A Roadmap for Integrating Sustainability into Software Engineering Education

ANA MOREIRA, NOVA LINCS & NOVA University Lisbon, Portugal
PATRICIA LAGO, Vrije Universiteit Amsterdam, Netherlands
ROGARDT HELDAL, Western Norway University of Applied Sciences, Norway
STEFANIE BETZ, Furtwangen University, Germany
IAN BROOKS, University of the West of England Bristol, U.K.
RAFAEL CAPILLA, Rey Juan Carlos University, Spain
VLAD CONSTANTIN COROAMĂ, Roegen Centre for Sustainability, Switzerland
LETICIA DUBOC, La Salle, Universitat Ramon Llull, Spain
JOÃO PAULO FERNANDES, New York University Abu Dhabi, United Arab Emirates
OLA LEIFLER, Linköping University, Sweden
NGOC-THANH NGUYEN, Western Norway University of Applied Sciences, Norway
SHOLA OYEDEJI, Lappeenranta-Lahti University of Technology, Finland
BIRGIT PENZENSTADLER, Chalmers University of Technology, Sweden
ANNE-KATHRIN PETERS, KTH Royal Institute of Technology, Sweden
JARI PORRAS, Lappeenranta-Lahti University of Technology, Finland
COLIN C. VENTERS, European Organisation for Nuclear Research (CERN), Switzerland

The world faces escalating crises: record-breaking temperatures, widespread fires, severe flooding, increased oceanic microplastics, and unequal resource distribution. Academia introduces courses around sustainability to meet the new demand, but software engineering education lags behind. While software systems contribute to environmental issues through high energy consumption, they also hold the potential for solutions, such as more efficient and equitable resource management. Yet, sustainability remains a low priority for many businesses, including those in the digital sector. Business as usual is no longer viable. A transformational change in software engineering education is urgently needed. We must move beyond traditional curriculum models and fully integrate sustainability into every aspect of software development. By embedding sustainability as a core competency, we can equip future engineers not only to minimise harm but also to innovate solutions that drive positive, sustainable change. Only with such a shift can software engineering education meet the demands of a world in crisis and prepare students to lead the next generation of sustainable technology. This paper discusses a set of challenges and proposes a customisable education roadmap for integrating sustainability into the software engineering curricula. These challenges reflect our perspective on key considerations, stemming from regular, intensive discussions in regular workshops among the authors and the community, as well as our extensive research and teaching experience in the field.

CCS Concepts: • **Social and professional topics** → **Software engineering education**; **Sustainability**; Codes of ethics; • **Software and its engineering** → *Software creation and management*.

Manuscript submitted to ACM

Additional Key Words and Phrases: Software engineering, Sustainability, Computing, Education, Software sustainability, Sustainable software, Sustainable development goals, Software competencies, Sustainability skills.

ACM Reference Format:
Ana Moreira, Patricia Lago, Rogardt Heldal, Stefanie Betz, Ian Brooks, Rafael Capilla, Vlad Constantin Coroamă, Leticia Duboc, João Paulo Fernandes, Ola Leifler, Ngoc-Thanh Nguyen, Shola Oyedeji, Birgit Penzenstadler, Anne-Kathrin Peters, Jari Porras, and Colin C. Venters. 2024. A Roadmap for Integrating Sustainability into Software Engineering Education. In *Proceedings of Special Issue (TOSEM)*. ACM, New York, NY, USA, 27 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

IT systems and services form the backbone of modern society, permeating sectors like health and commerce to communication, education, energy, finance, and defence. While these complex, large-scale, software-intensive systems might be vital enablers for sustainable development, their development and utilisation also present significant sustainability challenges. The etymological definition of sustainability as the capacity to endure [8] highlights the broad and profound impact of IT on economic, social, and environmental dimensions—affecting issues like social equity, carbon emissions, and resource consumption [28]. Assessing this impact is particularly difficult [16]. Moreover, our dependence on computers to comprehend our increasingly complex world has grown. However, the use of large-scale data collection and advanced AI tech is generally not directed towards sustainability, as societal goals tend to be prominently directed towards economic short-term gains [28].

In response to these intertwined challenges, there is a growing call for “rethinking growth, rethinking efficiency, rethinking the state, rethinking the commons, and rethinking justice” [71], which in turn would call for “redirecting digitalisation toward stabilisation of human and planetary systems” [28] and move away from “narrow techno-economic mindsets and ideologies of control” [104]. This forward-looking paradigm must prioritise minimising environmental impact, championing social equity, and fostering economic resilience. It is, therefore, essential to equip professionals and students with the knowledge, skills, and tools needed to design, develop, and manage software and systems for such a paradigm.

Education is a key leverage point in enabling transition mindsets and providing opportunities to learn about alternative pathways forward [111]. As educators, we can play an important role in designing educational frameworks that instil a sustainability mindset among current and future IT professionals. Hence, the challenge is:

> *How can we infuse sustainability principles and cultivate essential skills and competencies in an already crowded computing curriculum to nurture the future generation of software engineering professionals with a sustainability mindset?*

In this context, it is essential to recognise the two distinct perspectives within the field: one that focuses on the sustainability of software itself—e.g., its technical durability—and another that explores how software solutions can address broader sustainability challenges, known as software engineering for sustainability. Both perspectives are crucial in shaping a sustainable digital future.

During the past four years, we focused on laying the foundation for integrating sustainability into computing education, while identifying current professionals’ training needs. This involved conducting two comprehensive studies: one examining how academia addresses sustainability in teaching [85], and the other engaging with industry to understand their stated needs, challenges, and practices when handling sustainability in their businesses [50]. Drawing from the findings of these two studies, the intensive discussions among the authors in regular workshops, input from the broader community [86], and our own research and teaching experience, we have identified a set of 14 key educational challenges. These challenges are classified into three categories: foundational (raise awareness, establish core concepts, integrate inter- and intra skills, and building the business case), practical competencies (ethical thinking, holistic view, technical sustainability, AI for sustainability, metrics and indicators, and greenwashing), and systemic (legal requirements and standards, advocacy and lobbying, engaging unresponsive universities, and facilitating industrial adoption). These challenges form the basis of a generic, yet flexible, framework that allows organisations to select and tailor their roadmap based on their sustainability maturity and investment plans when designing future educational or training programs for integrating sustainability. Such a roadmap emphasises broader societal challenges and prepares students to navigate complex, wicked problems by fostering personal development, self- and ethical awareness, and the capacity to respond to complex problems with care and empathy.

This paper is structured as follows: Section 2 begins with a discussion of sustainability in software engineering, followed by an overview of the two studies that constitute the groundwork for our roadmap, and finishes by positioning software engineering within the broader field of computing. Section 3 provides a thorough discussion of the set of challenges that should be considered in sustainability training and education of software engineers. Each of these challenges is discussed and mapped into the primary sustainability dimensions that it impacts. Section 4 presents a flexible and customisable framework with a classification of the identified challenges, allowing academic or industry organisations to tailor their approach based on their current sustainability maturity and the investment they are willing to make to further advance their sustainability efforts. Section 5 offers related work, and, finally, Section 6 concludes the paper and proposes incorporating our roadmap into educational frameworks for SE and computing to foster a sustainability mindset in future professionals.

2 BACKGROUND

This background section begins by discussing sustainability within the context of Software Engineering (SE), followed by a summary of two foundational studies for this paper: one focused on the sustainability competencies and needs of industry [50], and the other how sustainability has been integrated in education [85]. It concludes by situating software engineering within the broader field of computing.

2.1 Sustainability and Software Engineering

Sustainability has become a major concern in different fields, including software engineering. In this section, we provide an overview of the concept of sustainability in general, and in software engineering specifically. The Oxford English Dictionary defines sustainability as “the quality of being sustained”, where sustained can be defined as “capable of being endured” and “capable of being maintained” [77]. Endured is defined as “continuing to exist” and maintained as “being supported” [77]. This suggests that the concept of longevity as an expression of time and the ability to maintain are key factors at the heart of understanding sustainability. While the principles of sustainability have been known to numerous human cultures throughout history, its first scientific usage was most likely elucidated in the Carlowitz’s principles of sustainable forestry from 1713 [115]; “*do not use more resources than are available.*” However, the same system can have additional and competing functions such as filtering air and water, holding soil and preserving biodiversity, and recreation. As a result, people think of different systems and functions to be sustained including different time horizons. In contrast, the commonly used Brundtland [18] definition of sustainability encompasses two aspects: distributive intra-generational justice and inter-generational justice, where the latter highlights “the essential needs of the world’s poor, to which overriding priority must be given”, and the latter emphasises the preservation of the biosphere is a prerequisite. Tainter [105] argues that sustainability depends on understanding and controlling complexity including understanding: What should be sustained? For whom? For how long? At what cost?

Software sustainability is generally understood as "the capacity of a socio-technical system to endure" [8]. However, from a software engineering perspective, defining software sustainability in this way requires greater precision if we are to engineer software systems. It is worth noting that other engineering disciplines would refer to this as the quality of durability [13]. Similarly, a number of definitions align software sustainability to one or more software quality attributes that contribute to the sustainability of the software artefact including maintainability and extensibility [20, 44, 83, 100]. However, one of the principal challenges in defining sustainability in terms of software qualities is the need to demonstrate that the quality factors have been addressed in a quantifiable way. With the exception of Seacord [100], none of the proposed definitions suggested appropriate metrics that could be employed to measure the sustainability of a software system. Nevertheless, it is argued that maintainability and evolution of the software artefact are key enablers to achieving long-living software [32].

In addition, there are also different views and perspectives on software and sustainability, which are similar to those of the broader "IT and sustainability" field [61]: one looking at the sustainability of software itself, i.e. a technical notion of sustainable software, the other in the deployment of software solutions to address sustainability challenges, i.e. Software Engineering for Sustainability (SE4S) [113]. Acknowledging both views, the "Karlskrona Manifesto for Sustainability Design" extends the triple bottom line of sustainability (i.e., environmental, social, and economic) by including the technical (to account for the desired long-term use of software) and individual (addressing personal freedom, dignity, and fulfilment) [8]. While the individual dimension is not always represented, most literature in the field accounts for both the technical as well as the environmental, economic, and social dimensions [62]. As is the case in general with sustainability, the dimensions are not entirely independent and there are often trade-offs among them [7]. While current software engineering practice gives high value to the technical and economic dimensions, the social and environmental ones (and thus the crucial components of the sustainability concept as understood by the Brundtland commission) are often ignored [62]. However, as the demand for software systems grows, the technical, environmental, economic, and societal impacts are becoming important concerns to address [9, 45]. Sustainable Software Engineering (SSE) aims to minimise software systems negative impacts on all the dimensions of sustainability (technical, environmental, economic, societal and individual) while enhancing societal benefits [73]. This includes developing software to support larger societal systems that are energy-efficient, resilient and supporting long-term human and planetary flourishing. One existing approach to sustainability in software engineering is the design and development of long-term viable software systems [114]. The technical dimension of sustainability addresses SE concerns such as maintainability, evolution, scalability, security and interoperability. The environmental dimension of SSE focuses on the energy consumption of software systems and hardware. Efficient coding practices, good hardware utilisation, and optimised energy-efficient algorithms can minimise software energy use, resulting in lower emissions. On the economic dimension, SSE ensures the cost-effectiveness of developing and extending software systems with the supporting hardware life-cycle through the design and development of software systems that are modular and maintainable, reducing the need for frequent updates and replacements. Overtime, this reduces the development cost and electronic waste as older hardware systems can be used for extended periods before the need for replacements. The social dimension of SSE addresses how software systems affect relationships and interactions among groups of people and within communities, focusing on aspects like trust, equality, and sense of belonging. It also emphasises the importance of ensuring software systems comply with ethical standards to promote fairness and inclusivity. The individual dimension of SSE focuses on how software solutions affect users' and developers' well-being, as well as ensuring equal access to services. Key factors include supporting adaptation and offering personalisation options for end users.

