Journal of Economics, Finance and Management Studies

ISSN (print): 2644-0490, ISSN (online): 2644-0504

Volume 09 Issue 03 March 2026

Article DOI: 10.47191/jefms/v9-i3-37, Impact Factor: 8.735

Page No: 1550-1558

From Ambition to Execution: A Framework for Integrating SDGs in Software Engineering Project Management

Xiaoli Liu [1], Abu bakar Abdul hamid[2], Kamaljeet Kaur[3]

[1,2,3]Faculty of Business, Information & Human Sciences, Kuala Lumpur University of Science and Technology, Kuala Lumpur, Malaysia

ABSTRACT: The imperative to achieve the United Nations' Sustainable Development Goals (SDGs) is increasingly permeating all industries, including the technology sector. Software engineering, as a critical enabler of global digitalization, holds significant potential to contribute to, but also detract from, sustainability objectives. However, a significant research gap exists in the form of a comprehensive, integrated theoretical framework to guide the systematic embedding of SDGs into the software engineering project management lifecycle. This paper adopts a constructivist approach, utilizing a comprehensive desktop literature review as its research method to build, rather than test, theory. By integrating three core theoretical pillars—Project Governance, Sustainable Software Engineering, and established development methodologies (Agile/Waterfall)—this paper constructs an integrative theoretical framework. This framework delineates the logic for embedding SDGs across the entire software project lifecycle, from initiation and planning to development, delivery, and operations. It clarifies the core elements, embedding logic, and adaptive conditions at each stage. The primary contribution of this research is a novel, actionable theoretical framework, complemented by an "Industry Applicability Matrix" that outlines differentiated adaptation strategies for various organizational contexts, such as large enterprises versus start-ups and projects utilizing Agile versus Waterfall methodologies. This study provides a vital theoretical foundation and practical guidance for project managers, technology organizations, and standard-setting bodies, aiming to bridge the gap between high-level sustainability ambitions and concrete project-level execution in the software industry.

KEYWORDS: Sustainable Development Goals (SDGs); Software Engineering; Project Management; Sustainability; Project Governance; Theoretical Framework

I. INTRODUCTION

The global community is at a critical juncture, facing complex and interconnected challenges ranging from climate change and environmental degradation to social inequality and economic instability. In response, the United Nations' 2030 Agenda for Sustainable Development, with its 17 Sustainable Development Goals (SDGs), provides a universal "shared blueprint for peace and prosperity for people and the planet" (United Nations, 2015/2025, para. 1). As digitalization accelerates, the software engineering industry has emerged as a powerful force with a dual role in this agenda. On one hand, software systems are instrumental in driving progress toward the SDGs, enabling innovations in renewable energy, healthcare access, education, and transparent governance (Calero et al., 2022). On the other hand, the information and communications technology (ICT) sector, including its vast software and data center infrastructure, is a significant consumer of energy and resources (International Telecommunication Union [ITU], 2024), with a carbon footprint comparable to that of the aviation industry (Ericsson, 2025; ITU, 2024). This duality creates an urgent imperative for the software engineering field to transform its practices to align with sustainability principles (Moreira et al., 2025).

While concepts like Green Software Engineering (GSE) and sustainable software development have gained traction, focusing primarily on the technical dimensions of energy efficiency and resource optimization (Calero & Piattini, 2017; Funke & Lago, 2026; Kern et al., 2013; Pathania et al., 2025), a concurrent rise in exploring the integration of sustainability within project management practices has occurred (Armenia et al., 2019; Green Project Management, 2025; Silvius & Schipper, 2014). However, existing research tends to address these two domains in isolation. Studies on GSE predominantly focus on

development and maintenance phases while leaving early-stage processes largely unexplored (Reis et al., 2025). Consequently, there is a dearth of work that holistically integrates the multi-dimensional aspirations of the SDGs, which span environmental, social, and economic pillars, into the structured lifecycle of software engineering project management (Fesenko et al., 2024; Soares et al., 2024).

This disconnect constitutes a significant research gap. Recent literature reviews explicitly state that a comprehensive framework to guide the integration of sustainability into software engineering project management is conspicuously absent (Huq, 2025; Sokka, 2024). Scholars also point to a chasm between the theoretical understanding of sustainability and its practical application in software engineering (König et al., 2025), as well as a disconnect between organizational-level SDG commitments and project-level execution (Reis et al., 2025; Sokka, 2024). This lack of an integrated theoretical structure leaves project managers and organizations without a clear, actionable roadmap, resulting in ad hoc, inconsistent, and often superficial efforts to align software projects with sustainability goals (Gurung et al., 2026; Reis et al., 2025). In the absence of a guiding framework, the logic for embedding SDGs remains unclear, the practical dilemmas are not systematically addressed, and appropriate evaluation metrics are lacking (Gurung et al., 2026; Sokka, 2024).

