Empowering Growth: Implementation of Sustainable Software Requirement Engineering Practices in Pakistan

Javaria Umbreen[1], Samra Siddiqui[1], Muhammad Mughees Khan[1], Muhammad Umar[2]

[1]Comsats University Islamabad, Wah Campus

[2]University of Wah, Wah Cantt

[*]**Correspondence**: javaria.umbreen@ciitwah.edu.pk, Samra.siddiqui@ciitwah.edu.pk, m.mughees559@gmail.com, umar.aftab8326@gmail.com.

Citation | Umbreen. J, Siddiqui. S, Khan. M. M, Umar. M, "Empowering Growth: Implementation of Sustainable Software Requirement Engineering Practices in Pakistan", IJIST, Special Issue. pp 70-80, Oct 2024

Received| Oct 06, 2024 **Revised|** Oct 10, 2024 **Accepted|** Oct 15, 2024 **Published|** Oct 19, 2024.

Introduction/Importance of Study: Sustainability must be integrated into Software Requirements Engineering due to the environmental implications of software systems.

Novelty Statement: This research addresses the current gap in sustainable Software Requirements Engineering (SRE) by providing guidelines for integrating sustainable practices into software development.

Material and Method: An online survey was conducted using self-developed questionnaires designed to gather information on current sustainability practices in Software Requirements Engineering (SRE) among software professionals. The questionnaires, distributed via Google Forms, aimed to capture respondents' perspectives on the relevance of sustainable practices in the field.

Result and Discussion: The findings indicate that active stakeholder engagement, the use of energy-efficient algorithms, and the establishment of continuous improvement procedures are crucial for sustainable Software Requirements Engineering (SRE). Additionally, financial incentives and well-defined criteria for evaluating environmental impact emerged as significant factors. Among the successful practices recommended for integration into software development are audits, training programs, and the adoption of renewable energy practices.

Concluding Remarks: Incorporating sustainability into Software Requirements Engineering (SRE) enhances environmental sustainability and supports organizations' Corporate Social Responsibility (CSR) objectives, positioning them as key contributors to sustainable software engineering.

Keywords: Sustainability; Software Requirement Engineering; Energy Efficiency; Stakeholder Involvement; Environmental Impact.

Introduction:

In light of global environmental challenges and the growing emphasis on corporate responsibility, various sectors worldwide are increasingly prioritizing sustainability as a core value [1]. This paradigm shift is not limited to established areas of the business economy; it has also extended into the innovative realm of software engineering, where sustainability principles aim to mitigate environmental costs while fostering long-term socio-economic benefits. Particularly within the Software Requirements Engineering (SRE) domain, integrating sustainability considerations is emerging as a significant focus for further research and innovation [2].

Software Requirements Engineering is a fundamental process in development that establishes, evaluates, and defines all requirements for project advancement to subsequent phases [3]. Traditionally, primary concerns in SRE have centered on system functionality, performance, and usability. However, the environmental impact of software products—stemming from factors such as energy consumption and the generation of electronic waste—has become increasingly evident [4]. This realization has prompted a reevaluation of SRE practices to better incorporate sustainability from the outset.

Several key concepts relate to sustainable SRE and serve as essential dimensions. First, this transition involves establishing requirements that focus on energy conservation and the efficient use of resources [5]. This entails developing applications that maximize functionality while minimizing computational resource use and are adaptable to various system architectures. Second, sustainable SRE emphasizes minimizing software's environmental impact through effective lifecycle management, and employing energy-efficient processes for testing and deployment [6].

Despite the growing recognition of sustainable SRE practices within contemporary organizations, several barriers hinder their effective implementation. These barriers include a lack of guidelines for assessing the environmental implications of software, the misalignment of sustainability criteria with existing SRE frameworks, and the need for training software engineers in sustainable development practices [3].

This paper seeks to examine the current state of sustainable SRE practices, identify opportunities for enhancing their implementation, and present a conceptual model for integrating sustainability into SRE development. Drawing on a literature review and analysis of case studies, this research offers propositions and recommendations to assist organizations in advancing toward more sustainable software development methodologies.