Recent research identifies sustainability as a critical quality factor in software development processes [58, 59, 97, 113]. Integrating sustainability into software development lifecycles is crucial for dealing with the growing environmental and societal concerns caused by the rapid expansion of digital infrastructure.

2.2 An industry perspective on the needed competencies and skills

In previous work, we explored industry's views on the sustainability education and training of software engineers [50]. To do so, we conducted interviews and focus groups with IT and sustainability professionals from 28 organisations across various sectors and countries. We aimed to learn about (i) their interest in sustainability; (ii) their sustainability goals; (iii) the sustainability-related competencies and skills needed to achieve these goals.

Concerning their **overall interest in sustainability**, we asked companies about four perspectives: business, customers, shareholders, and stakeholders. From the business point of view, they highlighted the business opportunities and increased competitiveness brought by sustainability. They also felt they had to respond to customers' environmental concerns and, to a lesser extent, social concerns. Regarding shareholders, the economic benefits of the companies were the unsurprising main interest, but social concerns were also mentioned. Finally, employees' interests, the media and the regulatory framework were also mentioned as drivers to address sustainability within their businesses.

When discussing their **sustainability goals**, interviewees sought to make their processes and products more sustainable, recognising the need for a culture change in their employees and customers. Achieving such goals requires collaboration with partners and external entities, and the transformation of work processes, and technical tools. However, they noted that employees lacked an understanding of sustainability and still saw it as a trade-off against profit. Interviewees also complained about economic barriers and inadequate policies, when trying to reach their sustainability goals.

When it came to the **sustainability-related competencies and skills**, interviewed individuals acknowledged the importance of soft skills and technical competencies, but six companies faced difficulties in hiring the right talent with suitable sustainability-related skills. They wished their employees to have better communication skills and knowledge about sustainability metrics, so, they often resorted to training courses, sustainability consultants and collaborations with universities.

Finally, noting how interviewees referred to the different sustainability dimensions was also interesting. Most of them talked about environmental issues, such as carbon emissions and climate change. Those discussing the economic dimension often meant the company's ability to maintain its profits over time. The social dimension was mostly addressed through initiatives for employees' well-being, customer satisfaction, and societal welfare. Finally, for the technical aspects, some mentioned well-known software qualities, such as reusability, robustness, and security. Surprisingly, only a minority of companies mentioned the UN Sustainable Development Goals (SDGs).

From the insights we got from these interviews, we highlight that professionals need: (1) a comprehensive understanding of the multiple dimensions of sustainability and their alignment to business opportunities; (2) proficiency in soft and sustainability skills to promote awareness among employees, customers, and collaborators; and (3) tools and metrics to integrate sustainability into their businesses' products and processes. The Code book [49] of the interviews is shared as supplementary material to help readers better understand our insights.

2.3 An overview of academic endeavours

Based on a systematic study of the scientific literature on sustainability in computing education work we explored the current state of practice and which changes would be called for given literature on education for sustainable development and transformative education [85].

The literature review was based on an initial set of 473 papers from a database search covering the period of 2000–2022, reduced to 45 papers to specifically include papers addressing sustainability in computing education, and later extended through snowballing to 89 papers. This set was analysed with respect to the specific learning goals activities and research methods employed. The analysis made comparisons of learning goals and activities to what we found in the literature on learning for sustainable development, and compared research methods to methods established through ACM empirical standards.

The review concluded with recommendations and open questions around four themes: (1) the research on sustainability in computing education in light of the severity of our challenges and demands for unprecedented change, (2) connections to research on equality, justice, norms, values and power, (3) implications of current approaches to curriculum development and directions for future work, and (4) limitations of existing research and outlook.

Through these themes, the review assessed approaches to include sustainability in computing education and provided recommendations for change around several axes: to make more meaningful, substantial changes that can engage with the seriousness of the issues, to move away from a curricular alignment with existing business profiles and towards exploration of new skills and attitudes, to help students take action while being grounded in a mature relationship to difficult emotions, to connect explicitly with work on understanding power relations, equality and justice, to escape the "narrow techno-economic mindset" [104] represented by computational thinking, and to ensure that students can cope in a mature way with material that challenges them emotionally [87].

As the literature review on computing education was ambivalent about whether existing practices could offer concrete guidance in light of this, it ended with a series of open questions.

2.4 Software engineering in the context of Computing

This paper considers Software Engineering Education but much of the literature relates to Computing Education (as demonstrated in [85], see previous section) so we must consider which findings are relevant to SE. The distinction between SE, Computing, Computer Science and Information Technology remains contentious. SWEBoK states that SE builds on Computing [15]. We adopt the model of SE set out in the ACM Computing Curricula 2020: Paradigms for Global Computing Education which defines Computing as the over-arching discipline with SE, Computer Science (CS), Information Systems, Computer Engineering, Cyber-Security, and Data Science as subsets (see figure 2.2 in [36]). So recommendations for Computing related to competencies or skills needed for *"the development and use of rigorous methods for designing and constructing software artifacts that will reliably perform specified tasks"* [36, p.28] are bound to apply to SE. We note that the current ACM Curriculum Guidelines for SE 2014 [64] has only one mention of 'sustainability' whereas the guidelines for CS 2023 [56] has 62 mentions. At the very least an SE roadmap for sustainability education must identify which of the CS guidelines are relevant to SE. In comparison, an Industry-orientated skills framework Skills for the Information Age (SFIA) v8 [37] does include a Sustainability skill but surprisingly this is not listed as an SE competency. Looking to adjacent engineering disciplines, we also note that most other engineering disciplines, e.g. civil engineering, operate in a context where sustainability impact assessments are expected and sometimes legally required. Any SE roadmap should anticipate how sustainability impact assessment will increasingly be expected for SE products and hence influence may this SE roadmap.

3 CHALLENGES FOR A FUTURE EDUCATION PROGRAMME

A growing number of Higher Education Institutions (HEI) are recognizing the need to align their educational offerings with the pressing societal transformations being demanded. Despite some guidance for educators in computing (e.g., [12, 89, 117]), effectively integrating these frameworks into curricula remains a significant challenge [35]. The limitations of this integration process are well discussed in [85, Sec. 7.1].

Building on the results of our two studies [50, 85], the intensive discussions among the authors in regular workshops, the input from the broader community [86], and our significant research and teaching experience, we have identified a set of educational challenges that should be considered when designing a curriculum for future computing courses, including software engineering, that integrate sustainability. While this list is not exhaustive, it highlights key areas of knowledge we believe merit focus. Our proposal extends beyond the core sustainability knowledge outlined in [12, 40], which proposes a list of minimum competencies aimed at facilitating transformative systems changes to improve overall quality of life. It addresses a broader range of challenges, including inter- and intra-personal skills, technical sustainability, the business case for sustainability, assessing sustainability impacts and metrics, ethical considerations and values, legal requirements and standards, advocacy and lobbying efforts, the implications of Generative AI, green washing awareness and detection, and the responsibility of academic and industry organisations. The first ten challenges we identified stem from our two studies [50, 85]. Reflections on the implications of AI were suggested during the '2030 Software Engineering workshop [86], while the challenges regarding academic- and industry impact, and green washing are drawn from our experience in teaching and collaborating with industry.

In the following subsections, we describe each challenge, how it relates to our studies (where applicable), and the primary sustainability dimensions involved. We summarise the challenges and the related primary sustainability dimensions in Table 1 at the end of this section.

3.1 Raising awareness

Challenge. Raising awareness of sustainability remains a central challenge [50, 85]. Students and professionals are often unaware of how sustainability relates to software engineering.

Discussion. Sustainability is often not regarded at all, or seen either as a purely technical and economic issue (i.e., related to evolving systems), or exclusively as an environmental issue (i.e., addressing the energy footprint of systems) [60]. Software engineering as both a discipline and the systems it delivers, however, have a much larger impact on society and the planet at large. As such, it is crucial to connect sustainability goals with the software life cycle, from conception to deployment and evolution.

By redesigning software engineering education to teach such sustainability-aware competencies, we empower the students to embed sustainability in the organisations they will join after graduation. Similarly, we need to inform, or retrain, practitioners at all levels, about the potential impact of software engineering on sustainability, as well as how to recognise the relevance in their own business, and trace the (technical) software engineering actions to the (organisational) sustainability targets.

Map to sustainability dimensions. Software engineering has traditionally considered its impact on the technical and economic dimensions. Recently, both education and practice are investing in understanding and creating awareness about the related environmental impact also thanks to the climate targets enforced globally by most countries. However, as raising awareness is about "individuals becoming conscious" about the relevance of sustainability for SE, we consider the individual dimension as primary.

3.2 Establish harmonised and clear sustainability concepts

Challenge. There are too many and too vague definitions of sustainability and related concepts. This hinders effective action.

Discussion. Sustainability is often mistakenly perceived as a vague or abstract concept, with diametrically opposed meanings and interpretations, largely due to inadequate education and poorly defined contextual boundaries. This lack of clarity hinders effective integration of sustainability into various fields, including software engineering, and leads to misconceptions about its true value and impact. A sustainable future demands that communities and individuals are cognisant of sustainability concepts and principles and of the ramifications of human actions on the interconnected environmental, economic, and social dimensions of sustainability. Given the fundamental role of IT systems and services in modern society, sustainability must be considered an essential concern in the design for every IT product or service.

In our study, the conceptualisations of sustainability were divergent and incoherent, both among interviewees, but especially when contrasted with understandings of the drivers of our current crises and inability to implement changes through research and education [104]. The call for core sustainability concepts is in line with the recommendation to make meaningful changes calibrated against drivers of our predicaments, and with core topics related to equality, justice and power relations [85, Sec. 7.1.2]. However, the interviews presented a limited view of some of the symptoms of our interlinked crises, notably carbon emissions and climate change. For example, 50% of our surveyed organisations revealed a significant challenge in understanding sustainability concepts. While some companies view sustainability primarily through an environmental lens, very few acknowledge its social, economic, and technical aspects. Surprisingly, only four organisations mentioned the Sustainable Development Goals as pertinent to their operations.

Insufficient comprehension of sustainability's core concepts may result in various problems: perceiving it as an optional feature rather than a crucial aspect; struggling to collaborate with sustainability experts, who might not fully see the value of it; shying away from public debate on technology and sustainability, not advocating necessary policy changes; and seeing it as an auxiliary skill to IT rather than an integral component. Therefore, we ask *how we may best move conversations forward to ensure a broader and more meaningful framing of concepts needed.*

Education plays a pivotal role in addressing this challenge by providing clear terminology, scoping sustainability, dispelling prevalent misconceptions and myths, presenting current statistics, elucidating key concepts (e.g. dimensions of sustainability [84], sustainable development goals [1]), and introducing various models for contextualising sustainability (e.g. the doughnut model [92], the nine planetary boundaries [93], and orders of impact [52]).