In light of this gap, this paper addresses the following open-ended, constructivist research question: *How can the Sustainable Development Goals be systematically and effectively embedded into the full lifecycle of software engineering project management?* Rather than proposing and testing variable-based hypotheses, this study seeks to build a new theoretical framework from existing knowledge. It aims to develop a holistic, integrative framework that synthesizes disparate streams of theory to provide a structured, logical, and practical pathway for action. By doing so, this paper seeks to make a significant theoretical contribution while offering tangible value to practitioners striving to make software engineering a more potent force for global sustainable development.

II. LITERATURE REVIEW

A. Conceptual Review

A review of the literature reveals a growing but fragmented body of knowledge at the intersection of sustainability and software engineering. A common understanding of the core concepts is foundational to any integrative effort. The Sustainable Development Goals (SDGs) are frequently cited as the primary macro-level driver for incorporating sustainability into business strategy (Sachs et al., 2022). In the context of software, the literature acknowledges their dual nature: software as an enabler for achieving SDGs, and the software industry's own responsibility to operate sustainably (Calero et al., 2022). However, many studies focus on specific SDG-related applications rather than the holistic integration of the goals themselves into the development process.

Parallel to the rise of the SDGs, Sustainable Software Engineering (SSE) has emerged as a distinct field of study. It is broadly defined as the discipline of creating and maintaining software systems whose direct and indirect environmental, social, and economic impacts are minimal and have a positive overall effect on sustainability (Naumann et al., 2011). This concept expands on the narrower notion of Green IT, which is primarily focused on energy efficiency (Calero & Piattini, 2015). The literature on SSE is heavily skewed towards the environmental dimension, often termed the "green" aspect, with extensive research on topics such as energy-efficient coding practices, green software architectures, and carbon-aware computing (Freed et al., 2023; König et al., 2025). In contrast, the social and economic dimensions, such as ethical AI, digital accessibility, and managing technical debt for long-term viability, are acknowledged but less explored in an integrated manner (Fatima & Lago, 2025).

The Software Engineering Project Management Lifecycle provides the process context for this integration. While various models exist, the literature consistently identifies core phases such as requirements engineering, design, implementation, testing, deployment, and maintenance (Project Management Institute [PMI], 2021; Reis et al., 2025). A challenge highlighted in recent reviews is that sustainability considerations are often treated as an afterthought or are confined to a single phase, rather than being integrated throughout the entire lifecycle (Reis et al., 2025; Rinne, 2024). For instance, although some research has focused on requirements engineering for sustainability (Duboc et al., 2020), a lifecycle-wide perspective remains less developed, with early-stage activities like requirements elicitation and prototyping being largely unexplored in current practice (Duboc et al., 2020; Reis et al., 2025; Rinne, 2024). Recent efforts have begun to address this gap by proposing frameworks that embed sustainability into Agile workflows, yet empirical validation and holistic integration across all lifecycle phases remain ongoing challenges (Huq, 2025; Rinne, 2024).

B. Theoretical Review

This section synthesizes three theoretical pillars to ground an integrated logic for embedding SDGs across the software project lifecycle: Project Governance, Sustainable Software Engineering (SSE), and established development methodologies.

The framework's foundational theoretical pillar is Project Governance, which establishes the rationale and accountability for sustainability integration. Project governance defines the structures and processes through which projects are directed and controlled to achieve strategic objectives (Too & Weaver, 2014). For sustainability integration, governance must articulate purpose, assign accountability, and embed decision rights for environmental and social outcomes alongside traditional cost, time, and scope targets. Global guidance codified in ISO 37000:2021 emphasizes purpose, value generation, stewardship, and oversight as governance principles, explicitly linking effective governance to sustainable performance (International Organization for Standardization [ISO], 2021). Furthermore, empirical and scientometric work on sustainable project governance demonstrates the importance of charter-level commitments, portfolio controls, and role accountability for sustainability outcomes (Zhang et al., 2023). This is complemented by stakeholder governance, which provides the normative rationale for integrating multi-stakeholder expectations and aligning SDG targets with stakeholder salience in project decisions (Freeman, 2010). Together, these strands imply that SDG objectives must be specified and governed at a project's inception and monitored via governance mechanisms throughout its lifecycle (Too & Weaver, 2014; Zhang et al., 2023).

Providing the substantive technical content for implementation is the next pillar, Sustainable Software Engineering (SSE). This body of theory extends traditional notions of software quality to include environmental, social, and economic sustainability as first-class attributes, which are operationalized as non-functional requirements, architectural tactics, and engineering practices (Calero & Piattini, 2015; Kern et al., 2013; König et al., 2025; Naumann et al., 2011). Recent systematic syntheses consolidate concrete design and implementation tactics, such as energy-efficient algorithms and carbon-aware architectures, and map them to measurable outcomes (Danushi et al., 2025). Crucially, measurement standards have matured; for instance, ISO/IEC 21031:2024 ("Software Carbon Intensity") specifies a methodology to quantify carbon emissions attributable to software systems, supporting goal-setting and comparability across projects (International Organization for Standardization/International Electrotechnical Commission [ISO/IEC], 2024). SSE theory thus enables the translation of high-level SDG aims, such as "Climate Action" (SDG 13), into verifiable technical requirements and traceable metrics (Danushi et al., 2025; ISO/IEC, 2024).

