigate the challenges faced in achieving sustainability goals within software development. Their study found that, despite organizations' eagerness to assess the sustainability impacts of their products and processes, a lack of expertise in the field hindered progress. This knowledge gap complicates the balancing of customer needs, short-term economic goals, and long-term sustainability missions.

A survey by (Condori-Fernandez and Lago, 2018) targeting ICT practitioners with expertise in sustainability aimed to explore their perception about how quality requirements may contribute to software sustainability. Their findings reveal that modifiability is perceived to significantly impact both technical and environmental sustainability, while attributes like functional correctness, availability, interoperability, and recoverability enhance software endurability. Security, satisfaction, and freedom from risk are also perceived as strong contributors to social sustainability, with satisfaction also positively influencing economic sustainability.

3 METHODOLOGY

A self-administered online questionnaire was sent to a list of IT practitioners by email. The study focused on IT practitioners working in Sweden-based institutions. The selected individuals were identified based on their experience as external thesis supervisors for Uppsala University students in IT, who supervised students in their respective companies either in 2023 or at the beginning of 2024. A total of 247 contacts were invited to participate in the study via email, which included a study overview and a link to the questionnaire. Participants completed the questionnaire voluntarily and anonymously.

The questionnaire, presented in Table 1 and developed using Google Forms, covers three main areas: participant background, awareness of software sustainability, and perceptions of its relationship to software quality, applicability, and challenges in practice. It includes both open-ended and multiple-choice questions, allowing for detailed responses and specific answers. The questionnaire was validated by two researchers and refined based on their feedback before distribution. The estimated time to complete the questions is between 5 to 10 minutes. It was open for responses from May 21 to June 10, 2024.

Table 1: Questionnaire. *Acronym: Ob. for Obligatory.*

ID	Question	Response Option	Ob.
Background			
Q1	Gender	Open	No
Q2	Age	18-24, 25-34, 35-44, 45-54, 55-64, 65+	Yes
Q3	What is the highest level of education you've completed?	Bachelor's Degree, Master's Degree, Doctorate/Ph.D., Other (Open)	Yes
Q4	In which sector are you active in?	Information Technology, Healthcare, Finance, Manufacturing, Education, Other	Yes
Q5	Job title	Open	No
Q6	How many years of experience do you have in industry?	Less than 1 year, 1-3 years, 3-5 years, 5-10 years, 10-20 years, 20+ years	Yes
Awareness			
Q7	How familiar are you with the concept of software sustainability?	Very familiar with software sustainability, Somewhat familiar with software sustainability, Never heard of software sustainability, Other (Open)	Yes
Q8	If you are familiar with software sustainability, please specify the source from which you gained your understanding of it:	Academic studies or courses, Industry training or workshops, Media, Other	No
Q9	How would you define software sustainability?	Open	Yes
Perception			
Q10	How important do you think sustainability is in software systems?	5-point Likert Scale from Not Important to Extremely Important	Yes
Q11	Which of these aspects do you think are important towards software sustainability?	- Social aspects, which aim at preserving societal communities in their solidarity and services. - Economic aspects, which aim at maintaining capital and added value. - Environmental aspects aim at improving human welfare by protecting natural resources such as water, land, air, minerals, and ecosystem services. - Individual aspects aim at maintaining human capital, including health, education, skills, knowledge, leadership, and access to services. - Technical aspects, which focus on the longevity of information, systems, and infrastructure, ensuring their adequate evolution with changing surrounding conditions. - Other (Open)	Yes
Q12	In which of the following stages do you think software sustainability should be considered?	Planning, Requirements, Design, Implementation, Testing, Deployment, Maintenance, Other (Open)	Yes
Q13	In your opinion, is there a relationship between software quality and software sustainability?	- Yes, software sustainability is a part of software quality. - Yes, software quality is a part of software sustainability. - Yes, they are complementary. - No, there is no relationship. - I don't know. - Other (Open)	Yes
Q14	How do you perceive the current practices in industry in relation to software sustainability?	Excellent, Satisfactory, Insufficient, No opinion, Other (Open)	Yes
Q15	In your opinion, which of the following actions are critical to advance sustainability in software systems?	- Increasing awareness about software sustainability among software professionals. - Enhancing sustainability education in higher education. - Implementing policies about software sustainability. - Developing standards about software sustainability. - No actions are required. - Other (Open)	Yes
Q16	Which of these product quality attributes do you think are relevant to software sustainability?	- Functional Suitability: the capability of software to meet the stated and implied needs of users under specified conditions. - Performance Efficiency: the capability of software to perform its functions within specified time and throughput parameters and efficiently use resources under specified conditions. - Compatibility: the capability of software to exchange information with other products and perform its functions while sharing the same environment and resources. - Interaction Capability: the capability of software to be used by specified users to exchange information with the system via the user interface to complete the intended task. - Reliability: capability of a software to perform specified functions under specified conditions for a specified period of time without interruptions and failure. - Security: capability of a software to protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors. - Maintainability: capability of a software to be modified by the intended maintainers with effectiveness and efficiency. - Flexibility: capability of a software to be adapted to changes in its requirements, contexts of use, or system environment. - Safety: capability of a software under defined conditions to avoid a state in which human life, health, property, or the environment is endangered. - None of the product quality attributes are important. - Other (Open)	Yes
Q17	Which of these quality-in-use attributes do you think are relevant to software sustainability?	- Usability: extent to which a software can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. - Accessibility: extent to which software systems, environments and facilities can be used by people from a population with the widest range of user needs, characteristics and capabilities to achieve identified goals in identified contexts of use. - Suitability: extent to which behaviors or outcomes, or both, of a software meet specified quality requirements when used. - Freedom from Economic Risk: extent to which a software mitigates the potential risk to financial status, efficient operation, commercial property, reputation, or other aspects in the intended contexts of use. - Freedom from Environmental and Societal Risk: extent to which a software mitigates the potential risk to the environment and society at large in the intended contexts of use. - Freedom from Health Risk: extent to which a software mitigates the potential risk to people's health in the intended contexts of use. - Freedom from Human Life Risk: extent to which a software mitigates the potential risk to people's lives in the intended contexts of use. - Experience: extent to which users or stakeholders accumulate knowledge acquired over time, especially that gained in a particular profession. - Trustworthiness: extent to which users or stakeholders have confidence that their expectations are met in a verifiable way. - Compliance: extent to which a user or other stakeholder has confidence that a software meets requirements, as required by rules or laws. - None of the quality-in-use attributes are important. - Other (Open)	Yes
Q18	How would you rate the consideration of end-users' perspectives in practical efforts toward sustainable software?	5-point Likert Scale from Very Poor to Excellent	Yes
Q19	In your opinion, what aspects of software sustainability do you believe are important to the end-user?	Open	No
Q20	In your opinion, are there any challenges or barriers you believe are hindering the advancement of software sustainability in practice? If so, please describe:	Open	No
Q21	Please use this space to share any additional comments, suggestions, or feedback you have regarding software sustainability in practice.	Open	No