To effectively integrate sustainability into software engineering, a standardized set of core concepts must be agreed upon, ensuring that software engineers have a solid foundation and understanding of sustainability's importance. This is not an easy task, in large part due the interdisciplinary nature of sustainability, but also due to the lack of consensus and even expertise in academia and industry, many of whose members still consider it as peripheral.

Finally, a roadmap for sustainability education in software engineering should identify which computing guidelines are relevant to developing sustainable software systems. This clarity will guide future curricula and ensure that sustainability is integrated as a fundamental, rather than a peripheral, aspect of the discipline.

---

[1] https://sdgs.un.org/goals

Map to sustainability dimensions. By establishing harmonised and clear concepts of sustainability we foresee that all the dimensions will gain more relevance. However, because the main challenges in education lie in changing social attitudes, individual mindsets, and technical frameworks, the social and individual dimensions are the most immediately relevant ones.

3.3 Integrating ethical thinking

Challenge. Critical ethical thinking and human values principles are often overlooked, yet they are essential due to the profound impact software systems have on modern society.

Discussion. Values are the core beliefs or principles that guide an individual's or society's decisions, shaping what they deem important or desirable [99]. Rooted in values, ethics are the rules or standards of conduct that govern how people should act in specific situations. Ultimately, values and ethics are fundamental concerns to making the world a fair and equitable place [101].

Over 57% of our interviewed organisations reveal that their customers and stakeholders want to protect the environment, and almost 30% are interested in focusing on sustainability due to moral concerns and social matters, resulting in the need for sustainability-value alignment of their business. Additionally, in our review of computing education literature, recommendations for incorporating values and ethics emphasise justice-centred computing and broader engagement with stakeholders who are marginalised or negatively impacted by IT systems. As values thinking requires students to reflect on their own motivations and actions, especially when engaging with stakeholders with different perspectives [116], it can lead to disruptive learning experiences. This occurs when students are encouraged to question the values underlying and critically examine the rationale behind IT project proposals in their capstone projects [31]. For these disruptions to be both productive and constructive, they must go beyond merely critiquing existing systems, instead offering alternative approaches to IT design.

Therefore, it is crucial for IT professionals to be guided by a clear code of ethics, such as the ACM code of ethics [1, 41, 42] to produce socially and environmentally responsible systems. Additionally, integrating values into the development process should be a key consideration to address broader societal and environmental concerns. While user-centered, user-experience, and value-sensitive design approaches address more than typical software qualities, they fall short of fully addressing core human values [3]. Also, work using human-values perspectives to identify gender bias in software [19] or incorporating ethical values into software design [2] show progress, but more systematic approaches are needed to fully integrate human values into software development.

In summary, a key challenge is how to integrate critical thinking into systems design courses in a way that fosters respectful, empathetic dialogue and broadens acceptance of the need to critically evaluate assumptions in IT design. At the same time, addressing how design decisions can minimize the potential misuse of software systems—such as in cyber warfare or the spread of misinformation through social media—requires a thoughtful combination of human values and ethics approaches within software engineering education. These interconnected issues present complex, open challenges for the field.

Map to sustainability dimensions. Human values and ethics are always linked with individuals, organisations and societies. Therefore, they map to the social and individual dimensions of sustainability.

3.4 Creating a holistic view

Challenge. A lack of a holistic view limits understanding of how sustainability dimensions interconnect and influence system design. Integrating systems thinking into the curriculum is essential to address this gap.

Discussion. Systems thinking is a term used to describe several knowledge traditions that seek to understand the underlying mechanisms of how systems work, including their dynamics and principles. These traditions—such as cybernetics, general systems theory, chaos theory, and system dynamics—invite thinking about social and ecological systems in terms of drivers and dynamics that are common to many systems, but need to be understood in context, especially when sustainability is involved [91]. Based on those traditions, systems thinking becomes a new lens, a new orienting principle for understanding the world and pathways of change in it [67]. We need systems thinking within software engineering education, as explained in Becker [6]. Hence, one challenge is how to teach this in a meaningful and effective way within software engineering. For example, we may teach systems thinking as a module but using examples from other than computing to explain systems thinking, hence there is always a slight disconnect. An open question here would be how to empower educators and professionals trained in computational thinking to shift perspectives towards holistic, inclusive and critical analyses of systems, and how to include self-awareness in relation to limitations of knowledge.

Systems thinking becomes a tool for critically reviewing the requirements of systems and finding better boundaries [85] when making judgements in reasoning about systems [30]. Also, it becomes a means to broaden the view of which types of competencies are valuable to orient education towards and to explicitly include the inner development of your capacity to be, care for, and act in, the complex adaptive systems of life.

Several of the interviewed organisations recognised the importance of systems thinking, emphasising the need for a holistic view of factors and interactions that could contribute to a better possible outcome [50]. This awareness presents an opportunity for educators, to integrate systems thinking into computing education. By adopting this approach, educators can lay the groundwork for fostering more just and sustainable futures in computing, moving away from current paradigms that may hinder progress in addressing societal challenges [6]. Therefore, students and professionals should be able to address more complex problems with larger positive impact on the various sustainability dimensions. In doing so, computing needs to be re-situated as a practice that seeks to redress problems in terms of those who are primarily affected negatively by contemporary exploitative and destructive practices [6].

Map to sustainability dimensions. Since systems thinking is about seeing and understanding the bigger picture and its complexity, it addresses all dimensions of sustainability. However, as we should primarily use systems thinking to reason about a system's design in a societal context, we consider both technical and social as primary dimensions.

3.5 Establish sustainability metrics and indicators

Challenge. Establishing metrics and indicators to grasp sustainability impacts remains a key challenge. These are essential to make improvements tangible and actionable.

Discussion. The role of metrics and other sustainability indicators are key to evaluating to what extent an organisation or a particular sustainability initiative achieve certain sustainability goals. Companies are adopting sustainable practices that have to be measured with adequate indicators aimed to prove they meet the relevant SDGs. While certain domains (e.g., energy or transportation) have well-established indicators aimed at evaluating the sustainability of a solution, others do not. Consequently, many organisations have difficulties assessing sustainability and they need to define appropriate indicators. Existing approaches to assess sustainability (e.g., [21, 31, 58, 59]) still need further development support to become widely used. In addition, public and private universities are being increasingly ranked according to sustainability indicators (e.g., Times Higher Education Impact Rankings), based on a set of predefined metrics [2].

Six of the interviewed organisations in this study [50] highlighted the lack of metrics to understand the direct impacts of the product/system adopting sustainable solutions, while another company observed there is a lack of awareness on whether the solutions adopted are sustainable enough, partly because of lack of metrics. These and other technical challenges aimed at calculating the carbon or energy footprint of sustainable solutions are one of the reasons to demand specific training on well-defined KPIs that can justify their achievement of a certain level of sustainability. Nevertheless, many companies today (e.g. Google, Microsoft, CGI) rely on public sustainability reports where they deliver a set of sustainability indicators achieved by their products. However, while many of these indicators concern energy savings (energy indicators being popular nowadays), it is unclear to what extent the metrics used by each company are standardised. Therefore, organisations providing sustainability indicators need to make public how these metrics and other sustainability indicators have been computed. How to collect the right data and collect the data sustainably remains a challenge that needs to be addressed. For example, while collecting marine data is expensive [74], many organisations have repeatedly collected the data already obtained by others due to the fragmented data sharing systems within the domain [66]. The environmental cost of operating computing platforms to handle the data is also an important metric to monitor [57]. In other situations the estimation of the metric is unclear or non-transparent. For instance, some airlines include in the customer e-ticket the number or CO2 tons consumed by a round-trip flight, but in many cases they don't mention who provides such estimations (i.e. lack of transparency). Also, they report the same number of CO2 tons equally for both flights, which is untrue if flights don't follow the same route.

Consequently, the use and adoption of existing sustainability metrics in various domains (e.g. climate change measures) and sustainability dimensions (e.g. technical and environmental) brings new opportunities for students to be taught on those metrics and rankings and these topics should be included in software engineering studies.

An open question remains whether a hands-on approach that provides concrete data and tracks progress is as crucial as fostering a broad understanding of the intricate and complex realities of sustainability. There is a clear need for lightweight tools that help assess sustainability impacts and measure the effects of changes within a software system. Such tools could also motivate technically inclined students to engage with sustainability, making them more receptive to the systems thinking paradigm by allowing them to quantify certain aspects of it.

A critical element of measuring the impact of sustainability is tool support. From a technical sustainability perspective, there are several commercial standalone and integrated tools available to aid in understanding the static and dynamic behaviour of software systems including (e.g., [51, 102, 106, 112]) that can be used to help raise the issue of sustainability-related requirements. While tools like this can be useful in helping CS students to identify sustainability concerns, light-weight tools are needed to help understand sustainability impacts and the effect of changes in a software system as the value of what these tools measure and how to understand what to change in a software system is often unclear.

Map to sustainability dimensions. While the envisioned measurements clarify in which way we are contributing to specific dimensions of sustainability (predominantly environmental, social, and individual), the definition and implementation of measurements and impacts is primarily a technical issue.

---

[2] https://www.timeshighereducation.com/impactrankings

3.6 Integrate SE competencies for sustainable software

Challenge. Ensuring technical sustainability (which includes stability, longevity, and maintainability, for example), of software systems remains an ongoing challenge. There is a pressing need to integrate technical sustainability into existing SE competencies, and where necessary, develop new skills that support the creation of truly sustainable systems.

Discussion. While there is no consensus within the field of software engineering as to how sustainability should be defined or understood, technical sustainability can generally be defined as the capacity of the software system to endure in changing environments [8, 95, 113]. Emerging views have evolved to argue that technically sustainable software is that which is "*explicitly designed for continuous maintainability and evolvability without incurring prohibitive technical debt and a negative impact on the dimensions of sustainability*" [114]. However, this raises the question regarding how we reason about software systems if the overall system relies on unsound material or other foundations for continued use. As such, should a software system still be explicitly designed for continued evolution or maintenance?

A study on industry needs for sustainability skills and competency in software engineering confirms that software development practitioners viewed sustainability in relation to quality attributes of IT products and services, such as reusability, robustness, security, etc. [50]. This aligns with previous studies investigating how software engineering professionals understand sustainability, highlighting traditional software quality concerns such as maintainability and extensibility [23, 44]. However, there are several fundamental software engineering skills and competencies that are missing from companies, including the application of software architecture in the design of sustainable software systems, and an understanding of and the application of software metrics to evaluate technical sustainability in a range of diverse application domains [50].

Software architectures play an important role and are fundamental to the development of technically sustainable, i.e., long-living, software systems, as they are the primary carrier of architecturally significant requirements (ASRs) and influence how developers can understand, analyse, test, and evolve a software system [65]. Software metrics provide a quantifiable measurement of software characteristics. While there is a large number of metrics to track software development to evaluate software quality and to certify software products, there is still a lack of information and understanding on how to choose the most suitable metrics in a particular context to evaluate technical sustainability [100]. While some organizations use metrics to measure their software systems, they often lack knowledge of how to integrate them automatically into the development pipeline for evaluating the overall quality of software systems for technical sustainability [50]. Designing effective software architectures and applying the right metrics can lead to the creation of more technically sustainable systems [114].