The procedural structure for implementation is provided by established Development Methodologies, such as Agile and Waterfall. These methodologies offer the scaffolding within which SSE practices are executed and governed. In plan-driven lifecycles, sustainability requirements are specified comprehensively upfront and validated at stage gates, consistent with PMBOK® principles (PMI, 2021). In Agile contexts, sustainability concerns are expressed as backlog items, acceptance criteria, and elements of the "Definition of Done," to be refined iteratively through ceremonies like Sprint Planning and Retrospectives. Evidence from hybrid/agile studies and tailored processes like SustainScrum shows viable patterns for institutionalizing sustainability within Agile workflows, including sustainability-aware backlog management and portfolio-level tracking (Huq, 2025; Leong et al., 2023; Mazak-Huemer et al., 2025; Sokka, 2024). This perspective positions methodology as an adaptive vehicle that sequences how sustainability work is planned, implemented, and assessed.

Integrating these three pillars yields a cohesive theoretical stance where governance provides direction, SSE provides technical content, and methodology orchestrates execution across the lifecycle. This interlock is the foundation for the framework developed in Section 4.

C. Empirical Review

Empirical evidence on the integration of SDGs in software engineering projects is emerging, primarily through case studies and qualitative surveys. A major recurring challenge identified in this body of work is the lack of practical guidelines and standardized metrics (Khalifeh et al., 2020). While organizations express a high level of commitment to SDGs at the corporate level, this often fails to translate into concrete project-level actions.

Nevertheless, these studies have identified several success factors. Strong leadership commitment and the creation of a supportive organizational culture are paramount (Ahmad & Al Baik, 2025; Piwowar Sulej & Iqbal, 2025). This includes providing training to developers, appointing "sustainability champions" within teams (Bambazek et al., 2022), and integrating sustainability metrics into performance evaluations. The adoption of Agile methodologies also appears to be a positive factor, as the iterative nature of Agile allows for emergent sustainability requirements to be incorporated and for teams to learn and adapt (Leong et al., 2023). For example, a development team might iteratively refine the energy consumption of a feature over several sprints.

Recent case studies, though limited, provide concrete examples of these principles in action. Research on large tech companies like Microsoft and SAP shows how they are building dedicated tools to provide project teams with the data needed to track their environmental impact (Jiménez et al., 2024). Other studies highlight the success of integrating sustainability when there is a clear business case, such as cost savings from reduced energy consumption or enhanced brand reputation. However, a

significant gap remains in empirical research on the social dimension of SSE, with most studies focusing on environmental metrics (Fatima & Lago, 2025; Ribeiro et al., 2024).

III. RESEARCH METHODOLOGY

This study employs a "review-based theoretical construction" methodology, structured around three analytical pillars, to synthesize existing literature into a novel framework. The process began with a Conceptual Review to define and delineate core concepts, including the Sustainable Development Goals (SDGs), sustainable software engineering, and the project management lifecycle, which established a unified analytical foundation. Subsequently, a Theoretical Review analyzed and integrated key principles from Project Governance theory, Sustainable Software Engineering theory, and established development methodologies. An Empirical Review then synthesized practical insights from industry standards, corporate sustainability reports, and real-world case studies to ground the framework in current practice. By systematically integrating insights from these three pillars, this research moves beyond a simple literature summary to actively construct new theory. This integrative methodology facilitated the identification of core principles, integration points, and practical challenges, the synthesis of which forms the basis for the novel, multi-layered theoretical framework presented below.

IV. THE CORE ANALYSIS: BUILDING AN INTEGRATIVE FRAMEWORK

This section constitutes the core of the paper, moving beyond a review of extant literature to the active construction of a new theoretical framework. Following the outlined methodology, the construction process is layered, starting with foundational concept definitions, moving to the integration of core theories, and culminating in the presentation of the final framework and its practical application guide.

A. The Conceptual Layer: Defining the Building Blocks

A clear and unified understanding of the framework's core concepts is essential before they can be integrated. A primary concept is the Sustainable Development Goals (SDGs), a set of 17 universal, interdependent goals addressing global challenges. For the purpose of this framework, it is crucial to move beyond viewing the SDGs as a monolithic block. Instead, they can be disaggregated into three dimensions: Environmental (e.g., SDG 13: Climate Action), Social (e.g., SDG 10: Reduced Inequalities), and Economic (e.g., SDG 8: Decent Work and Economic Growth). Software projects can have impacts across all three dimensions, and this tripartite view is essential for a holistic integration (Sachs et al., 2022; United Nations, 2015/2025).