Objectives:

- To investigate the incorporation of sustainability principles into Software Requirements Engineering (SRE).
- To evaluate current sustainability practices in software development among professionals.
- To identify key factors and effective practices for implementing sustainable SRE.

Literature Review:

The principle of sustainable software development has increasingly become a crucial element of Software Requirement Engineering (SRE) activities, given the environmental and socio-economic implications of creating eco-friendly software products. This literature review emphasizes the essential features and strategies relevant to various aspects of sustainable SRE.

Environmental Impact Considerations:

Power and energy consumption, along with carbon footprint, are critical factors influencing the environmental impact of software systems. The usage of hardware, primarily through data centers and computing devices, plays a significant role in analyzing software's effects on the environment. Servers that support the operation of application programs are among the most power-hungry components of infrastructure. According to the International Energy Agency (IEA), data centers consumed approximately 200 TWh of electricity in 2020, accounting for 1% of the world's total electricity consumption (IEA, 2020). This energy consumption is anticipated to rise alongside increasing data demands, underscoring the need for energy-efficient software solutions.

The authors [7] emphasize that specific energy efficiency criteria should be integrated into the SRE process from the outset to reduce operational energy consumption. This involves selecting energy-efficient algorithms and ensuring that software only computes when necessary. The authors [8] indicate that incorporating features like Dynamic Voltage and Frequency Scaling (DVFS) in green computing can reduce energy consumption by at least 30% in data-intensive applications, thereby lowering the software's carbon footprint.

Stakeholders in Environmental Discussions:

To effectively capture the values that promote environmental conservation within a given society, stakeholders from that society must be involved in software development priorities. Key stakeholders include environmentalists, government departments, residents, and customers. The authors [9] and [10] concur that all participants in software development should engage in roundtable discussions about the meaning and implications of sustainability. Involving external stakeholders, such as environmental organizations and regulatory agencies, can enhance the assessment of the sustainability of developed software requirements and improve organizations' transparency regarding the environmental impacts of their activities.

The authors [11] advocate for a collaborative model of sustainability in software development to improve the systemic achievement of sustainability targets. Engaging with diverse stakeholders is beneficial for assessing potential environmental impacts during software setup and ensuring that sustainability aspects are considered in the requirements.

Prioritization of Energy-Efficient Algorithms and Technologies:

The management of energy and power is recognized as a critical component of sustainable Software Requirement Engineering (SRE). This involves selecting and implementing efficient algorithms and technologies during the requirements gathering and design phases. The authors [12] and [5] emphasize that incorporating energy efficiency criteria into the prioritization of requirements can lead to reduced operating costs and positively impact environmental factors. Specifically, selecting algorithms that require minimal computational power and optimizing code for lower energy consumption can significantly decrease the overall energy usage of software systems.

Criteria for Minimizing Resource Consumption:

Efficiency in memory and processor usage should be one of the fundamental principles of sustainable Software Requirement Engineering (SRE). The authors [13] have presented metrics and guidelines to help identify resource consumption in software requirements, enabling the development of better solutions. When developing virtual software, developers can prioritize resource efficiency, enhancing the overall performance of the final product while minimizing negative environmental impacts associated with resource depletion.

Reducing resource consumption also entails creating software that operates effectively across various hardware platforms, thereby extending the life of older devices and reducing electronic waste. This approach aligns with sustainable development goals by optimizing resource utilization and mitigating the environmental impact of software systems.

Integration of Sustainability into Requirement Prioritization:

Incorporating sustainability criteria into the ranking process ensures that software requirements align with broader environmental objectives. This integration involves applying sustainable measures, such as carbon emissions and resource efficiency. The authors [14] noted that frameworks incorporating these measures can assist developers in selecting requirements that minimize environmental impacts. This approach not only addresses immediate sustainability concerns but also fosters the principles of sustainable management within the software development processes of projects.