Despite the emergence of various software engineering methodologies over the last fifty years—such as waterfall, iterative waterfall, spiral, V-model, DevOps, and more—it is argued that the field has not yet fully mastered the creation of consistently successful software systems. Projects still often run over time and budget or fail due to their size and complexity [39, 54]. As a result, there remains a pressing need to align these methodologies with a core set of software engineering skills and competencies that can lead to quantifiable improvements in code and architecture design, software comprehension, and reuse, ultimately promoting standardisation and sustainable software systems. However, there is still a weak connection between existing metrics and their ability to quantify technical sustainability, and this gap must be addressed in SE courses to guide students in selecting the most suitable metrics to measure sustainability.

An open question, therefore, is how best to combine these competencies to promote the longevity of sustainable designs, while also fostering the ability to dismantle systems that no longer serve our long-term interests and sustainability goals.

Map to sustainability dimensions. The technical dimension is the primary focus as this challenge focus on qualities that IT systems must *satisfice*, along with the necessary metrics to evaluate both software quality and sustainability.

3.7 Integrating skills for inter- and intra-personal teamwork

Challenge. Incorporating inter- and intra-personal skills, central in sustainability, requires careful consideration as this is often seen as less relevant in software engineering education than hard (or technical) skills.

Discussion. Intra- and inter-personal skills also known as "soft skills" [79], is used to describe various generic skills or competencies that could be used in a variety of job contexts as opposed to hard skills that are linked with the technical expertise needed for the work. While the soft skills research initially focused on interpersonal skills, i.e., people and social skills, like communication and leadership, the recent research emphasises also the ability to manage oneself, i.e., intra-personal skills [69].

From our analysis of company needs, we identified several skills that either exist in or are missing from a company. We divided these skills into profession-specific hard skills, soft skills, and sustainability-specific skills. Categorising the findings of the mentioned on sustainability and soft skills as inter-personal and intra-personal skills reveals that companies reported to both have and to need more intra-personal skills. Personal qualities like common sense, systemic and critical reflection, and problem-solving skills, as well as values and ethics, confidence and ways of working, and stress management, are valued. As for the interpersonal skills, companies emphasized the importance of communication, collaboration and ability to work on multidisciplinary environment, for example.

In the literature review, the terms "soft-skills" and "inter-personal and intra-personal skills" are not particularly emphasised in the recommendations for changes in computing curricula. Yet, there has been criticisms of pushing human-centred concerns outside of the purview of technical education, introducing an artificial distinction that makes it harder to see that all technology has social implications and is embedded in a social fabric of values [10]. Having said that, the importance of these kind of skills is increasingly recognised in general education. Guides, such as the UNESCO key competences for sustainability [110] advocate and detail how several of these inter-personal and intra-personal skills can be applied to education at all ages. Other guides, such as the UK QAA/HEA Education for Sustainable Development [89], adapt these competences to higher education, breaking them down in specific skills, learning outcomes and implementations guidelines. The later includes promoting interdisciplinary learning, addressing economic, environmental, and social challenges; and including diverse voices in course content.

These are absolutely needed, but there are many challenges associated to them. To mention a few: Often universities lack funding or resources to support interdisciplinary and community-based learning; Communities and other social entities often have different goals and timelines than universities; Teachers' workload may be already too demanding to ask them to design and implement new skills into the curriculum; Teachers may feel that the time they have to cover their specific (i.e., technical) curricula is already too short; The types of skills necessary for wider stakeholder engagement may require methods of teaching and assessment that are currently some steps removed from the traditional types of teaching in computing. Training in empathetic communication, and having the ability to refrain from building destructive systems may challenge current notions of valid approaches to computing education.

These difficulties are not specific to software engineering courses and require a change in a university's mindset and structures. However, they may be especially challenging in computing courses, where the teachers themselves may lack the awareness, knowledge, tools, and collaborations to truly integrate sustainability into their courses. An open question is how to provide training for computing educators, students, and industry professionals alike, to make them comfortable to include notions of wider stakeholder dialogues, future thinking and empathetic listening to understand the social fabrics of which we are part.

Map to sustainability dimensions. Encouraging students to recognise the value of non-technical skills in their technical work requires a mindset change, both for students and educators. This process immediately impacts the individual and social dimensions of sustainability during the software engineering activities.

3.8 Building the business case for sustainability

Challenge. Sustainability is often deprioritised due to unclear economic benefits and cost concerns. SE education must empower students to articulate a compelling business case for sustainability, highlighting its long-term value through examples like risk mitigation and regulatory compliance.

Discussion. Sustainability and the SDG opens up the possibility of enormous new market for organisations [80]. In the mid-to-long term, companies can benefit from creating new technologies and systems to exploit these business opportunities. As one organisation in the interview study stated quite frankly "*Why we are so interested? [...] it's money.*" Nevertheless, it is not only about business opportunities, it is also about survival. Climate change, human capital, data security, ecological impact and human rights, among others, are considered to be financially material risks for companies worldwide [76]. As a result, the integration of material sustainability matters is now being considered imperative by investors. Furthermore, to direct funding to more sustainable businesses, sustainability reporting is also moving from voluntary to mandatory. In Europe, for example, the Corporate Sustainability Reporting Directive (CSRD) requires companies not only to report on their sustainability impacts, but also to provide information about their whole value chain, which creates a ripple effect, whether the business partners fall within the obligation of the CSRD or not [33].

Therefore, IT professionals need to understand better what drives the companies they work for, the business opportunities that sustainability offers, and the threats they faced by businesses and IT products causing harm to the environment and society. Understanding this might help them champion the idea of sustainability internally and justify it in terms of economic, environmental, and societal reasons.

Preparing IT professionals to be advocates for sustainability requires education. While some methods and tools exist to support building a business case (e.g. [63, 81] or [64, Sec. 4.7]), educating on creating the business case also imposes challenges. For example, knowledge about accounting for sustainability is still not widespread and SE educators will need to collaborate with other disciplines to integrate concepts of sustainability value. Another difficulty is the belief, also seen in the interviews with software practitioners, that building a business case is not the core of a Software Engineer's work [11]. Finally, yet another example is that sustainability is often seen as a trade-off. SE educators and researchers are still lacking a good understanding of how sustainability requirements are being (partially) eliminated during tradespace exploration and how to counteract this. Such exploration could be based on the work of [103].

Map to sustainability dimensions. Building a business case for sustainability primarily impacts the economic dimension by demonstrating the financial benefits and cost-effectiveness of sustainable practices.

3.9 Adopt evolving legal requirements and standards

Challenge. Unlike other fields where legislation and standards drive sustainability, the software and systems domain lacks such standards, and their evolution is slow and uneven globally.

Discussion. One of the pressures moving the industry towards improved sustainability is the need to comply with legal requirements and adopt standards. In many jurisdictions, sustainability-related laws and regulations are becoming stricter, and the penalties for non-compliance are increasing [90]. However, many of these regulations do not specifically address software, presenting a significant challenge in educating our students about compliance in the software they develop. Although laws such as the Corporate Sustainability Reporting Directive (CSRD) indirectly impact software, there are currently no established metrics specifically for software.

The software domain has few specific legal requirements but is affected by data protection legislation, Data Protection Impact Assessments, working time regulations for developers and modern slavery due diligence for outsourced development among others. SE professionals also need to be aware of how other indirectly affecting regulations may become system requirements and how they may impact business cases. A cautionary tale is the Software Engineer who was sentenced to prison for their part in the 'Dieselgate' scandal [78]. Legal requirements may, for example, include carbon taxes, customer data privacy, mandatory sustainability reporting (e.g., CSRD), Waste Electrical and Electronic Equipment (WEEE) disposal, and Environmental Impact Assessment for Data Centre construction proposals. The need to comply with these may have a direct impact on the requirements of a project. For example, Org. 14 reported that "*we've been working with our customers and see how EU regulations have evolved*".

Sustainability regulation is evolving at different pace and with varied focus in jurisdictions around the world. It is a caricature that such regulation is stronger in the European Union, e.g., General Data Protection Regulation but this is not uniformly the case. This adds complexity to SE projects with multinational implementation, complexity which should be recognised in SE education. In addition to legally mandated requirements, a wide range of standards where compliance is not legally required but may be expected by customers and other stakeholders. For example, an increasing number of large businesses are requiring their suppliers to provide carbon footprint data (Scope 3 reporting) [43] and many investors are expecting voluntary sustainability reporting complying with the Global Reporting Initiative or International Sustainability Standards Board [88].

Businesses can gain competitive edge in the marketplace by achieving certification such as ISO14001 for Environmental Management Systems [17]. Some businesses may choose to change their legal foundation to benefit a wider range of stakeholders than just shareholders, e.g. B Corps [5]. SE professionals need education on the systematic impacts of such legal and standards compliance and choices. This need is echoed in the ACM SE 2014 curriculum "Curriculum Guideline 15: Ethical, legal, and economic concerns and the notion of what it means to be a professional should be raised frequently." [64, p. 46].

The challenge is to anticipate the direction and extent of future law and regulation. The United Nations SDGs provide a globally adopted statement of ambition but do not have the status of international law. Nevertheless they can be expected to influence the direction of future national legislation. The increasing experience of the physical impacts of climate change may also be expected to drive an "Inevitable Policy Response" with potentially wide-ranging change to law [94]. There is a likely trend towards increased regulation of significant software systems as presaged by Data Protection Impact Assessment and AI Impact Assessment. A component of this challenge will be to work with researchers from other engineering and legal disciplines to map how Impact Assessment and Technology Assessment will affect SE and hence the education curricula [14].

Map to sustainability dimensions. The current impact focus of law and regulation is primarily on social and environmental dimensions. However, their adoption is primarily a social/societal challenge.

3.10 Changing cultures through advocacy and lobbying

Challenge. Engaging all stakeholders—organisations, societies, and governments—to prioritise sustainability in their decision-making processes can be achieved through responsible advocacy and lobbying efforts.

Discussion. Policy often lags behind social trends and is shaped by them. Over recent decades, movements like Corporate Social Responsibility, Fairtrade, Slow Food, the UN Global Compact, Natural Capitalism, and B-Corporations have driven progress toward sustainability. As argued in [85], part of the scientific community feels the need for stronger advocacy and lobbying on behalf of sustainability [38]. Educators can play a stronger role in promoting and shaping movements and, together with IT professional bodies (e.g. ACM, BCS), engage in policy formation. [4].

The value of engaging with policymakers was also recognised in our interview study; as Org. 9 remarked "*a lot of policymakers don't have a clue on digitalisation matters, and because of that, they don't know what they're doing while writing the law.*". Given the slow pace of regulation, it is crucial to act on sustainability without delay. Technology evolves rapidly, often outpacing the ability of IT professionals to anticipate its sustainability impacts [22]. Hence, universities should train future IT professionals to combine their technical and sustainability expertise to become effective advocates for sustainability. This entails incorporating tools for advocacy, media engagement, and lobbying into the curricula of technical universities.

Another key question is how to better align the efforts on advocacy and lobbying for sustainability between academia and business, reducing resistance and opposition from influential industry stakeholders? Implementing this alignment in curriculum design can foster greater openness to academia-industry cooperation among future professionals. Furthermore, how can activism can be directed towards promoting change within the business and education sectors themselves? It is essential to ensure that such activism is perceived as constructive and empathetic while being firmly rooted in a critical understanding of our current position and future needs.