Another key concept, Sustainable Software Engineering (SSE), is an emerging discipline focused on creating software in a sustainable manner. It expands upon traditional software engineering concerns to include environmental, social, and economic impacts as first-class quality attributes (Calero & Piattini, 2015; Danushi et al., 2025; König et al., 2025; Naumann et al., 2011). Key principles of SSE include carbon efficiency, energy efficiency, and carbon awareness, with measurement approaches codified in standards like ISO/IEC 21031:2024 (Green Software Foundation, 2022, 2024; ISO/IEC, 2024). A holistic view of SSE also encompasses social goals, such as accessibility guided by standards like WCAG 2.2, and economic sustainability, such as minimizing technical debt (W3C, 2023).

The final conceptual building block is the Software Engineering Project Management Lifecycle, which provides the structure for embedding these goals. To systematically embed SDGs, they must be anchored to the distinct phases of a project. While specific terminology varies, a generic lifecycle can be conceptualized in four primary phases (PMI, 2021). The lifecycle begins with Initiation and Planning, which involves defining the project's vision and goals and where strategic alignment with SDGs must be established. This is followed by the Development phase, the core construction stage where the software is designed, coded, and tested, and where SSE principles are technically implemented. The third phase, Delivery and Deployment, involves releasing the software to users, with key sustainability considerations including the environmental impact of infrastructure and the social impact of the user rollout. Ultimately, the Operations and Maintenance phase focuses on the ongoing running, monitoring, and updating of the software, where the long-term energy consumption and responsible decommissioning are central to sustainability.

B. The Theoretical Layer: Integrating Core Theories

Having defined the building blocks, this section integrates the three core theoretical domains discussed earlier to form the framework's structural backbone. These theories are not simply listed but are interwoven to provide a multi-faceted and robust logic for SDG integration. Project Governance Theory serves as the starting point, establishing the top-down authority and accountability structures necessary for SDG integration by ensuring alignment with strategic objectives (Too & Weaver, 2014; Zhang et al., 2023). It reinforces principles like stewardship and value generation (ISO, 2021) and mandates that sustainability

requirements be formally embedded in project charters and stakeholder engagement processes (Feng et al., 2022; Freeman, 2010). Complementing this strategic direction, Sustainable Software Engineering (SSE) Theory provides the technical substance for the framework. It offers concrete principles and practices, from energy-efficient algorithms to carbon footprint measurement, that allow high-level SDG goals to be translated into tangible, non-functional requirements (Danushi et al., 2025; ISO/IEC, 2024; Kern et al., 2013; König et al., 2025). The integration is completed through the lens of Development Methodologies, which provide the procedural structure for implementation, whether through upfront requirements definition in a Waterfall model or through iterative refinement in an Agile context (Huq, 2025; Leong et al., 2023; Mazak-Huemer et al., 2025; Sokka, 2024; PMI, 2021). By weaving these three theoretical threads together, we can construct a framework that is strategically grounded, technically robust, and methodologically adaptable.

C. The Integrated Theoretical Framework

This section presents the primary contribution of this paper: the Integrative Framework for SDG Embedding in Software Engineering Project Management. The framework is organized around the four project lifecycle phases, with each phase characterized by a distinct embedding logic, a set of core activities, and a key output.

The framework commences with the Initiation and Planning phase, which focuses on strategic alignment and governance. The central logic during this stage is to align the project's purpose with specific SDG targets, an activity dominated by project governance principles (Too & Weaver, 2014; Zhang et al., 2023). Core activities include conducting an SDG materiality assessment to identify relevant goals, articulating the value of this alignment in a sustainability business case, and formally embedding these responsibilities into the project charter consistent with ISO 37000 principles (ISO, 2021; Sachs et al., 2022; Silvius & Schipper, 2014). This also involves defining high-level Key Performance Indicators (KPIs) based on the selected SDGs, using standardized measurement where possible, such as ISO/IEC 21031 (ISO/IEC, 2024; W3C, 2023). The culmination of this phase is a Project Charter that contains explicit SDG targets and measurable KPIs.

From strategic intent, the framework progresses to the Development phase, which centers on technical implementation and methodological adaptation. Here, the logic is to translate the strategic SDG targets from the planning phase into technical requirements, using SSE principles within the chosen development methodology (Danushi et al., 2025; Kern et al., 2013; König et al., 2025; Leong et al., 2023; Mazak-Huemer et al., 2025). In a Waterfall context, this involves defining detailed sustainability non-functional requirements in the design phase and validating them at stage gates (PMI, 2021). In an Agile environment, activities include creating sustainability-focused user stories for the backlog and incorporating sustainability criteria into the "Definition of Done," with iterative measurement of metrics like carbon intensity (ISO/IEC, 2024). The key output of this phase is developed software that incorporates sustainability features backed by verifiable metrics.