Innovative and Efficient Software Solutions:

Emphasizing sustainability principles in software design and development fosters creativity and leads to improved outcomes. The authors [15] and [16] highlighted that addressing sustainability issues can inspire innovative solutions while ensuring that developed software meets environmental standards, resulting in useful and eco-friendly products. For instance, developing algorithms that promote efficient resource and energy usage can significantly reduce costs and environmental impact. Additionally, prioritizing renewable energy sources over non-renewable ones is crucial in software development. [12] have guided sourcing sustainable energy for data centers and software operations to minimize environmental effects, underscoring the need to enhance sustainability practices within software systems.

Guidelines and Standards for Evaluating Environmental Impact:

The authors [9] asserted that for sustainable Software Requirement Engineering (SRE), it is essential to establish clear guidelines for evaluating environmental impact, with adherence to these standards being mandatory. Established frameworks, such as ISO 14001, provide guidelines for conducting environmental audits and improving an organization's environmental performance. These standards emphasize the importance of systematically evaluating and minimizing an organization's environmental effects, enabling the development of sustainable practices in software development. By adhering to these guidelines, developers can align their practices with globally accepted environmental management principles, thereby enhancing their sustainability efforts.

Corporate Social Responsibility (CSR) Goals:

Integrating sustainable software requirements supports Corporate Social Responsibility (CSR) goals, particularly in the realm of environmental responsibility. Research conducted by [16] indicates that aligning the software development process with CSR objectives enhances corporate image and boosts shareholder confidence.

Audits and Reviews for Sustainability Alignment:

Conducting routine audits and reviews of each aspect of the software can help assess its alignment with established sustainability criteria. A study by [17], along with [18], emphasizes the importance of continuous evaluation and refinement processes to enhance the environmental performance of software throughout all stages of development.

Training Programs for Software Engineers:

SRE practices greatly benefit from educational programs that enhance software engineers' knowledge and skills regarding sustainability integration [19]. Training initiatives provided by organizations such as IEEE have offered clear guidelines and frameworks for effectively incorporating ecological assessments into software development processes.

Continuous Improvement Processes:

The application of such systems ensures that the requirements incorporated into the software support sustainability strategies, aligning with emerging trends and technologies. Authors such as [20] and [21] advocate for the use of agile methodologies, which facilitate iterative development, enabling organizations to adapt to evolving environmental standards.

Financial Incentives for Sustainable Requirements Adoption:

Therefore, offering financial incentives encourages organizations to prioritize sustainable software requirements. The authors [22] conclude that adoption is more likely when incentive programs lower initial costs and provide long-term environmental benefits.

Lifecycle Environmental Impacts:

In addition to waste disposal and recycling, assessing the life cycle environmental impact offers a comprehensive approach to evaluation. Research by [23] and [6] introduces methods known as life cycle analysis, which evaluates the environmental impact associated with the various phases of product manufacturing and disposal.

Regulatory Compliance and Energy Efficiency Metrics:

Legal compliance, including adherence to environmental legislation and efficiency standards such as Energy Star ratings, serves as a tool for demonstrating compliance with regulations and comparing organizational sustainability performance. Research by [24] and [25] provides key insights into energy regulation and metrics for assessing software energy efficiency.

Materials and Methods:

A questionnaire-based survey was conducted to identify various sustainability practices integrated into the software requirement engineering process. The questionnaire was developed using Google Forms and consisted of two sections. The first section collected demographic information from respondents, including their names, company names, company locations, company sizes, job roles, and major services provided by their companies. The second section contained 20 questions regarding the incorporation of sustainability practices in the software requirement engineering process. Topics included considerations for environmental impact, stakeholder inclusion, the use of energy-efficient algorithms, audits and reviews for sustainability, training programs to educate employees, financial incentives for sustainability performance, compliance with regulatory standards, and performance metrics for efficient energy utilization. Participants rated their opinions on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree) regarding the importance of integrating different sustainability practices into the software requirement engineering process. Purposive sampling was employed to gather data specifically from individuals associated with the software industry. A total of 108 responses were received, and further details about the respondents are provided in the next section.