4 RESULTS

Only 23 practitioners out of 247 participated in the study. Questionnaire results can be accessed here: https://shorturl.at/rEZAu.

4.1 Background

Table 2 presents the results of Q1-4 and Q6. Out of 23 responses, 10 (43%) had a master's degree, 9 (39%) had a Ph.D and the rest had a bachelor's degree. The demographics of the respondents indicate that the minimum education level is a bachelor's degree. More than half of the respondents have over 10 years of industry experience: 22% have 10-20 years, 35% have more than 20 years, 26% have 5-10 years, 9% have 3-5 years, and 9% have 1-3 years. Except for two respondents, all stated that they work in the information technology sector, one respondent works in R&D, but their specific sector is unknown, and one works in manufacturing.

Table 2: Respondent's Background.

Item	Total	%
Gender		
Male	19	83%
Unspecified	4	17%
Age		
25-34	4	17%
35-44	10	43%
45-54	7	30%
55-64	2	9%
Education		
Bachelor's Degree	4	17%
Doctorate/Ph.D.	9	39%
Master's Degree	10	43%
Sector of activity		
Education	1	4%
Information Technology	20	87%
Manufacturing	1	4%
R&D	1	4%
Years of experience		
1-3 years	2	9%
3-5 years	2	9%
5-10 years	6	26%
10-20 years	5	22%
20+ years	8	35%

Table. 3 presents the results of Q5. The participants include professionals from both industry and academia. The majority of respondents have leadership responsibilities and decision-making roles in their organizations, holding either managerial or senior positions.

Table 3: Job titles as reported by participants.

Job Title	Participant ID	Total
Project Manager	P7, P20	2
Researcher	P15, P22	2
Senior Software Engineer	P5, P11	2
Software Engineer	P6, P12	2
Analytics Engineer	P8	1
CTO	P3	1
Developer	P1	1
Director of Data Engineering	P23	1
Expert, Adj Professor	P18	1
Head of Operations	P14	1
Integration Lead	P4	1
Manager Mobile Development	P2	1
Professor	P13	1
Senior Backend Developer	P17	1
Sr. Lead Software Engineer	P16	1
Systems Developer	P9	1
Team Lead Software	P19	1
Vice Principal (Associate Professor)	P21	1
(blank)	P10	1

4.2 Awareness of Respondents Toward Software Sustainability

Fig. 1 shows the results of Q7. Surprisingly, 8 out of 23 participants had never heard of the concept of software sustainability before. One participant (P9) selected other and responded *"I have not heard the explicit term but I have heard discussions about environmental sustainability of software."* Table 4 shows the responses to Q7 according to participants' educational level and years of experience.