These efforts come with challenges. A significant one is balancing the impartiality of researchers and educators with their involvement in advocacy and lobbying. While some sustainability topics are well-grounded in established science (e.g. Climate Science [53]), others are often shaped by more contestable normative values (e.g. acceptable levels of inequality in society). While we support researchers and educators taking informed, principled stances and being transparent about their values and positionality, we also recognize the challenges they may face in fostering open and constructive discussion that accommodate diverse perspectives.

Map to sustainability dimensions. The primary dimensions directly impacted are social and individual. Advocacy and lobbying foster societal shifts toward sustainability by influencing businesses, governments, and institutions to adopt more socially responsible practices, while also promoting personal responsibility for sustainability and driving behavioural change.

3.11 Reorienting AI to drive sustainability

Challenge. AI development can amplify societal and environmental challenges, from resource consumption to social inequalities. Reorienting AI for sustainability involves reshaping its objectives to address global issues, such as climate change, responsible resource management, and ethical decision-making.

Discussion. AI systems are developed to achieve specific, computationally manageable goals. They leverage vast computational power to recognise and replicate patterns from diverse datasets, including human-created content (code, text, audio, video) and scientific or socio-economic data. The aim is to accomplish well-defined tasks more efficiently than humans, such as generating code, texts, or videos. Optimising for such goals carries similar risks as noted with respect to how digitalisation in general can compound inequality, ecological overshoot and political dysfunction [28].

The advent of Generative Artificial Intelligence (GenAI) models, such as OpenAI's ChatGPT, GitHub's Copilot, or Google's LaMDA, has dramatically transformed many fields, including software engineering education [29]. Educators have expressed concerns that this technology could negatively affect students' acquired knowledge, particularly in areas like coding skills and assessment integrity [46].

Code generated by GenAI models can output vast amounts of code, but the results may be hard to assess, and frameworks for evaluating generated code continuously reveal new types of limitations [68], much as the inherent limitations of AI in autonomous vehicles reveal new types of challenges that have been outside of the scope of the design of vehicle control software [25]. If automation of coding practices persists, newly graduated software engineers must be able to validate whether systems are designed and implemented as intended, even as such alignment builds on notions of human intentionality that are theoretically difficult to justify [108] and the assumption that we can retain an ability to distinguish valuable code in the presence of AI technology has been empirically challenged in other contexts [24]. High-level concepts of software systems, such as requirements engineering, software architecture, and software testing [55] are argued to play major parts in the students' acquired knowledge. For GenAI models to be useful in software engineering practices, students would need to retain their capacity for discernment and understand the inner workings of machine learning models to effectively verify the outputs produced by those models [82]. Against the backdrop of what other fields have already experienced [24], this may prove to be a tough challenge.

AI development also intensifies many non-technical challenges. These include a possible lack of consideration for fundamental human values and the failure to conceptualise digital technologies within broader systemic frameworks, beyond mere computational terms. Moreover, the training and inference of GenAI and general AI models induces a substantial energy consumption in data centres worldwide. After a decade of fairly constant data centre energy consumption, in which efficiency gains could largely offset the growth in data centre service demand [70], with the advent of AI into the mainstream since about 2018, the global energy consumption of data centres has been abruptly rising – for example, by a factor of about 2.5 over 4 years for large global data centre operators [109]. Moreover, current GenAI models face several limitations, including issues related to data bias and a lack of explainability. These shortcomings raise important ethical concerns, such as liability, professional responsibility, or legal obligations of using GenAI models in producing software systems [55].

Map to sustainability dimensions. The technical dimension is fundamental, as we focusing on using AI responsibly to help addressing global issues and deliver sustainable benefits.

3.12 Activating academic organisations

Challenge. Academic organisations fall short in their work to shift educational efforts towards emancipatory education that can help students chart relevant pathways ahead given the scope and character of our challenges.

Discussion. The authors of this paper are dedicated to advancing the field of sustainability, recognising both its ethical imperative and its alignment with self-interest. Many of us are involved in teaching and research initiatives that emphasise sustainability, for example integrating compulsory modules and certified courses. However, what about those who have not yet invested in this area? How do we aim to engage them in emancipatory education for creating a more sustainable world?

Integrating sustainability into academic courses requires investment. The question arises: Who should bear this expense? Currently, the crisis is not within academia but in the world at large. So why should organisations change if they do not perceive an immediate crisis other than financial constraints and are constantly looking for more money?

There is extensive research on the challenges of organisational change [26, 98]. While it is easy to declare at a strategic level that sustainability is a priority for a university, without clear alignment of the incentives of organisations measurement, such statements can become meaningless. For true integration of sustainability, initiatives need both top-down and bottom-up approaches—neither can succeed on its own.

More research is needed to incorporate sustainability into our education system effectively. This will be a demanding task, as curricula are usually very packed. Therefore, setting some priorities will be necessary. One cannot simply add new material; some existing content will need to be removed as well. It is essential to acknowledge that very few individuals were initially employed in academia specifically for their expertise in sustainability. Thus, fostering a supportive environment for all faculty, regardless of their initial focus, is crucial for the collective advancement toward curricular with more focus on sustainability than today.

There is already research on incorporating sustainability into our education system, as demonstrated in our paper [85]. However, most existing studies fall into categories such as vision/proposals, experience reports, or proposals for ongoing initiatives. Among these, only 13 studies were case studies, and only a few followed rigorous research methodologies. For example, many papers lack sections addressing validity threats, undermining their trustworthiness. It is crucial to adopt more stringent methodological standards to enhance the impact and reliability of future research in this area. In particular, work is needed to provide concrete solutions to increase educators' knowledge and integrate sustainability into the curriculum without too high cost.

Map to sustainability dimensions. The social and individual dimensions are impacted, as addressing this challenge requires transforming educational structures, fostering collaboration, and reshaping mindsets.

3.13 Facilitating industrial adoption

Challenge: How to create a better impact in the industry and facilitate long-lasting changes. A long-term challenge is to foster meaningful change within companies and extend influence beyond isolated case studies and short-term action research interventions.

Discussion. When it comes to industrial organisations, partners often allow us to conduct interviews or observations, enabling us to propose potential solutions. However, it is significantly more challenging to get them to implement these solutions or participate in their evaluation [48]. Consequently, we often develop many solutions that we don't get to test in real-life settings, which is a significant issue, or they get implemented as one-off case studies done by researchers but the solutions don't not rolled over into daily processes because of the required resources for transfer. To effectively tackle sustainability, collaboration with industry and society at large is essential. Improved collaboration would enable us to teach our future students how to handle concrete solutions, ensuring that our academic efforts translate into real-world impact.

The paper we authored [50] represents a significant step towards identifying our future focus areas based on industry needs. Many of the challenges identified in this paper come from this study. Companies must take part in translating our findings into concrete teaching objectives, otherwise it is hard to know the final values of the needs. However, given that the industry is currently uncertain about the best approaches to sustainability, academia has a significant opportunity to influence and guide the industry. However, how to share these responsibilities is a challenge.

We must simultaneously address two key objectives: (1) supporting the industry in its present trajectory in its sustainability work and (2) envisioning the future in which we aim to achieve a more sustainable world, which goes beyond what the industry is willing to do today. As academics, we may need to assume greater responsibility in this dual role. While teaching concepts relevant and of interest to the industry is essential, it is equally important to be independent of its current demands. Balancing these considerations could ensure meaningful progress and lasting impact.

Map to sustainability dimensions. As this challenge revolves around collaboration between academy and industry to influence how sustainability can be taught, the social dimension is of primary importance. In addition, economic factors undoubtedly play a crucial role for industry partners, as increased commitment and investment is required from their side.

3.14 Identifying greenwashing

Challenge. How can greenwashing be identified to ensure that the term “sustainability" is not being misused. Learning how to identify greenwashing can help students who care about sustainability make informed career choices.

Discussion. The detection of greenwashing practices in the European Union is a fundamental aspect that requires attention, as EU [27] and national regulations demand organisations to incorporate certain green practices that, for instance, many companies report annually[3]. Overall, these claims about positive impact must be proven so that not only customers but all other stakeholders can assess whether (self-proclaimed) eco-friendly and green labels are supported by data or specific regulations [96].

Therefore, we need to train software engineering students on ways to detect greenwashing pertaining to the scope of software engineering (e.g., sustainability metrics in technical and environmental dimensions), while other disciplines will require different teaching materials to detect greenwashing.

Today, many companies tend to over-report sustainability achievements without further proof of truth. By leveraging advancements in artificial intelligence, we can efficiently and accurately analyse large volumes of text-based information, such as sustainability reports, enabling quicker insights and solutions [72]. On the other hand, it is essential to consider the significant energy consumption expected from data centers that support AI algorithms and other software. Hence, using the right metrics and tools to measure energy consumption in diverse software engineering areas like AI require teaching different sustainability metrics.

Map to sustainability dimensions. Economics plays an primary role, as some companies profit by misusing the term “sustainability."

4 EDUCATIONAL ROADMAPPING

An educational roadmap for integrating sustainability into SE should offer a structured yet flexible approach. It should outline foundational key topics and practical competencies required to achieve specific sustainability goals. The scale and ambition of these goals will differ based on the context of the academic or industry organisation, as some may already have established sustainability initiatives, while others may be in the early stages. As a result, our roadmap

---

[3] https://www.europarl.europa.eu/topics/en/article/20240111STO16722/stopping-greenwashing-how-the-eu-regulates-green-claims

Table 1. Overview of Challenges and related mapping on Sustainability Dimensions
(**Te**: Technical; **En**: Environmental; **Ec**: Economic; **So**: Social; **In**: Individual)

#	CHALLENGE	SHORT DESCRIPTION	PRIMARY DIMENSION				
			Te	En	Ec	So	In
1	Raising Awareness	The relevance of sustainability for SE is often not well understood.	☐	☐	☐	☐	■
2	Establish harmonised and clear sustainability concepts	We lack agreement and clarity on the concepts of sustainability.	☐	☐	☐	■	■
3	Integrating ethical thinking	Critical ethical thinking and human values principles are often overlooked.	☐	☐	☐	■	■
4	Creating a holistic view	Lack of holistic view hinders understanding of the interconnections among sustainability dimensions and system design.	■	☐	☐	■	☐
5	Establish sustainability metrics and indicators	Metrics and indicators are essential to make improvements tangible and actionable.	■	☐	☐	☐	☐
6	Integrate SE competencies for sustainable software	Technical sustainability (e.g., stability, longevity, maintainability) of software systems remains a challenge.	■	☐	☐	☐	☐
7	Integrate inter- and intra-personal skills	These are central in sustainability and require careful consideration as they are often seen as less relevant.	☐	☐	☐	■	■
8	Building the business case for sustainability	Sustainability is often deprioritised due to unclear economic benefits and cost. concerns.	☐	☐	■	☐	☐
9	Adopt evolving legal requirements and standards	Software and systems domain has very limited adoption of sustainability standards, and their evolution is globally slow and uneven.	☐	☐	☐	■	☐
10	Changing cultures through advocacy and lobbying	Become more effective in engaging all stakeholders to prioritise sustainability.	☐	☐	☐	■	■
11	Reorientating AI to drive sustainability	The good use of AI for addressing global issues.	■	☐	☐	☐	☐
12	Activating academic organisations	Push academic organisations to shift educational efforts towards emancipatory education.	☐	☐	☐	■	■
13	Facilitating industrial adoption	Foster meaningful, long-lasting change within companies.	☐	☐	■	■	☐
14	Identifying greenwashing	Techniques for detecting greenwashing practices to prevent the misuse of the term "sustainability".	☐	☐	■	☐	☐

serves as a customisable framework, enabling organisations to plan their investments and prioritise the challenges that align with their specific sustainability objectives.