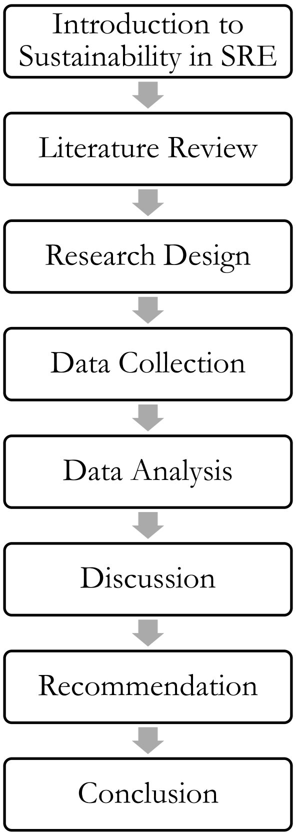

Figure 1. Flow of Study for Integrating Sustainability into Software Requirement Engineering (SRE)

Results:

The data were collected from software firms operating in various regions of Pakistan. Respondents came from a diverse array of backgrounds, including requirements engineers, technical architects, software developers, quality assurance engineers, and project managers. Details regarding the respondents' roles, along with the number of responses from each role, are provided below:

Table 1. Respondent Designation

Designation	No. of Responses
Requirement Engineer	6
Technical Architect	5
Software Developer	41
Quality Assurance Engineer	28
Project Manager	8
Others	20

The respondents were inquired about their relevant industry experience, which, as illustrated in the table below, varies from 1 year to over 10 years. Additionally, participants were asked to provide information about their companies and the types of services they offer. The results indicate that company sizes, in terms of the number of employees, range from micro (fewer than 10 employees) to large (more than 250 employees). Furthermore, the services provided by these companies include web development, mobile application development, cloud-based solutions, embedded system design, and API development.

Table 2. Respondent Experience

Experience	No. of Responses
Less than 1 year	52
1 to 5 years	36
5 to 10 years	16
More than 10 years	4

Table 3. Company size

Company Size	No. of Responses
Micro (<10 employees)	24
Small (10 to 50 employees)	23
Medium (50 to 250 employees)	26
Large (> 250 employees)	35

Table 4. Services provided by company

Web development
Mobile application development
Cloud application development
Embedded system development
API development

Section 2 of the questionnaire focused on gathering respondents' perspectives regarding the integration of sustainability principles into the software requirement engineering process. Participants were asked to rank various practices based on their importance for sustainability. The top 10 practices recommended for incorporation into the software engineering process are presented in the table below:

Table 5. Sustainability practices in software requirement engineering process

Sustainability Practices	Rank
Stakeholders should be actively involved in discussions regarding the environmental benefits of proposed software requirements.	1
Energy-efficient algorithms and technologies should be prioritized when designing software requirements.	2
Continuous improvement processes should be implemented to enhance the sustainability aspects of software requirements over time	3
Financial incentives (e.g., cost savings from energy-efficient software) should be provided to encourage the adoption of sustainable software requirements	4
There should be clear guidelines and standards in place for evaluating the environmental impact of software requirements.	5
Software requirements should include criteria to minimize resource consumption such as memory and processing power.	6
Regular audits and reviews should be conducted to assess how well software requirements align with sustainability goals.	7
Training programs should be implemented to educate software engineers and developers about integrating sustainability into their requirement engineering practices.	8
Software requirements should prioritize the use of renewable energy sources and minimize reliance on non-renewable resources.	9
Performance metrics related to energy efficiency and environmental impact should be regularly monitored and reported for software requirements	10

Discussion:

The sustainability of software requirements engineering practices, which encompass the elicitation, documentation, analysis, and management of software requirements, has gained significant attention in recent years. This practice ensures that software development and the resulting products enhance environmental sustainability, social welfare, and economic development. Engaging stakeholders in discussions about environmental benefits makes software development more responsive to the core principles of sustainability [26]. Identifying features that minimize adverse environmental impacts while meeting stakeholders' expectations during the requirement's engineering stage is more effective. Such an approach fosters trust among stakeholders, increasing the likelihood of adopting and implementing sustainable software solutions.