The subsequent phase is Delivery and Deployment, where the focus shifts to sustainable infrastructure and user empowerment. The underlying logic is to ensure that the deployment infrastructure is environmentally sustainable and that the software rollout is socially responsible. Core activities include green deployment practices, such as selecting low-carbon cloud regions and configuring energy-efficient CI/CD pipelines (Cao et al., 2022; Sarkar et al., 2024). This phase also involves an ethical rollout, ensuring that training materials are accessible and that users are clearly informed about the product's sustainability features (W3C, 2023). Moreover, sustainability checks must be incorporated into release criteria and pipeline gates (Alamer & Alharbi, 2025; Nayak et al., 2024). This phase results in software deployed on a low-impact infrastructure.

The framework concludes with the ongoing Operations and Maintenance phase, which is concerned with lifecycle optimization and reporting. The embedding logic here is to manage the software's long-term environmental and social footprint while reporting on performance against the governance KPIs established in the first phase. Key activities include sustainable operations, such as implementing auto-scaling and continuously monitoring the application's energy consumption in production (Alharthi et al., 2024; Kaur et al., 2024). It also encompasses planning for responsible hardware decommissioning and e-waste disposal. A crucial activity in this phase is continuous sustainability reporting, where KPIs are tracked and progress is communicated to stakeholders, often in harmony with corporate ESG disclosures (Green Project Management, 2025). The outputs of this ongoing phase are a continuously optimized system and regular sustainability performance reports.

D. The Practical Layer: Industry Applicability Matrix

To bridge the gap between theory and practice, the framework's application must be adapted to different organizational contexts. The Industry Applicability Matrix, presented in Table 1, offers a practical guide for this adaptation by outlining differentiated strategies for distinct organizational environments and methodologies.

Table 1. Industry Applicability Matrix

Context Dimension	Large Enterprise (e.g., Microsoft, SAP)	Startup / SME
Framework Focus	Formal Governance & Reporting: Emphasis on Phase 1 (Governance) and Phase 4 (Reporting). Leverage established PMOs and link to corporate ESG reporting (Jiménez et al., 2024).	Agile Implementation & Impact: Emphasis on Phase 2 (Development). Governance is lighter, often driven by founder values.
Resource Allocation	Dedicated Roles & Budgets: Can afford dedicated "Sustainability Champions" and specialized tools.	Shared Responsibility & Lean Tools: Sustainability is a shared responsibility. Rely on open-source tools. Business case must be tightly linked to core metrics (e.g., reduced cloud bills).
Methodology (Waterfall)	Robust Upfront Design: Significant investment in defining sustainability requirements upfront with formal checkpoints (PMI, 2021).	Generally Avoided: If used, the scope is smaller and requirements are less formal.
Methodology (Agile)	Scaled Agile with Formalisms: May use frameworks like SAFe. Sustainability stories are formally estimated and tracked for portfolio level reporting (Leong et al., 2023; Mazak-Huemer et al., 2025).	Core to the Backlog: Sustainability is baked into the product backlog. The build-measure-learn loop is used to iteratively improve impact (Leong et al., 2023; Sokka, 2024).

V. DISCUSSION: CORE ARGUMENTS AND PRACTICAL IMPLICATIONS

This study's construction of an integrative theoretical framework leads to a core argument: the effective embedding of SDGs into software engineering project management requires a holistic approach that simultaneously operates at the strategic (governance), technical (engineering), and procedural (methodology) levels. A singular focus on "green code," agile processes, or top-down mandates is insufficient. Sustainable outcomes are an emergent property of the systematic interaction between these three pillars across the project lifecycle (Danushi et al., 2025; PMI, 2021; Silvius & Schipper, 2014). The framework thus demonstrates that SDG integration is not a one-off task but a continuous process of alignment, implementation, measurement, and adaptation.

Furthermore, the analysis reveals that a crucial prerequisite for success is the translation of abstract SDG ambitions into concrete, measurable, and context-relevant project KPIs. Without this translation, SDG alignment remains a superficial branding exercise. The challenge for the industry lies not in a lack of awareness, but in the lack of an operationalized and integrated structure to bridge the gap from global goals to lines of code (ISO/IEC, 2024; Sachs et al., 2022; United Nations, 2015/2025).

The practical implications of this framework are significant and can be targeted at three key stakeholders. For Project Managers, this framework provides a direct, actionable roadmap. The primary implication is the need to expand the traditional project management triple constraint of scope, time, and cost to include the "triple bottom line" of sustainability (Silvius & Schipper, 2014). Project managers are empowered to initiate sustainability conversations, champion integration, and measure relevant outcomes throughout the project lifecycle. For Technology Organizations, the framework underscores the need to create an enabling environment by formalizing governance, investing in competencies, and adapting sustainability practices to their specific context. As the matrix suggests, large enterprises should focus on robust governance and reporting systems, while startups can leverage agility to innovate on sustainable product features (Green Project Management, 2025; Jiménez et al., 2024). For Industry Standard and Policy Bodies like the PMI and IEEE, the findings call for an evolution of industry standards. These bodies should continue to integrate sustainability more deeply into core guides like the PMBOK®, while technical standards bodies can further mature metrics for measuring software's environmental and social impact, addressing the data consistency challenges identified in the literature (ISO/IEC, 2024; PMI, 2021).