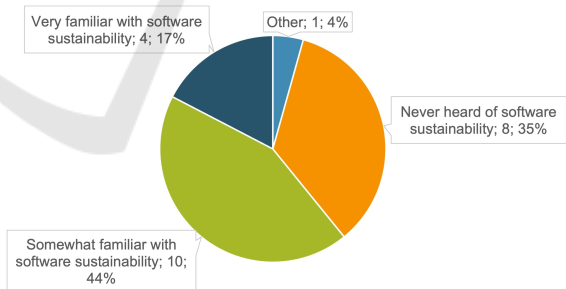

Figure 1: Familiarity with the Concept of Software Sustainability.

The results of Q8 show that among those who had prior awareness of software sustainability, their understanding came from industry workshops/training (38%), academia (19%), media (38%), and peers (5%).

Results of Q9 show that, with the exception of two respondents who did not know how to define software sustainability, the rest of the participants provided diverse definitions of software sustainability, focusing on several key themes:

Table 4: Familiarity with Software Sustainability according to educational level and years of experience.

Familiarity with Software Sustainability	Years of Experience	Bachelor's Degree	Doctorate/Ph.D.	Master's Degree
Other	1-3 years	P9		
Never heard of software sustainability	5-10 years	P1, P16	P8, P10	
	10-20 years		P14, P22	
	20+ years	P12, P18		
Somewhat familiar with software sustainability	3-5 years			P6, P11
	5-10 years			P5, P19
	10-20 years	P23	P20	
	20+ years	P2, P3	P15	P7
Very familiar with software sustainability	1-3 years		P13	
	10-20 years			P4
	20+ years	P17	P21	

- Environmental Impact: Many participants emphasize sustainability in relation to the environment, particularly in terms of reducing energy consumption (P11 and P16) and minimizing the environmental footprint of software development and usage (P5, P6, P10, and P14). Some highlight the importance of green practices in the software lifecycle (P23).
- Longevity and Maintainability: Several participants associate sustainability with software's long-term viability, maintainability, and adaptability (P7, P13, P18, P20, and P22). This includes ensuring that software can evolve, remain relevant, and integrate well with other systems without requiring excessive rewrites (P3 and P15).
- Resource Efficiency: A common theme is the efficient use of resources—whether it is hardware (P11) or computational resources (P2, P8, and P9). Participants also mention writing code that minimizes waste and is resource-efficient (P8, P9, and P13).
- Social and Ethical Considerations: A few participants connect sustainability to broader social aspects, such as the impact on society (P5 and P11). They suggest that software should promote social sustainability by being secure and safe, as well as avoiding unnecessary reliance on specific hardware (P11).
- Quality and Usability: Sustainability is also linked to ensuring software quality over time—meaning it remains functional, secure, and easy to maintain, inspect, and test (P4, P19, and P20).

Participants defined software sustainability using both narrow and broad terms. Narrow definitions focused on aspects like energy efficiency, resource optimization, and minimizing negative environmental impacts, with comments such as *"developing software that consumes less energy in practice"* (P16) and the idea that *"software is sustainable if it is easy to maintain"* (P9). These perspectives emphasized specific criteria, including limiting unnecessary resource consumption (P2). In contrast, broad definitions acknowledged the importance of social and environmental costs in software development, as highlighted by the need to analyze these costs (P17). Participants considered the software's lifecycle, including its integration, adaptability, and documentation (P13), and emphasized that sustainability should incorporate stakeholder involvement and consider societal impact (P20).

4.3 Perception of Respondents Toward Software Sustainability

Fig. 2 presents the results of Q10. More than half of the respondents considered software sustainability to be an important or very important topic. Seven participants (P3, P5, P11, P13, P16, P19, and P21) viewed it as very important. Among them, two (P13 and P21) were very familiar with the term, while one (P16) had never heard of it before. On the other hand, two respondents (P10 and P12) regarded software sustainability as unimportant, and both had never encountered the term prior to the questionnaire study.

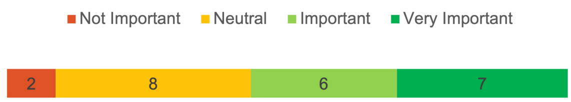

Figure 2: Perception of the importance of Software Sustainability.

For Q11, respondents were provided with definitions of each aspect as outlined by the Karlskrona Manifesto (Becker et al., 2015)—economic, environmental, technical, social, and individual—and asked to select the aspects they believed were important for software sustainability. Three participants either did not know (P1 and P12) or did not provide a response (P19). Fig. 3 shows the count of important aspects selected by the remaining 20 participants. The responses clearly indicate that the environmental and technical aspects are perceived by the majority of participants as particularly significant.

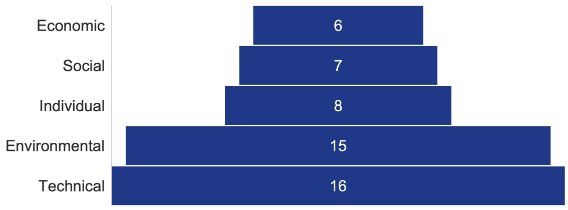

Figure 3: Important aspects for Software Sustainability (N=20).