Energy consumption of algorithms and technologies is a critical factor that should be prioritized to enhance the operational efficiency and environmental impact of software systems [27][28]. During the requirements engineering phase, energy efficiency must be prioritized, as reducing environmental harm enables organizations to achieve better efficiency and lower costs. It is essential to apply various strategic improvement initiatives to drive the progressive development of requirements, thereby consistently enhancing the sustainability of software solutions. By conducting cyclical assessments of environmental performance indicators, organizations can effectively manage emerging threats, optimize resource usage, and demonstrate corporate social responsibility aligned with standard sustainability goals.

The use of incentives encourages changes that align with socially responsible standards, motivating organizations to adopt sustainable software requirements to qualify for these incentives. Effectively presenting the economic benefits of utilizing energy-saving software solutions, such as through cost-effectiveness and ROI, can help organizations justify sustainable measures while improving efficiency and environmental management [29]. Standard procedures guide the assessment of the environmental effects associated with creating sustainable software products. ISO standards systematically identify energy use, carbon emissions, and resource consumption throughout the software's lifecycle, from conception and design to implementation, testing, maintenance, and disposal. This structured approach simplifies compliance with statutory regulations and guidelines, builds stakeholder trust, and facilitates informed decision-making.

Incorporating criteria for resource conservation within the software requirements supports sustainable software development and operation. By establishing well-defined objectives for the efficient use of available resources, organizations can control implementation costs, enhance equipment longevity, and mitigate environmental strain [30]. Consequently, the implementation of sustainable software requirements should be periodically audited and reviewed to assess its effectiveness and alignment with the organization's sustainability policies. Evaluating environmental performance indicators and the organization’s adherence to sustainability benchmarks enables the identification of improvement opportunities, risk management, and resource usage optimization [31].

Implementing training programs equips software engineers and developers with the necessary knowledge to incorporate sustainability into requirements engineering. By investing in employee education, organizations cultivate a culture of innovation, teamwork, and lifelong learning, leading to sustained returns in software development initiatives [32]. According to [33], prioritizing renewable energy sources when establishing software development goals ensures alignment between technological and environmental timelines. Incorporating renewable energy specifications into software development and data center architecture reduces greenhouse gas emissions, enhances energy security, and demonstrates corporate social responsibility through the delivery of green technological solutions [33].

Regular assessment and communication of performance indicators related to energy conservation and environmental impacts are essential for evaluating compliance with sustainable software requirements. This enables clear visibility into achieved gains, identifies areas for improvement, and facilitates reporting results to stakeholders through well-defined KPIs and measurement methodologies [34].

Conclusion:

Incorporating sustainability principles into software requirement engineering represents a significant advancement in addressing environmental, social, and economic issues within the digital realm. By fostering engagement, emphasizing technologies that minimize energy consumption, promoting continuous improvements, and prioritizing stakeholder involvement alongside financial incentives, the sustainability aspects of software requirements can be significantly enhanced. These practices encompass the clear specification of guidelines, resource consumption criteria, periodic audits, and staff training, all of which bolster sustainability initiatives throughout every stage of software development.

Additionally, focusing on renewable energy sources and continuously monitoring the effectiveness of various parameters demonstrates a company's commitment to environmental conservation and the promotion of sustainable technologies. As the reliance on software grows within modern economies and societies, adhering to such practices not only ensures compliance with relevant legislation and market standards but also signifies an organization's readiness to operate responsibly and sustainably while producing high-quality software. Therefore, it is essential to build on this knowledge and emphasize cross-disciplinary collaboration to extend and mainstream sustainable software requirement engineering, ultimately contributing to shared climate goals in our increasingly interconnected world.