VI. CONCLUSION

In response to the open-ended research question regarding how the Sustainable Development Goals can be systematically and effectively embedded into the software engineering project management lifecycle, this paper has argued and demonstrated that the answer lies in an integrative, multi-level approach. By constructing a new theoretical framework from the synthesis of Project Governance, Sustainable Software Engineering, and development methodologies, this study offers a clear and structured pathway for practitioners and scholars.

The core contribution of this paper is the Integrative Framework for SDG Embedding itself, a novel theoretical construct that is both comprehensive and actionable. Complemented by the Industry Applicability Matrix, the framework provides a robust tool for both academics and practitioners. For academics, it offers a new theoretical lens for analyzing sustainability in the technology sector. For practitioners, it serves as a practical guide for moving beyond aspirational goals to tangible project-level implementation.

This research is, by design, foundational and theoretical, and it lays the groundwork for numerous avenues of future empirical research. Future studies could conduct multi-case study validation to examine the framework's effectiveness in different organizational contexts. Another valuable direction would be to develop a quantitative model to measure the impact of the framework's adoption on both project performance and sustainability outcomes. Additionally, focused research could deepen the analysis of embedding specific, complex SDGs, such as SDG 5 (Gender Equality) or SDG 16 (Peace, Justice, and Strong Institutions), into software projects.

In conclusion, as software continues to shape our world, the responsibility to ensure it does so for the better has never been greater. The framework presented in this paper provides a crucial piece of the puzzle, offering the logic and structure needed to harness the power of software engineering in service of a more sustainable and equitable future for all (Sachs et al., 2022; United Nations, 2015/2025).

REFERENCES

1) Ahmad, M. O., & Al Baik, O. (2025). Fostering social sustainability in large scale Agile projects: Insights from Swedish software companies. Journal of Decision Systems, 34(1), Article 2464749. https://doi.org/10.1080/12460125.2025.2464749
2) Alamer, R., & Alharbi, O. (2025). Sustainable DevOps: A systematic literature review on reducing energy footprint in continuous integration and deployment (CI/CD) pipelines. International Journal of Computations, Information and Manufacturing, 10(1), 1–15. https://www.journals.gaftim.com/index.php/ijcim/article/view/565
3) Alharthi, S., Alshamsi, A., Alseiari, A., & Alwarafy, A. (2024). Auto scaling techniques in cloud computing: Issues and research directions. Sensors, 24(17), Article 5551. https://doi.org/10.3390/s24175551
4) Armenia, S., Dangelico, R. M., Nonino, F., & Pompei, A. (2019). Sustainable project management: A conceptualization-oriented review and a framework proposal for future studies. Sustainability, 11(9), Article 2664. https://doi.org/10.3390/su11092664
5) Bambazek, P., Groher, I., & Seyff, N. (2022, June). Sustainability in Agile software development: A survey study among practitioners. In 2022 International Conference on ICT for Sustainability (ICT4S) (pp. 13–23). IEEE. https://doi.org/10.1109/ICT4S55073.2022.00013
6) Calero, C., Moraga, M. Á., & García, F. (2022). Software, sustainability, and UN Sustainable Development Goals. IT Professional, 24(1), 41–48. https://doi.org/10.1109/MITP.2021.3117344
7) Calero, C., & Piattini, M. (2015). Introduction to green in software engineering. In C. Calero & M. Piattini (Eds.), Green in software engineering (pp. 3–27). Springer. https://doi.org/10.1007/978-3-319-08581-4_1
8) Calero, C., & Piattini, M. (2017). Puzzling out software sustainability. Sustainable Computing: Informatics and Systems, 16, 117–124. https://doi.org/10.1016/j.suscom.2017.10.011
9) Cao, Z., Zhou, X., Hu, H., Wang, Z., & Wen, Y. (2022). Toward a systematic survey for carbon neutral data centers. IEEE Communications Surveys & Tutorials, 24(2), 895–936. https://doi.org/10.1109/COMST.2022.3161275
10) Danushi, O., Forti, S., & Soldani, J. (2025). Carbon-efficient software design and development: A systematic literature review. ACM Computing Surveys, 57(10), Article 1. https://doi.org/10.1145/3728638
11) Duboc, L., Penzenstadler, B., Porras, J., Akinli Kocak, S., Betz, S., Chitchyan, R., … & Venters, C. C. (2020). Requirements engineering for sustainability: An awareness framework for designing software systems for a better tomorrow. Requirements Engineering, 25(4), 469–492. https://doi.org/10.1007/s00766-020-00336-y
12) Ericsson. (2025, June 10). ICT energy evolution: Telecom, data centers, and AI [White paper].

https://www.ericsson.com/en/reports-and-papers/white-papers/ict-energy-evolution-telecom-data-centers-and-ai