For Q12, respondents were asked which stages of the software development cycle—requirements, planning, design, implementation, testing, and maintenance—should incorporate software sustainability. Two respondents (P1 and P12) indicated they did not know. Fig. 4 presents the responses from the remaining 21 participants. The results show that the stages of requirements, planning, and design were prioritized the most. Six participants (P3, P7, P9, P15, P17, and P23) agreed that sustainability should be considered at all stages, while others provided varying opinions.

Figure 4: Software Sustainability Consideration in the Development Stages (N=21).

Table 5 shows the results of Q13, which addresses participants' perceptions of the relationship between software sustainability and software quality. A total of 35% of respondents consider software sustainability to be a part of software quality, while 22% believe software quality is a part of software sustainability. Also, 17% view them as complementary. Four respondents (17%) (P1, P10, P12, and P14) indicated that they did not know, and one respondent (P8) stated that there is no relationship.

Fig. 5 shows that only 2 respondents perceived that the current practices in industry in relation to software sustainability are satisfactory. The majority of respondents consider them as insufficient and one respondent (P23) perceives them as non existent stating *"Not really existent. Google Cloud talks about this some, but not much."*

Fig. 6 presents respondents' views on critical actions to advance sustainability in software systems.

Table 5: Relationship between Software Quality and Software Sustainability.

Response	Total	%
Yes, software sustainability is a part of software quality.	8	35%
Yes, software quality is a part of software sustainability.	5	22%
Yes, they are complementary.	4	17%
No, there is no relationship.	1	4%
Yes, since efficiency and in-code documentation are a part of it.	1	4%
Don't know	4	17%

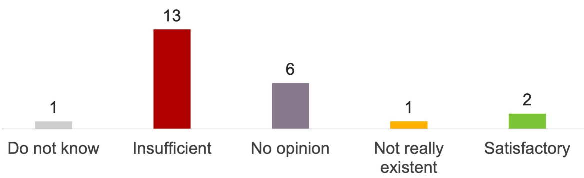

Figure 5: Perception of current practices in industry in relation to Software Sustainability.

The majority of respondents (83%) consider increasing awareness about software sustainability among software professionals as a critical action. Nearly half of the respondents (48%) think enhancing sustainability education in higher education and developing standards for software sustainability are important steps. One respondent (P2) also suggested the need for appropriate metrics to measure sustainability.

To respond to Q16, respondents were given the option to select relevant product quality characteristics from the ISO/IEC 25010:2023 model in relation to software sustainability. Fig. 7 shows that *Maintainability* had the highest number of selections, chosen by 16 respondents (70%), while *Functional Suitability* had the lowest number of selections, chosen by 6 respondents (26%). One respondent (P10) selected *Other* and stated that *"All those things seem like obvious good things. I would think about them more as normal requirements and not being about software sustainability."*

Fig. 8 presents respondents' views on software sustainability in relation to the Quality in Use characteristics from the ISO/IEC 25010:2023 model. *Freedom from Environmental and Societal Risk* was selected by 13 respondents (57%), making it the most chosen characteristic, while *Experience* was the least selected, chosen by only two respondents (9%). P2 selected *Other* and gave the same response as in Q16. One respondent (P23), in addition to selecting some characteristics, noted that software should function within regulatory limits and cautioned that excessive requirements can lead to over-design. While AI risks should be considered, he emphasized that such rules should not apply universally to all software.

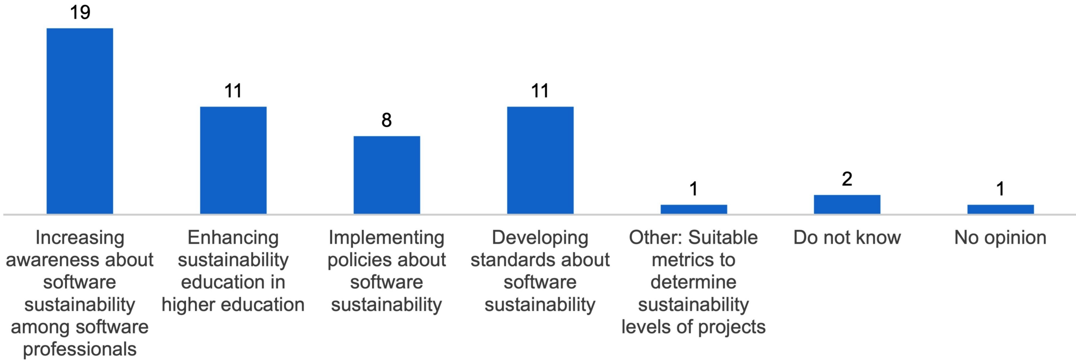

Figure 6: Critical Actions to Advance Sustainability.