References:

1. Tseng, M.-L., Tran, T. P. T., Ha, H. M., Bui, T.-D., & Lim, M. K. (2021). Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: A data driven analysis. Journal of Industrial and Production Engineering, 38(8), 581–598.
2. Duboc, L., Penzenstadler, B., Porras, J., Akinli Kocak, S., Betz, S., Chitchyan, R., Leifler, O., Seyff, N., & Venters, C. C. (2020). Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow. Requirements Engineering, 25, 469–492.
3. Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. Journal of Systems and Software, 172, 110851.
4. Gupta, U., Kim, Y. G., Lee, S., Tse, J., Lee, H.-H. S., Wei, G.-Y., Brooks, D., & Wu, C.-J. (2021). Chasing carbon: The elusive environmental footprint of computing. 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 854–867.
5. Reya, N. F., Ahmed, A., Zaman, T., & Islam, M. M. (2023). GreenPy: evaluating application-level energy efficiency in Python for green computing. Annals of Emerging Technologies in Computing.
6. Itten, R., Hischier, R., Andrae, A. S. G., Bieser, J. C. T., Cabernard, L., Falke, A., Ferreboeuf, H., Hilty, L. M., Keller, R. L., & Lees-Perasso, E. (2020). Digital transformation—life cycle assessment of digital services, multifunctional devices and cloud computing. The International Journal of Life Cycle Assessment, 25, 2093–2098.
7. Gupta, U., Elgamal, M., Hills, G., Wei, G.-Y., Lee, H.-H. S., Brooks, D., & Wu, C.-J. (2022). ACT: Designing sustainable computer systems with an architectural carbon modeling tool. Proceedings of the 49th Annual International Symposium on Computer Architecture, 784–799.
8. Bharany, S., Sharma, S., Khalaf, O. I., Abdulsahib, G. M., Al Humaimeedy, A. S., Aldhyani, T. H. H., Maashi, M., & Alkahtani, H. (2022). A systematic survey on energy-efficient techniques in sustainable cloud computing. Sustainability, 14(10), 6256.
9. Bendtsen, E. B., Clausen, L. P. W., & Hansen, S. F. (2021). A review of the state-of-the-art for stakeholder analysis with regard to environmental management and regulation. Journal of Environmental Management, 279, 111773.
10. Hidellaarachchi, D., Grundy, J., Hoda, R., & Mueller, I. (2023). The influence of human aspects on requirements engineering-related activities: Software practitioners' perspective. ACM Transactions on Software Engineering and Methodology, 32(5), 1–37.
11. Kolpondinos, M. Z., & Glinz, M. (2020). GARUSO: a gamification approach for involving stakeholders outside organizational reach in requirements engineering. Requirements Engineering, 25(2), 185–212.
12. Katal, A., Dahiya, S., & Choudhury, T. (2023). Energy efficiency in cloud computing data centers: a survey on software technologies. Cluster Computing, 26(3), 1845–1875.
13. Chofreh, A. G., Goni, F. A., Klemeš, J. J., Malik, M. N., & Khan, H. H. (2020). Development of guidelines for the implementation of sustainable enterprise resource planning systems. Journal of Cleaner Production, 244, 118655.
14. Fuchs, C. (2020). Information Technology and Sustainability in the Information Society. 11(688768), 2431–2461.
15. Wankhede, V. A., Agrawal, R., Kumar, A., Luthra, S., Pamucar, D., & Stević, Ž. (2024). Artificial intelligence an enabler for sustainable engineering decision-making in uncertain environment: a review and future propositions. Journal of Global Operations and Strategic Sourcing, 17(2), 384–401.