For instance, similar to the ACM curricula, which provide knowledge units and key concepts for academic institutions to choose from, our proposed roadmap framework (Figure 1) offers a categorisation of the challenges presented in Section 3 organised into three columns: challenges that tackle *systemic* problems, i.e., general and encompassing entire organisations; challenges that address *foundational* problems, i.e., those that, when addressed, provide the prerequisites to learn SE skills and competencies; and, finally, challenges related to the practical *how-to competencies*, i.e., those focused on the application of specific skills in SE context. This flexible structure allows organisations—whether academic or industrial—to tailor their approach based on their current sustainability maturity, enabling them to select and focus on the relevant challenges that align with their existing sustainability efforts. By doing so, organisations can prioritise these challenges and create targeted educational roadmaps.

In Figure 1.(a), we imagine three focused examples:

Blue roadmap. In this example, an organisation would focus on ⑧ building the business case for ⑥ adopting AI techniques and learn the competencies ⑪ to balance the actual business benefits with the costs, e.g., needed to embed AI in the existing software systems; and ③ the possible ethical considerations related to, e.g., data collection and processing.

Yellow roadmap. In this example, the educational priority is to provide solid education for being able to detect ⑭ greenwashing, based on ② a clear foundational notion of sustainability and its relevance for software systems.

Red roadmap. To facilitate ⑬ industrial adoption, education could focus on teaching how to ⑫ measure the impact achieved with the efforts done, for example, to align software systems with the target sustainability goals. To facilitate the alignment, it is important to see the big picture, by ④ creating an holistic view.

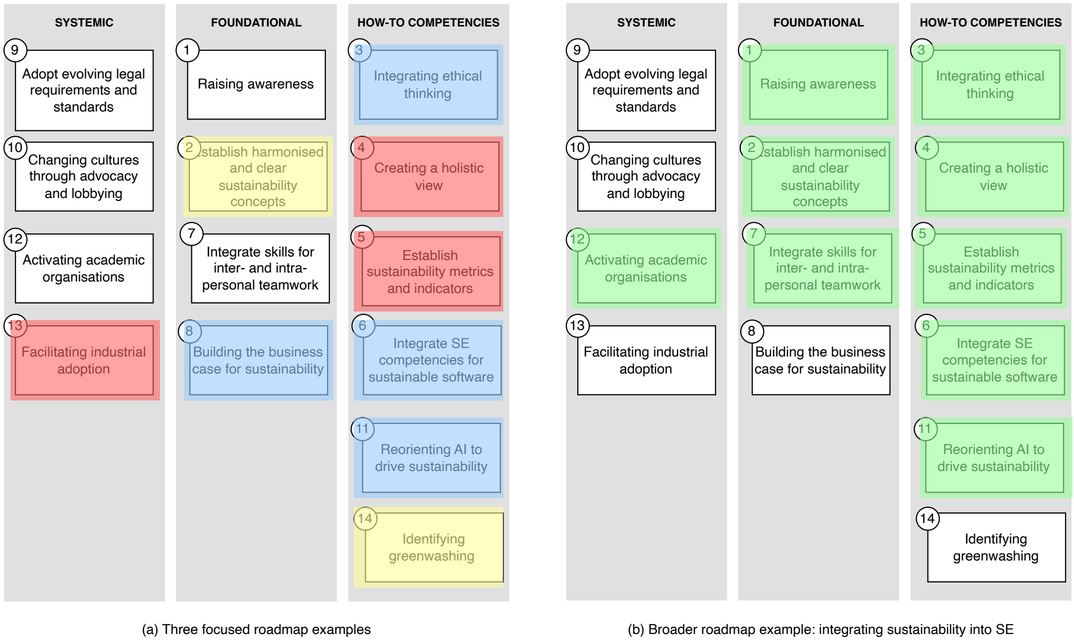
In addition, in Figure 1.(b) we provide a broader example:

Integrating sustainability into SE roadmap. In this example, ⑫ a university wants to start integrating sustainability into its SE education. As foundational skills and competencies, the prioritised challenges would include: ① modules meant to raise awareness in the students, ② provide them with a sound foundation of reference sustainability concepts, and ⑦ integrate, where applicable, teamwork skills with a special emphasis on their added value to master sustainability in the context at hand. As how-to competencies, we imagine that challenges ③, ④, ⑥, and ⑫ are all desirable in a modern SE programme. In addition, this example adopts challenge ⑪ as a running case, given the global role that AI plays in the digital transformation of modern society.

5 RELATED WORK

Haar [47] presents a sustainability roadmap as part of a comprehensive strategy, emphasising the need for clear targets, actionable steps, resource allocation, and a baseline for measuring Environmental, Social, and Governance (ESG) impacts alongside Sustainable Development Goals (SDGs). The roadmap stresses the importance of compliance with EU regulations, such as the Sustainable Finance Disclosure Regulation and the Corporate Sustainability Reporting Directive. It outlines essential steps, including value chain analysis, double materiality assessments, prioritisation of SDG/ESG criteria, target setting, and regular updates to the roadmap. Haar also addresses the challenge of greenwashing, where companies misrepresent their sustainability efforts to exaggerate their alignment with SDGs, potentially misleading customers and stakeholders.

In contrast to this business-oriented roadmap focused on SDGs, our customisable challenge-oriented roadmap identifies specific sustainability challenges categorised into systemic, foundational, and practical competencies that must be addressed to integrate sustainability into software engineering education. While there are overlapping themes—such as SDG selection, greenwashing, and the integration of sustainability into strategic business decisions—our roadmap uniquely targets the educational challenges associated with teaching sustainability within the context of software engineering.



(a) Three focused roadmap examples

(b) Broader roadmap example: integrating sustainability into SE

Fig. 1. Challenges and roadmap examples
(Numbers indicate the order in Table 1)

Evans [34] offers a comprehensive revision of existing research on competencies and pedagogies for sustainability education, identifying five key competencies: interpersonal communication, creative and strategic thinking, critical and normative analysis, transdisciplinary engagement, and systems thinking. The study recommends various pedagogical approaches, such as integrative, active, collaborative, experiential, research-based, and problem-based learning. The implementation of these findings in a Bachelor of Arts in Sustainability Studies program provides a practical roadmap for other educators. While this serves as a concrete example of integrating these competencies into an undergraduate degree, we propose a challenge-oriented, flexible and customizable roadmap for embedding sustainability principles within software engineering education.

Finally, it is worth acknowledging guidelines such as the OCDE PISA Global Competence framework [75] and the Education for Sustainable Development Guidance [89]. Both present detailed guidance on how to incorporate key sustainability competences into general education. They include several topics that we also highlighted, such as accommodating different perspectives and worldviews, taking actions, thinking systemically and critically, understanding norms and values. The former focus on school-level education, while the later targets university students. As expected, and different from ours, none of them reflects on the specificity of software engineering courses. They also focus more on the competences and learning outcomes that students should achieve, rather then the challenges that educators may have.

6 CONCLUSIONS

Based on previous work, this paper presents a set of challenges that should be addressed to effectively embed sustainability into software engineering education. We then organise these challenges into a customisable roadmap meant to inspire actionability and provide guidance for organisations to select the challenges they aim to prioritise. In particular, for academic organisations we envisage their use of our roadmap as a framework to incorporate sustainability-related learning objectives and courses, in general in computing education curricula, and in particular in software engineering education curricula.

The challenges and related discussions are grounded in (i) insights gained from interviews with IT professionals [49], (ii) a comparative analysis of these insights with recommendations from a recent literature review on sustainability in computing education [85], (iii) feedback received from the SE community [86], and (iv) our experience as researchers and educators dedicated to teaching sustainability.

With this roadmap, we aim to support a shift towards a stronger focus on sustainability in SE education. In addition, the challenges highlight areas where further action from both practice and research are needed. This is also an invitation to industry and society as a whole to actively participate in this important transformation.

As future work, we will instantiate our roadmap to a real case at a university or industry. We believe that our roadmap could and should be considered for inclusion in educational reference frameworks for SE, and computing, to result in a revised framework like MSIS' 2016 revision for Information Sciences [107]. By teaching the future generation of SE professionals to integrate sustainability in their skills and competencies, we can inject a brand new sustainability mindset in all sectors.

ACKNOWLEDGMENTS

The authors would like to thank all the interviewees who took part in the study, and also: NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP; SFI SmartOcean NFR Project 309612/F40, by KISS (16DHBKI061) with the financial support of the Federal Ministry of Education and Research, Germany; Catalan Government (Departament de Recerca i

Universitats) for the grant 2021 SGR 01396 given to the HER group; VU Digital Sustainability Center (DiSC); Chalmers Area of Advance ICT for Sustainability project 37460087.

REFERENCES

[1] 2018. ACM Code of Ethics and Professional Conduct. https://ethics.acm.org/. Accessed: 2024-09-19.

[2] Raizeh Alidoosti, Patricia Lago, Eltjo Poort, Maryam Razavian, and Antony Tang. 2022. Incorporating Ethical Values into Software Architecture Design Practices. In *19th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 124–127.

[3] Razieh Alidoosti, Patricia Lago, Maryam Razavian, and Anthony Tang. 2022. *Ethics in Software Engineering: A Systematic Literature Review*. Technical Report. Vrije Universiteit Amsterdam. http://tiny.cc/bf74vz

[4] Association for Computing Machinery. 2024. *ACM TechBriefs*. Technical Report. New York. https://www.acm.org/public-policy/techbriefs

[5] B Lab. 2018. About B Corps. https://bcorporation.net/about-b-corps

[6] Christoph Becker. 2023. *Insolvent: How to Reorient Computing for Just Sustainability*. MIT Press.

[7] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M Easterbrook, Birgit Penzenstadler, Norbet Seyff, and Colin C Venters. 2015. Requirements: The key to sustainability. *IEEE Software* 33, 1 (2015), 56–65.

[8] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C Venters. 2015. Sustainability design and software: The Karlskrona manifesto. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. IEEE, 467–476.

[9] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. 2015. Sustainability Design and Software: The Karlskrona Manifesto. In *Proceedings of the 37th International Conference on Software Engineering - Volume 2 (ICSE '15)*. IEEE Press, Piscataway, NJ, USA, 467–476. http://dl.acm.org/citation.cfm?id=2819009.2819082

[10] Catherine GP Berdanier. 2022. A hard stop to the term "soft skills". *Journal of Engineering Education* 111, 1 (2022), 14–18.

[11] Stefanie Betz, Dominic Lammert, and Jari Porras. 2022. Software Engineers in Transition: Self-Role Attribution and Awareness for Sustainability. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

[12] Guia Bianchi, Ulrike Pisiotis, and Marcelino Cabrera Giraldez. 2022. *GreenComp The European sustainability competence framework*. JRC Research Reports JRC128040. Joint Research Centre (Seville site). https://EconPapers.repec.org/RePEc:ipt:iptwpa:jrc128040

[13] Jan Bijen. 2003. *Durability of Engineering Structures Design, Repair and Maintenance*. Woodhead Publishing.