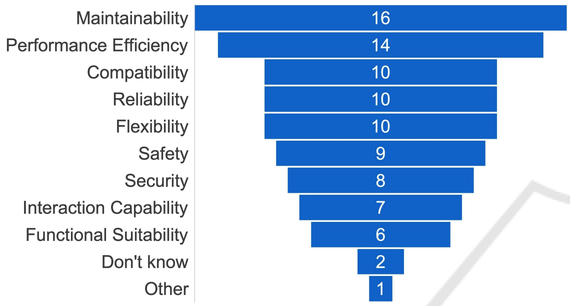

Figure 7: Relevant Software Product Quality Characteristics to Software Sustainability.

Fig. 9 shows how respondents rated the considerations of end-users' perspectives in practical effort towards sustainable software. None of the respondents considered that these considerations are excellent. However, the majority consider them good and very good.

Out of the 16 respondents to Q19, thirteen identified key aspects of software sustainability important to end-users, including reliability, stability, and ease of use. P3 stated, *"Reliability and ease of use are paramount"*, while P9 added that *"slow software is not fun to use, and unstable software is unreliable"*. Environmental impact emerged as a recurring theme, with many participants highlighting low energy consumption; P5 mentioned this alongside reliability and efficiency. P11 noted the growing interest in eco-friendly labels, suggesting that such markers can influence organizational decisions. Usability and trustworthiness were also deemed essential, as P7 indicated that *"suitability, usability, and trustworthiness might be the most important aspects"*. P16 emphasized data portability and long-term system stability as vital for sustainable software.

Respondents stressed the importance of clear communication about sustainability aspects. P15 remarked, *"A clear specification of which aspects are being considered for the product is crucial"*. P23 emphasized the need for user awareness of sustainability issues, stating that *"people need to be aware of the issue and the benefits"*. However, P13 expressed skepticism about end-users' understanding of sustainable software, highlighting a gap in knowledge with this comment: *"I am not convinced the end user can fully grasp all of the dimensions required to build sustainable software"*.Overall, while critical sustainability aspects were recognized, respondents called for improved communication and awareness to empower end-users in making informed decisions.

For Q20, fifteen respondents identified several challenges hindering the advancement of software sustainability in practice. P3 cited a short-sighted economic focus, while P4 pointed to the need for better education on sustainability. P5 mentioned a lack of political will, distrust in scientific consensus, and profit-driven development as significant barriers. P7 noted the complexity of balancing customer and stakeholder expectations with budget constraints, which often impedes investment in sustainability initiatives. P8 highlighted time constraints in software projects, and P9 criticized modern development methods for neglecting performance, leading to resource drains.

P11 expressed concern that energy consumption is often overlooked in the rush to develop new features, stating that current practices prioritize *"cool"* innovations without considering their environmental impact. P13 emphasized the lack of diversity among software developers, which may limit awareness of essential sustainability aspects. P15 found the broadness of the concept of sustainability to be a barrier, as the numerous factors to consider can be time- and cost-prohibitive for projects. P16 remarked that capitalism and sustainability often appear mutually exclusive, while P17 echoed the sentiment that general education on sustainability is lacking. P19 highlighted the pressure of short deadlines imposed by stakeholders. P20 noted that many developers lack the maturity

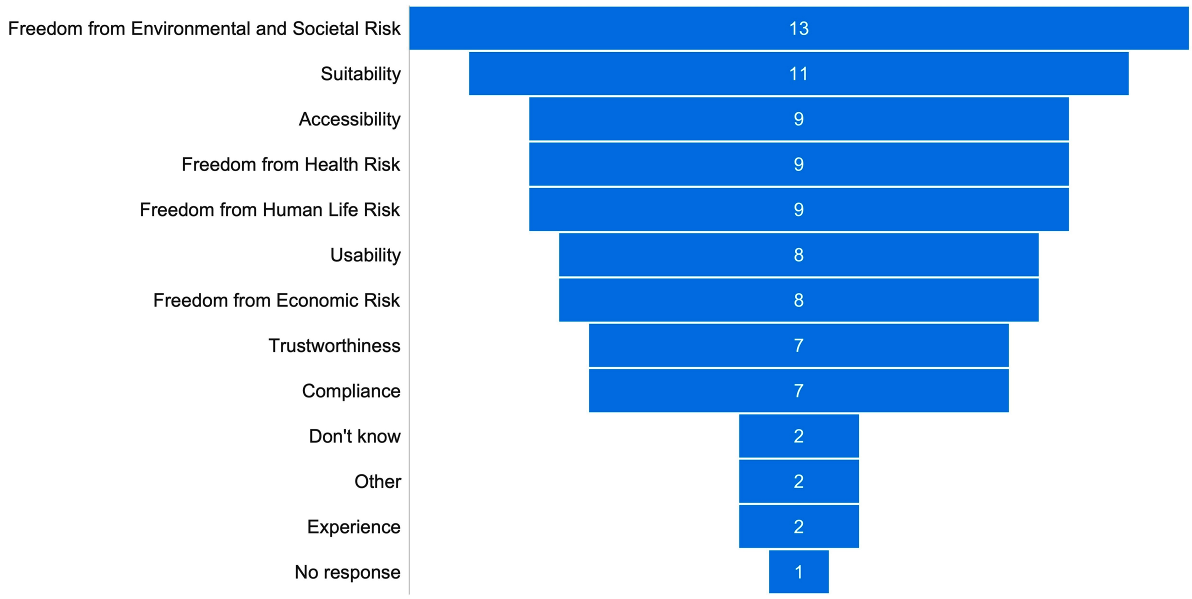

Figure 8: Relevant Quality in Use (sub)Characteristics to Software Sustainability.