13) Fatima, I., & Lago, P. (2025, June). Ten years of Software Engineering in Society. In 2025 11th International Conference on ICT for Sustainability (ICT4S) (pp. 1–11). IEEE. https://doi.org/10.1109/ICT4S68164.2025.00010
14) Fesenko, T., Fesenko, G., Fesenko, H., Golovko, G., Bibik, N., & Bortnyk, A. (2024, May). Conceptualizing the ICT project management in the sustainability context. In 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST) (pp. 587–593). IEEE. https://doi.org/10.1109/SIST61555.2024.10629257
15) Feng, G., Hao, S., & Li, X. (2022). Project sustainability and public–private partnership: The role of government relation orientation and project governance. Sustainability, 14(8), Article 4724. https://doi.org/10.3390/su14084724
16) Freed, M., Bielinska, S., Buckley, C., Coptu, A., Yilmaz, M., Messnarz, R., & Clarke, P. M. (2023, August). An investigation of green software engineering. In European Conference on Software Process Improvement (pp. 124–137). Springer. https://doi.org/10.1007/978-3-031-42307-9_10
17) Freeman, R. E. (2010). Strategic management: A stakeholder approach. Cambridge University Press. https://doi.org/10.1017/CBO9781139192675
18) Funke, M., & Lago, P. (2026). Injecting sustainability in software architecture: A rapid review. In 10th IEEE/ACM International Workshop on Green and Sustainable Software (GREENS'26): GREENS @ ICSE 2026. IEEE/ACM.
19) Green Project Management. (2025). GPM P5 standard for sustainability in project management (Version 3.1). https://www.pmi.org/standards/gpm-p5-standard-for-sustainability-in-project-management-v3-1
20) Green Software Foundation. (2022). Sustainable systems: Architectural choices and sustainability. https://greensoftware.foundation/articles/sustainable-systems-architectural-choices-and-sustainability/
21) Green Software Foundation. (2024). Introduction to green software. Learn Green Software. https://learn.greensoftware.foundation/introduction/
22) Gurung, R., Khan, M. A., Siemon, D., & Porras, J. (2026). Toward a holistic conceptualization of green coding: A multi level perspective on sustainable software practices. In Proceedings of the 59th Hawaii International Conference on System Sciences (pp. 878–887). https://hdl.handle.net/10125/111499
23) Huq, A. T. U. (2025). Design and implementation of a framework for sustainability integration in Scrum Agile software development [Master's thesis, LUT University]. LUTPub. https://urn.fi/URN:NBN:fi-fe2025081181999
24) International Organization for Standardization (ISO). (2021). ISO 37000:2021 – Governance of organizations: Guidance. https://www.iso.org/standard/65036.html
25) International Organization for Standardization/International Electrotechnical Commission (ISO/IEC). (2024). ISO/IEC 21031:2024 – Information technology: Software carbon intensity (SCI). https://www.iso.org/standard/86612.html
26) International Telecommunication Union (ITU). (2024, March 20). Measuring the emissions & energy footprint of the ICT sector: Implications for climate action. https://www.itu.int/hub/publication/d-ind-clim-2023-01/
27) Jiménez, E., Moraga, M. Á., García, F., Calero, C., & García Mireles, G. A. (2024). Towards a software industry corporate social responsibility reference model for software sustainability. Sustainable Development, 32(4), 3471–3494. https://doi.org/10.1002/sd.2849
28) Kaur, J., Chana, I., & Bala, A. (2024). An autoscalable approach to optimize energy consumption using smart meters data in serverless computing. Science and Technology for Energy Transition, 79, Article 83. https://doi.org/10.2516/stet/2024078
29) Kern, E., Dick, M., Naumann, S., Guldner, A., & Johann, T. (2013). Green software and green software engineering: Definitions, measurements, and quality aspects. In ICT4S 2013: First International Conference on Information and Communication Technologies for Sustainability (pp. 87–91). ETH Zurich.
30) Khalifeh, A., Farrell, P., Alrousan, M., Alwardat, S., & Faisal, M. (2020). Incorporating sustainability into software projects: A conceptual framework. International Journal of Managing Projects in Business, 13(6), 1339–1361. https://doi.org/10.1108/IJMPB-12-2019-0289
31) König, C., Lang, D. J., & Schaefer, I. (2025). Sustainable software engineering: Concepts, challenges, and vision. ACM Transactions on Software Engineering and Methodology, 34(5), Article 1. https://doi.org/10.1145/3709352
32) Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L., & Ramasamy, R. K. (2023). Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. Sustainability, 15(2), Article 1121. https://doi.org/10.3390/su15021121
33) Mazak-Huemer, A., Vierhauser, M., & Groher, I. (2025). SustainScrum: Integrating sustainability assessment in a tailored Scrum process for computing quantitative sustainability indicators. Software and Systems Modeling, 24(2), 385–410.