[14] Alan Bond and Jiří Dusík. 2020. Impact assessment for the twenty-first century - rising to the challenge. *Impact assessment and project appraisal* 38 (2020), 94–99. Issue 2. https://doi.org/10.1080/14615517.2019.1677083

[15] Pierre Bourque. 2020. The SWEBOK guide—More than 20 years down the road. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 1–2.

[16] Christina Bremer et al. 2023. *Assessing Energy and Climate Effects of Digitalization: Methodological Challenges and Key Recommendations*. Technical Report. nDEE Framing Paper Series. https://doi.org/10.2139/ssrn.4459526

[17] British Standards Institute. 2015. BS EN ISO 14001:2015: Environmental management systems. Requirements with guidance for use. , 48 pages. https://knowledge.bsigroup.com/products/environmental-management-systems-requirements-with-guidance-for-use?version=tracked

[18] Gro Harlem Brundtland. 1987. *Our Common Future: Report of the World Commission on Environment and Development*. Technical Report A/42/427. United Nations. https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf

[19] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.

[20] C. Calero, M. F. Bertoa, and M. Á. Moraga. 2013. A systematic literature review for software sustainability measures. In *2nd International Workshop on Green and Sustainable Software (GREENS)*. 46–53.

[21] GMV Chalmers. 2019. The SDG Impact Assessment Tool-a free online tool for self-assessments of impacts on Agenda 2030. *Policy* 1 (2019), 150–167.

[22] Kate Chatfield, Elisabetta Borsella, Elvio Mantovani, Andrea Porcari, and Bernd Carsten Stahl. 2017. An investigation into risk perception in the ICT industry as a core component of responsible research and innovation. *Sustainability (Basel, Switzerland)* 9, 8 (2017), 1424. https://doi.org/10.3390/su9081424

[23] Ruzanna Chitchyan, Christoph Becker, Stefanie Betz, Leticia Duboc, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. 2016. Sustainability Design in Requirements Engineering: State of Practice. In *Proceedings of the 38th International Conference on Software Engineering Companion* (Austin, Texas) *(ICSE '16)*. 533–542.

[24] Avishek Choudhury and Zaira Chaudhry. 2024. Large Language Models and User Trust: Consequence of Self-Referential Learning Loop and the Deskilling of Health Care Professionals. *J Med Internet Res* 26 (Apr 2024), e56764.

[25] Amit Chougule, Vinay Chamola, Aishwarya Sam, Fei Richard Yu, and Biplab Sikdar. 2024. A Comprehensive Review on Limitations of Autonomous Driving and Its Impact on Accidents and Collisions. *IEEE Open Journal of Vehicular Technology* 5 (2024), 142–161.

[26] Clayton M. Christensen. 2000. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Harvard Business Review Press, Boston, MA.

[27] European Commission. 2023. Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on substantiation and communication of explicit environmental claims (Green Claims Directive). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52023PC0166

[28] Felix Creutzig et al. 2022. Digitalization and the Anthropocene. *Annual review of environment and resources* 47 (2022), 479–509.

[29] Marian Daun and Jennifer Brings. 2023. How ChatGPT will change software engineering education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 110–116.

[30] Leticia Duboc, Curtis McCord, Christoph Becker, and Syed Ishtiaque Ahmed. 2020. Critical Requirements Engineering in Practice. *IEEE Software* (February 2020).

[31] Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff, and Colin C Venters. 2020. Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow. *Requirements Engineering* 25, 4 (2020), 469–492.

[32] Zoya Durdik, Benjamin Klatt, Heiko Koziolek, Klaus Krogmann, Johannes Stammel, and Roland Weiss. 2012. Sustainability guidelines for long-living software systems. In *28th IEEE International Conference on Software Maintenance (ICSM)*. 517–526.

[33] European Union. 2024. Corporate Sustainability Reporting Directive (CSRD). Official Journal of the European Union.

[34] Tina Lynn Evans. 2019. Competencies and pedagogies for sustainability education: A roadmap for sustainability studies program development in colleges and universities. *Sustainability* 11, 19 (2019), 5526.

[35] Evelien S Fiselier and James WS Longhurst. 2018. A critical evaluation of the representation of the QAA and HEA guidance on ESD in public web environments of UK higher education institutions. *Implementing sustainability in the curriculum of universities: Approaches, methods and projects* (2018), 223–246.

[36] CC2020 Task Force. 2020. *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA.

[37] SFIA Foundation. 2021. Skills Framework for the Information Age v8: The Framework Reference. , 268 pages. https://sfia-online.org/en/sfia-8/documentation

[38] Charlie J. Gardner, Aaron Thierry, William Rowlandson, and Julia K. Steinberger. 2021. From Publications to Public Actions: The Role of Universities in Facilitating Academic Advocacy and Activism in the Climate and Ecological Emergency. *Frontiers in Sustainability* 2 (2021).

[39] Mahbouba Gharbi, Arne Koschel, and Andreas Rausch. 2019. *Software Architecture Fundamentals: A Study Guide for the Certified Professional for Software Architecture®–Foundation Level–iSAQB compliant*. dpunkt. verlag.

[40] Harold Glasser and Jamie Hirsh. 2016. Toward the development of robust learning for sustainability core competencies. *Sustainability: The Journal of Record* 9, 3 (2016), 121–134.

[41] DW Gotterbarn, Bo Brinkman, Catherine Flick, Michael S Kirkpatrick, Keith Miller, Kate Vazansky, and Marty J Wolf. 2018. ACM code of ethics and professional conduct. (2018).

[42] Don Gotterbarn, Amy Bruckman, Catherine Flick, Keith Miller, and Marty J Wolf. 2017. ACM code of ethics: a guide for positive action. *Commun. ACM* 61, 1 (2017), 121–128.

[43] Greenhouse Gas Protocol. 2015. Corporate Standard. http://www.ghgprotocol.org/corporate-standard

[44] Iris Groher and Rainer Weinreich. 2017. An Interview Study on Sustainability Concerns in Software Development Projects. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 350–358. https://doi.org/10.1109/SEAA.2017.70

[45] Jennifer Gross. 2024. Designing for Sustainability - The Effect of Software Sustainability on Quality-in-Use. In *Quality of Information and Communications Technology*. Springer, 427–434. https://doi-org.ezproxy.cc.lut.fi/10.1007/978-3-031-70245-7_31

[46] You Jeen Ha, Sue Hendrickson, Adam Nagy, Elisabeth Sylvan, and Tom Zick. 2023. The Impacts of Generative AI on Teaching and Learning. https://cyber.harvard.edu/story/2023-06/impacts-generative-ai-teaching-learning Accessed: 2024-09-18.

[47] Gitte Haar. 2024. Sustainability Roadmap and SDGs. In *Rethink Economics and Business Models for Sustainability: Sustainable Leadership based on the Nordic Model*. Springer, 163–172.

[48] Rogardt Heldal. 2024. Is generalisation hindering the adoption of your findings?. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM/IEEE.

[49] Rogardt Heldal, Ngoc-Thanh Nguyen, Ana Moreira, Patricia Lago, Leticia Duboc, Stefanie Betz, Vlad C Coroamă, Birgit Penzenstadler, Jari Porras, Rafael Capilla, et al. 2024. Codebook of paper "Sustainability Competencies and Skills in Software Engineering: An Industry Perspective". https://doi.org/10.5281/zenodo.13826933

[50] Rogardt Heldal, Ngoc-Thanh Nguyen, Ana Moreira, Patricia Lago, Leticia Duboc, Stefanie Betz, Vlad C Coroamă, Birgit Penzenstadler, Jari Porras, Rafael Capilla, et al. 2024. Sustainability competencies and skills in software engineering: An industry perspective. *Journal of Systems and Software* 211 (2024), 111978. https://doi.org/10.1016/j.jss.2024.111978

[51] hello2morrow. [n. d.]. Sonargraph. https://www.hello2morrow.com/products/sonargraph Accessed on: Sept., 30, 2024.

[52] Lorenz M Hilty and Bernard Aebischer. 2015. ICT for sustainability: An emerging research field. In *ICT innovations for Sustainability*. Springer, 3–36.

[53] Intergovernmental Panel on Climate Change. 2022. Sixth Assessment Report. https://www.ipcc.ch/assessment-report/ar6/

[54] Ivar Jacobson. 2018. 50 years of SE, so now what? 40th International Conference on Software Engineering (ICSE).

[55] Vassilka D Kirova, Cyril S Ku, Joseph R Laracy, and Thomas J Marlowe. 2024. Software engineering education must adapt and evolve for an llm environment. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1.* 666–672.

[56] Amruth N Kumar, Rajendra K Raj, Sherif G Aly, Monica D Anderson, Brett A Becker, Richard L Blumenthal, Eric Eaton, Susan L Epstein, Michael Goldweber, Pankaj Jalote, et al. 2024. Computer Science Curricula 2023.

[57] Saurabh Kumar and Rajkumar Buyya. 2012. Green cloud computing and environmental sustainability. *Harnessing green IT: principles and practices* (2012), 315–339.

[58] P Lago. 2019. Architecture Design Decision Maps for Software Sustainability. In *41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE/ACM, 61–64.

[59] Patricia Lago and Nelly Condori-Fernandez. 2022. The Sustainability Assessment Framework (SAF) Toolkit: Instruments to help Sustainability-driven Software Architecture Design Decision Making. https://github.com/S2-group/SAF-Toolkit

[60] Patricia Lago, Danny Greefhorst, and Eoin Woods. 2022. Architecting for Sustainability. In *International Conference on Informatics for Environmental Protection: Environmental Information and Communication Technologies (EnviroInfo)*, Wohlgemuth V., Naumann S., Arndt H.-K., Behrens G., and Hob M. (Eds.). Gesellschaft fur Informatik (GI), 199 – 209.

[61] Patricia Lago, Rick Kazman, Niklaus Meyer, Maurizio Morisio, Hausi A Müller, and Frances Paulisch. 2013. Exploring initial challenges for green software engineering: summary of the first GREENS workshop, at ICSE 2012. *ACM SIGSOFT Software Engineering Notes* 38, 1 (2013), 31–33.

[62] Patricia Lago, Sedef Akinli Koçak, Ivica Crnkovic, and Birgit Penzenstadler. 2015. Framing sustainability as a property of software quality. *Commun. ACM* 58, 10 (2015), 70–78.

[63] Dominic Lammert. 2024. Bridging academic software sustainability design with corporate business planning. (2024).

[64] R LeBlanc, A Sobel, JL Diaz-Herrera, and TB Hilburn. 2014. Software Engineering 2014-Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering a Volume of the Computing Curricula Series. *Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery* (2014).

[65] C. Lilienthal. 2019. *Sustainable Software Architecture: Analyze and Reduce Technical Debt.* Rocky Nook.

[66] Keila Lima et al. 2022. Marine data sharing: Challenges, technology drivers and quality attributes. In *International Conference on Product-Focused Software Process Improvement*. Springer, 124–140.

[67] Björn-Ola Linnér and Victoria Wibeck. 2021. Drivers of sustainability transformations: leverage points, contexts and conjunctures. *Sustainability Science* 16, 3 (2021), 889–900.