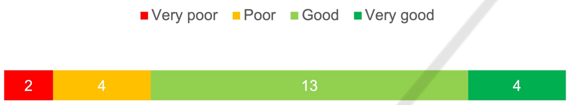

Figure 9: Consideration of end-users perspectives in sustainable software.

to understand the importance of sustainability, while P21 mentioned cybersecurity concerns as a barrier. Lastly, P23 suggested that quantifying sustainability benefits could help illustrate its importance, stating that showcasing savings from sustainable practices could encourage adoption.

For Q21, only a few respondents shared comments on software sustainability. P1 and P12 expressed frustration with being asked questions about a topic they didn't understand. P18 criticized the survey for not defining "software sustainability". P10 recognized software sustainability as related to reducing environmental impact but found the term too broad, suggesting it overlaps with standard software development goals. P11 emphasized the need to integrate sustainability into educational curricula, calling for legislative support and improved tools like energy-aware integrated development environments (IDEs) to make early energy-saving decisions. P16 warned of the rapid growth of harmful applications for large language models (LLMs) and their significant energy demands, stating, *"urgent action is required to avoid catastrophic long-term consequences"*. P20 highlighted the importance of considering the broader context of software, especially in embedded systems, noting: *"You cannot look at code alone"*. P21 mentioned cybersecurity concerns. Finally, P23 acknowledged a lack of awareness about software sustainability while recognizing its potential benefits.

5 DISCUSSION

5.1 Main Findings

5.1.1 Limited and Narrow Perspective of Software Sustainability

This study highlights a significant gap in practitioners' understanding of software sustainability, with 8 out of 23 participants reporting they had never encountered the term, even among those with over 20 years of experience. Most participants' definitions centered on technical aspects, with limited acknowledgment of economic and individual dimensions, indicating a narrow perspective. Interestingly, this differs slightly from the study by (Bambazek et al., 2022), where environmental and technical aspects were more heavily emphasized. Similar to findings by both (Bambazek et al., 2022) and (Heldal et al., 2024), our results underscore a widespread lack of understanding of software sustainability, even as interest in the concept grows.

Our findings diverge from the study by (Oyedeji et al., 2021), which linked sustainability primarily to cost-effectiveness and maintainability. In contrast, our participants—19 out of 23 of whom had over five years of industry experience—had a broader view that included social and individual dimensions. This may reflect the higher level of experience among our participants compared to (Oyedeji et al., 2021)'s sample, where only five had similar expertise.

Environmental sustainability also emerged as a recurring theme, with several respondents emphasizing the importance of energy-efficient software. The growing influence of eco-friendly labels and environmental considerations highlights an increasing awareness of sustainability in software systems. While technical aspects dominated the discussion, respondents did acknowledge the significance of economic and individual dimensions, aligning with the five aspects of sustainability outlined in the Karlskrona Manifesto (Becker et al., 2015). This suggests an evolving recognition of the need for a more comprehensive approach to software sustainability.

5.1.2 Other Challenges to Software Sustainability

Time pressures and budget constraints were also cited as significant challenges by participants. These constraints often push teams to prioritize immediate functionality over long-term sustainability, resulting in sustainable practices being sidelined. Profit-driven strategies and skepticism toward the scientific consensus on sustainability further complicate the issue (Kretschmer, 2022). Participants such as P3 and P5 expressed doubts about the relevance of sustainability metrics, viewing them as costly or impractical for meeting short-term project goals. Moreover, the absence of regulatory frameworks mandating sustainable practices leaves organizations with little incentive to prioritize sustainability. In such cases, it is often left to individual organizations to champion sustainability initiatives, limiting broader industry adoption. Legislation may be required to create constraints on software in order to motivate the creation of more sustainable software, as emphasized by P11.

Stakeholder involvement is another important aspect to consider in relation to software sustainability. The work by (Karita et al., 2022) motivated the importance of involving stakeholders and assessing sustainability early and throughout all stages of software development, which aligns with our findings. P19 stated that short deadlines imposed by stakeholders pose a challenge to incorporating sustainability in software development. P7 corroborated this by highlighting that the complexity and requirements of the software, combined with budget constraints, make it difficult to identify ways to integrate sustainability. This suggests a disconnect between stakeholders' expectations for the system and the time allocated for refactoring or sustainability analysis. Alternatively, it may indicate the absence of an internal framework for assessing sustainability.