16. Semerikov, S. O., Striuk, A. M., Striuk, L., Striuk, M., & Shalatska, H. M. (2020). Sustainability in Software Engineering Education: a case of general professional competencies. E3S Web of Conferences. The International Conference on Sustainable Futures: Environmental, Technological, Social and Economic Matters (ICSF 2020), 166, 1–13.
17. Poth, A., & Nunweiler, E. (2022). Develop sustainable software with a lean ISO 14001 setup facilitated by the efiS® framework. International Conference on Lean and Agile Software Development, 96–115.
18. Muravev, M., Golovkov, A., Shershen, K., Somenkova, A., & Brazhenko, D. (2023). HYBRID SOFTWARE DEVELOPMENT METHODS: EVOLUTION AND THE CHALLENGE OF INFORMATION SYSTEMS AUDITING. Journal of the Balkan Tribological Association, 29(4).
19. Duarte, A., & Krajsic, A. V. (2017). Global Economy Iss European Journal of Training and Development Journal of Consumer Marketing Iss. Journal of Enterprising Communities: People and Places in the Global Economy, 9(2310), 94–113. http://dx.doi.org/10.1108/JEC-09-2014-0018%5Cnhttp://dx.doi.org/10.1108/EJTD-02-2013-0019%5Cnhttp://
20. Bambazek, P., Groher, I., & Seyff, N. (2023). Requirements engineering for sustainable software systems: a systematic mapping study. Requirements Engineering, 28(3), 481–505.
21. Galán, O. A. A., Valdéz, J. L. C., Medina, H. F., Contreras, G. A. V., & Sumuano, J. L. S. (2020). Proposal of a sustainable agile model for software development. International Journal of Advanced Computer Science and Applications, 11(1).
22. Klewitz, J., & Hansen, E. G. (2014). Sustainability-oriented innovation of SMEs: A systematic review. Journal of Cleaner Production, 65, 57–75. https://doi.org/10.1016/j.jclepro.2013.07.017
23. Verdecchia, R., Lago, P., Ebert, C., & De Vries, C. (2021). Green IT and green software. IEEE Software, 38(6), 7–15.
24. Gährs, S., & Knoefel, J. (2020). Stakeholder demands and regulatory framework for community energy storage with a focus on Germany. Energy Policy, 144, 111678.
25. Pargaonkar, S. (2023). Synergizing Requirements Engineering and Quality Assurance: A Comprehensive Exploration in Software Quality Engineering. International Journal of Science and Research (IJSR), 12(8), 2003–2007.
26. Attanasio, G., Preghenella, N., De Toni, A. F., & Battistella, C. (2021). Stakeholder engagement in business models for sustainability: The stakeholder value flow model for sustainable development. Business Strategy and the Environment, 31(3), 860–874. https://doi.org/10.1002/bse.2922
27. Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., … Akyildiz, B. (2022). Sustainable AI: Environmental Implications, Challenges and Opportunities. Proceedings of Machine Learning and Systems, 4, 795–813. Retrieved from https://proceedings.mlsys.org/paper_files/paper/2022/hash/462211f67c7d858f663355eff93b745e-Abstract.html
28. Bradu, P., Biswas, A., Nair, C., Sreevalsakumar, S., Patil, M., Kannampuzha, S., … Gopalakrishnan, A. V. (2022). Recent advances in green technology and Industrial Revolution 4.0 for a sustainable future. Environmental Science and Pollution Research International, 30. https://doi.org/10.1007/s11356-022-20024-4
29. None Akoh Atadoga, Umoga, J., Lottu, A., & Oluwademilade, E. (2024). Tools, techniques, and trends in sustainable software engineering: A critical review of current practices and future directions. World Journal of Advanced Engineering Technology and Sciences, 11(1), 231–239. https://doi.org/10.30574/wjaets.2024.11.1.0051

30. Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2020). Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. Journal of Systems and Software, 172, 110736. https://doi.org/10.1016/j.jss.2020.110736
31. Grecu, V., Ciobotea, R.-I.-G., & Florea, A. (2020). Software Application for Organizational Sustainability Performance Assessment. Sustainability, 12(11), 4435. https://doi.org/10.3390/su12114435
32. Peláez, R. M., Brust, A. O., Rivera, S., Félix, V. G., Ostos, R., Brito, H., … Mena, L. J. (2023). Role of Digital Transformation for Achieving Sustainability: Mediated Role of Stakeholders, Key Capabilities, and Technology. Sustainability, 15(14), 11221. mdpi. https://doi.org/10.3390/su151411221
33. Pazienza, A., Baselli, G., Daniele Carlo Vinci, & Maria Vittoria Trussoni. (2024). A holistic approach to environmentally sustainable computing. Innovations in Systems and Software Engineering. https://doi.org/10.1007/s11334-023-00548-9
34. Fattahi, A., Sijm, J., & Faaij, A. (2020). A systemic approach to analyze integrated energy system modeling tools: A review of national models. Renewable and Sustainable Energy Reviews, 133, 110195. https://doi.org/10.1016/j.rser.2020.110195