https://doi.org/10.1007/s10270-025-01266-5

34) Moreira, A., Lago, P., Heldal, R., Betz, S., Brooks, I., Capilla, R., Coroamă, V. C., Duboc, L., Fernandes, J. P., Leifler, O., Nguyen, N.-T., Oyedeji, S., Penzenstadler, B., Peters, A.-K., Porras, J., & Venters, C. C. (2025). A roadmap for integrating sustainability into software engineering education. ACM Transactions on Software Engineering and Methodology, 34(5), Article 1. https://doi.org/10.1145/3709351

35) Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). The GREENSOFT model: A reference model for green and sustainable software and its engineering. Sustainable Computing: Informatics and Systems, 1(4), 294–304. https://doi.org/10.1016/j.suscom.2011.06.004

36) Nayak, K., Route, S., Sundararajan, M., & Ramesh, S. (2024). Sustainable continuous testing in DevOps pipeline. In 2024 1st International Conference on Sustainable Development in Software Engineering (pp. 1–6). IEEE. https://www.researchgate.net/publication/382512542

37) Pathania, P., Bamby, N., Mehra, R., Sikand, S., Sharma, V. S., Kaulgud, V., … & Burden, A. P. (2025, April). Calculating software's energy use and carbon emissions: A survey of the state of art, challenges, and the way ahead. In 2025 IEEE/ACM 9th International Workshop on Green and Sustainable Software (GREENS) (pp. 92–99). IEEE. https://doi.org/10.1109/GREENS66463.2025.00018

38) Piwowar Sulej, K., & Iqbal, Q. (2025). Sustainability and software development projects: Leadership, core self evaluation and empowerment as critical success factors. European Business Review, 37(2), 371–394. https://doi.org/10.1108/EBR-12-2023-0387

39) Project Management Institute (PMI). (2021). A guide to the project management body of knowledge (PMBOK® guide) (7th ed.). Project Management Institute.

40) Reis, T., Araújo, A., Gusmão, R., Farias, A., Silva, J., & Silva, A. (2025). Are we building sustainable software? Adoption, challenges, and early-stage strategies. In Proceedings of the 21st International Conference on Web Information Systems and Technologies (pp. 113–120). SciTePress. https://doi.org/10.5220/0013656100003985

41) Ribeiro, Q., Santos, A., Oliveira, K., Castro, J., & Lencastre, M. (2024, April). A view of the technical, individual, and social dimensions of sustainable software systems: A systematic literature review. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (pp. 1169–1177). ACM. https://doi.org/10.1145/3605098.3636031

42) Rinne, T. (2024). Adapting sustainable software development methods into agile processes [Master's thesis, University of Turku]. UTUPub. https://www.utupub.fi/handle/10024/179491

43) Sachs, J. D., Kroll, C., Lafortune, G., Fuller, G., & Woelm, F. (2022). The Sustainable Development Report 2022. Cambridge University Press. https://dashboards.sdgindex.org/reports/sustainable-development-report-2022

44) Sarkar, S., Naug, A., Luna, R., Guillen-Perez, A., Gundecha, V., Ghorbanpour, S., Mousavi, S., Markovikj, D., & Babu, A. R. (2024). Carbon footprint reduction for sustainable data centers in real-time. Proceedings of the AAAI Conference on Artificial Intelligence, 38(20), 22322–22330. https://doi.org/10.1609/aaai.v38i20.30238

45) Silvius, A. J., & Schipper, R. P. (2014). Sustainability in project management: A literature review and impact analysis. Social Business, 4(1), 63–96. https://doi.org/10.1362/204440814X13948909253866

46) Soares, I., Fernandes, G., & Santos, J. M. (2024). Sustainability in project management practices. Sustainability, 16(10), Article 4275. https://doi.org/10.3390/su16104275

47) Sokka, J. (2024). Managing software engineering with sustainability goals – A literature review [Master's thesis, University of Helsinki]. HELDA – Digital Repository of the University of Helsinki. http://hdl.handle.net/10138/576696

48) Too, E. G., & Weaver, P. (2014). The management of project management: A conceptual framework for project governance. International Journal of Project Management, 32(8), 1382–1394. https://doi.org/10.1016/j.ijproman.2013.07.006

49) United Nations. (2015/2025). The 17 Goals. Department of Economic and Social Affairs. https://sdgs.un.org/goals

50) World Wide Web Consortium (W3C). (2023, October 5). Web Content Accessibility Guidelines (WCAG) 2.2. https://www.w3.org/TR/WCAG22/

51) Zhang, L., Mohandes, S. R., Tong, J., Abadi, M., Banihashemi, S., & Deng, B. (2023). Sustainable project governance: Scientometric analysis and emerging trends. Sustainability, 15(3), Article 2441. https://doi.org/10.3390/su15032441

There is an Open Access article, distributed under the term of the Creative Commons Attribution – Non Commercial 4.0 International (CC BY-NC 4.0) (https://creativecommons.org/licenses/by-nc/4.0/), which permits remixing, adapting and building upon the work for non-commercial use, provided the original work is properly cited.