Another challenge is the end users' lack of understanding of what goes into creating sustainable software. A lack of transparency was also identified as a key reason why end users may struggle to fully grasp the impact of software sustainability on them. This includes the issue of trust highlighted by P11, who stated, "[End users] don't understand what enormous risk they impose on themselves when trusting software systems with every aspect of their life". However, trustworthiness, a quality-in-use characteristic, was identified by nine participants as an important factor in software sustainability.

5.1.3 Perceptions of the Relationship Between Software Sustainability and Quality

The majority of participants recognized a connection between software sustainability and quality. This aligns with the work of (Condori-Fernandez and Lago, 2018), which maps sustainability dimensions to ISO/IEC quality characteristics. In total, 18 participants identified a relationship between software quality and sustainability, while only one stated that no such relationship existed. However, this participant was unfamiliar with the concept of software sustainability and still acknowledged the importance of several software quality characteristics (performance efficiency, compatibility, maintainability, usability, and suitability) in sustainability.

Among the quality characteristics identified by respondents (in more than 10 responses) as relevant to software sustainability, two were related to product quality (maintainability and performance efficiency), and two pertained to quality in use (freedom from environmental and societal risk, and suitability). In contrast, characteristics such as interaction capability, functional suitability, compliance, trustworthiness, and user experience received fewer than eight responses and were overlooked by the majority.

Participants' perspectives on the relationship between sustainability and software quality varied. While 36% viewed sustainability as intrinsic to software quality—aligning with evolving quality standards such as ISO/IEC 25010:2023—27% considered software quality a subset of sustainability, and 14% saw the two as complementary. These differing perspectives highlight a gap in understanding that impacts decision-making, prioritization, and the integration of sustainability metrics in software development. Despite growing recognition of the importance of sustainability and its relationship to software quality, this variation suggests that further education and clarification are needed to bridge the gap.

5.1.4 Recommendations for Advancing Software Sustainability

To advance software sustainability, increasing awareness and education among practitioners is crucial. The majority of respondents (83%) emphasized the need to incorporate sustainability into both industry practices and educational curricula. Research has been conducted on integrating sustainability into education by embedding it in requirements engineering and software engineering courses; however, further efforts are needed (Oyedeji et al., 2023; Moreira et al., 2024). Engaging with a more diverse range of stakeholders can facilitate a more comprehensive approach to understanding sustainability. Given the broad scope of the topic, there is a risk of focusing on one aspect while overlooking its wider implications.

Our findings highlight the need to embed sustainability considerations across all stages of the software development lifecycle—from requirements to maintenance. This holistic approach is essential for fostering long-term sustainable practices in software systems. To do so, there is a need to consider sustainability as a starting point when developing software systems (Gross and Ouhbi, 2024b). Designing for sustainability, such as using the Karlskrona Manifesto for Sustainability Design, involves not only examining the software system itself but also considering its surrounding environment and broader impacts (Becker et al., 2014).

Clear communication about sustainability aspects and the development of metrics to measure its benefits were also recommended by the participants. This need has been emphasized beyond this research as well (König et al., 2024; Gross and Ouhbi, 2024a). Conducting a comprehensive evaluation of sustainability requires access to clear measures and criteria (König et al., 2024; Gross and Ouhbi, 2024a). Having quantifiable data can facilitate comparisons between upfront costs and long-term savings, as well as justify investments in one area to promote benefits in another. Providing end users with clear, non-technical information can help them understand the sustainability choices made in software development, increasing their awareness of associated risks and decisions. Highlighting features that support more sustainable usage can also empower users to make informed decisions that reduce environmental impact.

Stronger legislation is needed to require companies to disclose sustainability metrics and provide stakeholders with transparent information. The lack of standardization enables practices like greenwashing, which the EU defines as ”giving a false impression of the environmental impact or benefits of a product” (European Parliament, 2024). In response, the EU has introduced laws to combat greenwashing, enforce verifiable sustainability claims, and develop regulations on the right to repair, ecodesign standards, and green claim verification—all of which could drive sustainable software development.

5.2 Limitations

This study has some limitations that may impact the generalizability of its findings, such as the limited number of participants and the focus on practitioners solely in Sweden. Another limitation is that the majority of respondents were male, which, while expected given the gender imbalance in the IT industry, may still influence the perspectives captured in this study. Moreover, three participants without prior knowledge contributed minimally to the study. However, since the study aimed to capture individual perceptions, other participants with no prior knowledge provided valuable insights into how they perceive software sustainability in practice. As a result, the findings should be viewed as preliminary insights for further research and may not fully represent the general perspectives of practitioners in Sweden. Furthermore, the data was collected through an online questionnaire, which limited the depth of understanding regarding respondents' opinions.