[68] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 21558–21572. https://proceedings.neurips.cc/paper_files/paper/2023/file/43e9d647ccd3e4b7b5baab53f0368686-Paper-Conference.pdf

[69] Sara Isabel Marin-Zapata, Juan Pablo Román-Calderón, Cristina Robledo-Ardila, and Maria Alejandra Jaramillo-Serna. 2022. Soft skills, do we know what we are talking about? *Review of managerial science* 16, 4 (2022), 969–1000.

[70] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. 2020. Recalibrating global data center energy-use estimates. *Science* 367, 6481 (2020), 984–986. https://doi.org/10.1126/science.aba3758

[71] Timon McPhearson et al. 2021. Radical changes are needed for transformations to a good Anthropocene. *Npj urban sustainability* 1, 1 (2021), 5.

[72] Wayne Moodaley and Arnesh Telukdarie. 2023. Greenwashing, Sustainability Reporting, and Artificial Intelligence: A Systematic Literature Review. *Sustainability* 15, 2 (2023). https://doi.org/10.3390/su15021481

[73] Stefan Naumann, Eva Kern, Markus Dick, and Timo Johann. 2015. Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects. In *ICT Innovations for Sustainability. Advances in Intelligent Systems and Computing*, Vol. 310. Springer, 427–434. https://doi-org.ezproxy.cc.lut.fi/10.1007/978-3-319-09228-7_11

[74] Ngoc-Thanh Nguyen et al. 2023. Engineering Challenges of Stationary Wireless Smart Ocean Observation Systems. *IEEE Internet of Things Journal* 10, 16 (2023), 14712–14724.

[75] OECD. 2018. Preparing Our Youth for an Inclusive and Sustainable World: The OECD PISA Global Competence Framework. https://www.oecd.org/content/dam/oecd/en/topics/policy-sub-issues/global-competence/Handbook-PISA-2018-Global-Competence.pdf Accessed: 2024-09-30.

[76] OECD. 2024. *Global Corporate Sustainability Report 2024.* Report. OECD Publishing, Paris. https://doi.org/10.1787/8416b635-en

[77] OED. 2023. *Oxford English Dictionary.* Oxford University Press.

[78] Office of Public Affairs. 2017. Volkswagen Engineer Sentenced for His Role in Conspiracy to Cheat U.S. Emissions Tests. https://www.justice.gov/opa/pr/volkswagen-engineer-sentenced-his-role-conspiracy-cheat-us-emissions-tests

[79] Joseph A Olmstead et al. 1974. Research on Utilization of Assessment Results and Methods. Final Technical Report. (1974).

[80] Jeremy Oppenheim, Olivia Boyd, Gina Campbell, Nishita Dewan, Alex Evans, Melinda George, Saira George, Gail Klintworth, Alessandra Kortenhorst, Dinah McLeod, Janez Potocnik, Corinne Sawers, and Aniket Shah. 2017. Better business, better world. *Business & Sustainable Development Commission* (2017).

[81] Shola Oyedeji, Ahmed Seffah, and Birgit Penzenstadler. 2017. Sustainability quantification in requirements informing design. *6th Int. Work. Requir. Eng. Sustain. Syst* 1 (2017).

[82] Francesc Pedro, Miguel Subosa, Axel Rivas, and Paula Valverde. 2019. Artificial intelligence in education: Challenges and opportunities for sustainable development. (2019).

[83] B. Penzenstadler. 2013. Towards a definition of sustainability in and for software engineering. In *28th Annual ACM Symposium on Applied Computing (SAC)*. 1183–1185.

[84] Birgit Penzenstadler. 2014. Infusing Green: Requirements Engineering for Green In and Through Software Systems.. In *RE4SuSy@ RE*. 44–53.

[85] Anne-Kathrin Peters et al. 2024. Sustainability in computing education: A systematic literature review. *ACM Transactions on Computing Education* 24, 1 (2024), 1–53.

[86] Mauro Pezzè, Matteo Ciniselli, Luca Di Grazia, o, and Ketai Qui. 2024. The Trailer of the ACM 2030 Roadmap for Software Engineering. *Software engineering notes* (2024), 10.

[87] Panu Pihkala. 2022. The Process of Eco-Anxiety and Ecological Grief: A Narrative Review and a New Proposal. *Sustainability* 14, 24 (2022).

[88] Simone Pizzi, Salvatore Principale, and Elbano de Nuccio. 2023. Material sustainability information and reporting standards. Exploring the differences between GRI and SASB. *Meditari accountancy research* 31, 6 (2023), 1654–1674. https://doi.org/10.1108/MEDAR-11-2021-1486

[89] Quality Assurance Agency for Higher Education and Advance HE. 2021. *Education for Sustainable Development Guidance*. Technical Report. QAA and Advance HE, Gloucester. 1–51 pages. https://membershipresources.qaa.ac.uk/s/article/Education-for-Sustainable-Development-Guidance

[90] Lavanya Rajamani and Jacqueline Peel. 2021. Reflections on a decade of change in international environmental law. *Cambridge International Law Journal* 10, 1 (2021), 6–31. https://doi.org/10.4337/cilj.2021.01.01

[91] Magnus Ramage and Karen Shipp. 2020. *Systems Thinkers* (second edition ed.). Springer.

[92] Kate Raworth. 2017. *Doughnut economics: seven ways to think like a 21st century economist*. Chelsea Green Publishing, White River Junction, Vermont.

[93] Katherine Richardson, Will Steffen, Wolfgang Lucht, Jørgen Bendtsen, Sarah E Cornell, Jonathan F Donges, Markus Drüke, Ingo Fetzer, Govindasamy Bala, Werner von Bloh, Georg Feulner, Stephanie Fiedler, Dieter Gerten, Tom Gleeson, Matthias Hofmann, Willem Huiskamp, Matti Kummu, Chinchu Mohan, David Nogués-Bravo, Stefan Petri, Miina Porkka, Stefan Rahmstorf, Sibyll Schaphoff, Kirsten Thonicke, Arne Tobian, Vili Virkki, Lan Wang-Erlandsson, Lisa Weber, and Johan Rockström. 2023. Earth beyond six of nine planetary boundaries. *Science Advances* 9, 37 (sep 2023), eadh2458. https://doi.org/10.1126/sciadv.adh2458

[94] Nicole Rogers and Michelle Maloney. 2022. The Anthropocene judgments project: A thought experiment in futureproofing the common law. *Alternative law journal* 47, 3 (2022), 173–178.

[95] Maria Salama and Rami Bahsoon. 2016. A Taxonomy for Architectural Stability. In *31st Annual ACM Symposium on Applied Computing (SAC)*. ACM, New York, NY, USA, 1354–1357.

[96] Célia Santos, Arnaldo Coelho, and Alzira Marques. 2024. A systematic literature review on greenwashing and its relationship to stakeholders: state of art and future research agenda. *Management Review Quarterly* 74, 3 (2024), 1397–1421. https://doi.org/10.1007/s11301-023-00337-5

[97] Kamal Uddin Sarker, Aziz Bin Deraman, Raza Hasan, and Ali Abbas. 2020. SQ-Framework for Improving Sustainability and Quality into Software Product and Process. *International Journal of Advanced Computer Science and Applications* 11 (2020).

[98] Edgar H Schein. 2010. *Organizational culture and leadership*. Vol. 2. John Wiley & Sons.

[99] Shalom H Schwartz. 2007. Basic human values: Theory, measurement, and applications. *Revue française de sociologie* 47, 4 (2007), 929.

[100] R.C. Seacord, J. Elm, W. Goethert, G.A. Lewis, D. Plakosh, J. Robert, L. Wrage, and M. Lindvall. 2003. Measuring software sustainability. In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.* 450–459. https://doi.org/10.1109/ICSM.2003.1235455

[101] Richard Shearman. 1990. The meaning and ethics of sustainability. *Environmental management* 14 (1990), 1–8.

[102] Sonar. [n. d.]. SonarCloud. https://www.sonarsource.com/products/sonarcloud Accessed on: Sept., 30, 2024.

[103] Eric Spero, Christina L Bloebaum, Brian J German, Art Pyster, and Adam M Ross. 2014. A research agenda for tradespace exploration and analysis of engineered resilient systems. *Procedia Computer Science* 28 (2014), 763–772.

[104] Isak Stoddard, Kevin Anderson, Stuart Capstick, Wim Carton, Joanna Depledge, Keri Facer, Clair Gough, Frederic Hache, Claire Hoolohan, Martin Hultman, Niclas Hällström, Sivan Kartha, Sonja Klinsky, Magdalena Kuchler, Eva Lövbrand, Naghmeh Nasiritousi, Peter Newell, Glen P. Peters, Youba Sokona, Andy Stirling, Matthew Stilwell, Clive L. Spash, and Mariama Williams. 2021. Three Decades of Climate Mitigation: Why Haven't We Bent the Global Emissions Curve? *Annual Review of Environment and Resources* 46, 1 (2021), 653–689.

[105] Joseph A Tainter. 2006. Social complexity and sustainability. *ecological complexity* 3, 2 (2006), 91–103.

[106] Teamscale. [n. d.]. Teamscale. https://teamscale.com/ Accessed on: Sept., 30, 2024.

[107] Heikki Topi, Helena Karsten, Sue A Brown, João Alvaro Carvalho, Brian Donnellan, Jun Shen, Bernard C Y Tan, and Mark F Thouin. 2017. MSIS 2016 Global Competency Model for Graduate Degree Programs in Information Systems. *Communications of the Association for Information Systems* 40, 1 (2017), 18. https://aisel.aisnet.org/cais/vol40/iss1/18/

[108] Alexey Turchin. 2019. AI alignment problem:"human values" don't actually exist. https://philpapers.org/rec/TURAAP.

[109] UNCTAD. 2024. *Digital Economy Report 2024 – Shaping an environmentally sustainable and inclusive digital future*. Technical Report. UN trade & development. https://unctad.org/publication/digital-economy-report-2024

[110] UNESCO. 2017. *Education for Sustainable Development Goals: Learning Objectives*. UNESCO Publishing, Paris. https://unesdoc.unesco.org/ark:/48223/pf0000247444

[111] United Nations Educational, Scientific, and Cultural Organization. 2017. Education for Sustainable Development Goals – Learning Objectives.

[112] Valgrind. [n. d.]. Valgrind. https://valgrind.org/ Accessed on: Sept., 30, 2024.

[113] Colin C. Venters et al. 2018. Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software* 138 (2018), 174–188. https://doi.org/10.1016/j.jss.2017.12.026

[114] Colin C. Venters et al. 2023. Sustainable software engineering: Reflections on advances in research and practice. *Information and Software Technology* 164 (2023), 107316. https://doi.org/10.1016/j.infsof.2023.107316

[115] Hanns-Carl Von Carlowitz. 1732. *Sylvicultura Oeconomica oder Haußwirthliche Nachricht und Naturmäßige Anweisung zur Wilden Baum-Zucht.* Vol. 1. Bey Johann Friedrich Brauns sel. Erben.

[116] Arnim Wiek, Michael J Bernstein, Rider W Foley, Matthew Cohen, Nigel Forrest, Christopher Kuzdas, Braden Kay, and Lauren Withycombe Keeler. 2015. Operationalising competencies in higher education for sustainable development. In *Routledge handbook of higher education for sustainable development.* Routledge, 265–284.

[117] Arnim Wiek, Lauren Withycombe, and Charles L Redman. 2011. Key competencies in sustainability: a reference framework for academic program development. *Sustainability science* 6 (2011), 203–218.