6 CONCLUSION AND FUTURE WORK

This paper presents a questionnaire study exploring how industry practitioners in Sweden perceive and implement software sustainability, focusing on their definitions, its relationship with software quality, and the challenges of adoption. The findings reveal a significant gap in understanding the term, with many experienced practitioners unfamiliar with it. The findings highlight also that sustainability is rarely considered in current practices, and several barriers hinder its adoption.

These findings underscore the need for improved education, standardized guidelines, clearer communication, and potentially legislative action to promote sustainable practices. Strengthening awareness and discourse around software sustainability will be essential for driving meaningful progress in the industry.

Future research should expand to various regions and industries to better assess software sustainability practices. By using diverse data collection methods and a larger participant pool, future studies can achieve more accurate results. We also plan to supplement our findings with interviews on current industry practices in Sweden and replicate the study in other countries.

ACKNOWLEDGMENT

We would like to express our gratitude to all the participants who actively took part in this questionnaire study.

REFERENCES

Bambazek, P., Groher, I., and Seyff, N. (2022). Sustainability in agile software development: A survey study among practitioners. In *2022 International Conference on ICT for Sustainability (ICT4S)*, pages 13–23. IEEE.

Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Mahaux, M., Penzenstadler, B., Rodriguez-Navas, G., Salinesi, C., Seyff, N., Venters, C., et al. (2014). The karlskrona manifesto for sustainability design. *arXiv preprint arXiv:1410.6968*.

Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., and Venters, C. C. (2015). Sustainability design and software: The Karlskrona Manifesto. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 467–476. IEEE.

Calero, C., Moraga, M. Á., and Piattini, M. (2021). *Introduction to software sustainability*. Springer.

Condori-Fernandez, N. and Lago, P. (2018). Characterizing the contribution of quality requirements to software sustainability. *Journal of Systems and Software*, 137:289–305.

Danushi, O., Forti, S., and Soldani, J. (2024). Environmentally sustainable software design and development: A systematic literature review. *arXiv preprint arXiv:2407.19901*.

European Parliament (2024). Stopping greenwashing: how the EU regulates green claims. https://www.europarl.europa.eu/topics/en/article/20240111STO16722/stopping-greenwashing-how-the-eu-regulates-green-claims. Accessed: Feb 12, 2025.

Gross, J. and Ouhbi, S. (2024a). Clearing the path for software sustainability. Accepted in 2030 Software Engineering Workshop. https://arxiv.org/abs/2405.15637.

Gross, J. and Ouhbi, S. (2024b). Sustainability as a starting point in software development. In *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 252–255. IEEE.

Heldal, R., Nguyen, N.-T., Moreira, A., Lago, P., Duboc, L., Betz, S., Coroamă, V. C., Penzenstadler, B., Porras, J., Capilla, R., et al. (2024). Sustainability competencies and skills in software engineering: An industry perspective. *Journal of Systems and Software*, 211:111978.

Karita, L., Mourão, B. C., and Machado, I. (2022). Towards a common understanding of sustainable software development. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*, pages 269–278.

König, C., Lang, D. J., and Schaefer, I. (2024). Sustainable software engineering: Concepts, challenges, and vision. *ACM Transactions on Software Engineering and Methodology*.

Kretschmer, A. (2022). Green pioneer or greenwasher? the tech industry and sustainability. URL: https://www.iresearchservices.com/blog/green-pioneer-or-greenwasher-the-tech-industry-and-sustainability/ [Accessed on Oct 10 2024].

McGuire, S., Schultz, E., Ayoola, B., and Ralph, P. (2023). Sustainability is stratified: Toward a better theory of sustainable software engineering. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1996–2008. IEEE.

Moreira, A., Lago, P., Heldal, R., Betz, S., Brooks, I., Capilla, R., Coroamă, V. C., Duboc, L., Fernandes, J. P., Leifler, O., et al. (2024). A roadmap for integrating sustainability into software engineering education. *ACM Transactions on Software Engineering and Methodology*.

Oyedeji, S., Adisa, M. O., Abdullai, L., and Porras, J. (2023). Application of sustainability awareness framework in software engineering courses: Perspectives from ICT students. In *ICT4S (Doctoral Symposium, Demos, Posters)*, pages 122–133.

Oyedeji, S., Shamshiri, H., Porras, J., and Lammert, D. (2021). Software sustainability: academic understanding and industry perceptions. In *Software Business: 12th International Conference, ICSOB 2021, Drammen, Norway, December 2–3, 2021, Proceedings 12*, pages 18–34. Springer.

Statista (2024). Global software market - statistics & facts. www.statista.com/outlook/tmo/software/worldwide. Accessed: Feb 11, 2025.

Venters, C. C., Capilla, R., Nakagawa, E. Y., Betz, S., Penzenstadler, B., Crick, T., and Brooks, I. (2023). Sustainable software engineering: Reflections on advances in research and practice. *Information and Software Technology*, page 107316.

Verdecchia, R., Lago, P., Ebert, C., and De Vries, C. (2021). Green IT and green software. *IEEE Software*, 38(6):7–15.