Chapter 11
Sustainable Software Engineering: Curriculum Development Based on ACM/IEEE Guidelines

Alok Mishra and Deepti Mishra

Abstract Climate change risk and environmental degradation are the most critical issues of our society. Our technology-influenced daily lifestyle involves many types of software and apps which are used by society at large, and their use is increasing more than ever before. Sustainability is a significant topic for future professionals and more so for software engineers due to its impact on society. It is crucial to motivate and raise concern among students and faculty members regarding sustainability by including it in the Software Engineering (SE) curriculum. This chapter discusses how sustainability can be included in various courses of the SE curriculum by considering ACM/IEEE curriculum guidelines for the SE curriculum, literature review, and various viewpoints so that SE students can attain knowledge on sustainable software engineering. It also includes an assessment of key competences in sustainability for proposed units in the SE curriculum.

11.1 Introduction

Software has become an integral part of our everyday life, gradually impacting human beings and society. The current industrial growth and the increasing adoption of ICT threaten the future of sustainability and cause environmental issues [1, 2]. Sustainability is becoming a crucial concern in information technology and software for our future. Sustainability management is one of the upcoming movements of the twenty-first century, but, until now, it is not getting as much attention from software engineering as it should. Furthermore, ICT has a major role in sustainable

A. Mishra
Molde University College, Molde, Norway

Department of Software Engineering, Atilim University, Ankara, Turkey
e-mail: alok.mishra@himolde.no; alok.mishra@atilim.edu.tr

D. Mishra (✉)
Department of Computer Science, Norwegian University of Science and Technology, Gjøvik, Norway
e-mail: deepti.mishra@ntnu.no

C. Calero et al. (eds.), *Software Sustainability*,
https://doi.org/10.1007/978-3-030-69970-3_11

development, specifically in software and green computing [3]. It is important that environmental concerns are addressed in the development, implementation, and operation of software. In this respect, the contribution of ICTs for energy and environmental sustainability has attracted attention of both researchers and professionals [1] as software contributes significantly to every aspect of our lives.

Dick et al. [4] proposed the first definitions for sustainable software and sustainable software engineering (SSE) in 2010, which have become the foundation for later explanations. According to Ray et al. [5], the term “sustainable” applies to both the longer life and greener aspects of software. Brooks et al. [6] suggest three dimensions to sustainability: environmental, economic, and social. These are interrelated and these should be selected in a way to attain optimum arrangement and alliance. Sustainable software is defined as the software whose direct and indirect negative influence on economy, society, human beings, and environment that result from development, implementation, and usage of the software are minimum [7]. Green software generates the minimum amount of e-waste during its operation and development [8].

Sustainable software engineering is developing software through a sustainable software engineering process which satisfies the purpose of sustainability by reducing the environmental impact of software implementation and operations to human beings and society at large. Sustainable software engineering (SSE) is based on the foundation of designing and developing software by taking into consideration various dimensions of sustainability which are economic, environmental, individual, social, and technical [9, 10]. A number of recent studies were performed to find out how sustainability is identified and included in software engineering process towards sustainable software development [11], which can reduce its environmental impact on society.

In many research studies, sustainability and energy efficiency are observed as crucial expertise for future software engineers [12–15]. However, a recent survey by Manotas et al. [16] of 3860 software professionals from Google, ABB, IBM, and Microsoft revealed that present higher educational programs do not prepare professionals to undertake sustainability, although they are inclined to learn about it [16]. Further, they noticed extensive significance of greenability and sustainability. Another study on teaching sustainability in software engineering also supports that sustainability is not included in the software engineering (SE) courses and that the present focal point is on energy efficiency issues [12, 14, 15]. Scientists have recently recognized that issues related to sustainable software engineering should be part of the discipline that has a significant role in the future of human beings.

Software engineers presently perform many tasks that may ensure sustainability, for instance, Agarwal et al. [3] consider the capabilities and benefits of green software and suggest more efficient algorithms will take less time to execute and that this will lead to sustainability. However, sustainability is considered as an additional feature in many software projects as software engineers are tied by time-to-market pressure and are often less inclined to administer sustainable methods and techniques [17]. For now, apart from cost, factors such as environment, social, and human sustainability are required to be considered in any planning, implementation, and running initiative related to software systems [18]. Organizations are now beginning to understand that not only cost efficiency, but also long-term and continued prosperity can be gained from sustainability. Therefore, apart from factors like cost, time, and quality, sustainability has become one of the significant objectives in developing, configuring, operating, and working software systems. Therefore, there is a need to support the transition to sustainability and incorporate it into software systems and other underlying business processes [19].

Green and Sustainable Software Engineering (GSSE) is the art of developing green and sustainable processes [20]. The objective of sustainable software process is to reduce the environmental impact of software solutions and their deployment on human beings, society, economy, and environment [21]. Presently, the effect of IT on sustainable advancement—in particular, of software—is an emerging issue due to the global concern on climate change. The education sector has to play a significant role in ensuring future software engineers understand sustainability dimensions and integrate them into the SE curriculum. Gibson et al. [22] supported that the educational sector has an important part to play in ensuring software professionals understand sustainability issues in software development. Therefore, there is a need to integrate sustainability in the software engineering discipline curricula. Gibson et al. [22] further observed that it is mentioned just once even in ACM/IEEE guidelines and twice in SWEBOK with respect to the software economics area. They argued in the current scenario there is need for sustainable software engineering education guidelines and components in such curricula for future software engineers. Therefore, this chapter advances effort in this direction. It extends our previous work [23] by following ACM/IEEE guidelines for SE curriculum along with authors' long academic experience to first list and categorize relevant SE courses offered in SE programs. Later sustainability-related units are introduced in existing SE courses followed by an assessment of these units with respect to key competences in sustainability.

The rest of the chapter is organized as follows: In Sect. 11.2 related work to describe the relationship between software quality and sustainability along with initiatives to include sustainability in SE higher education programs are presented. Section 11.3 introduces curricula development on sustainable software engineering. Section 11.4 includes points of discussion, concluding with a brief viewpoint for future direction.

11.2 Related Work

11.2.1 Software Quality and Sustainability

Sustainability is usually referred to as a nonfunctional requirement in software systems [19]. Nonfunctional requirements are also known as quality requirements. Although organizations have recognized that sustainability can be incorporated in quality issues, for instance, maintainability, usability, and agility, they could not do so due to time and budget constraints in software management [24].

Amri and Bellamine Ben Saoud [25] proposed Generic Sustainable Software Star Model (GS3M) to examine sustainable software and noticed some studies consider sustainability as a part of quality, while others observe quality and sustainability as different concepts and use quality attributes to support sustainability. Calero et al. [26] and Calero [27] applied the hypothesis that sustainability is a factor of the software quality, thus, unified it as a quality characteristic with three other subcharacteristics: energy consumption, resource optimization, and perdurability. Calero [26] also noticed that operationalization in this way includes introducing some modifications in the ISO quality standard ISO/IEC 25010 to support sustainability as a quality component. Albertao et al. [28] and Kern et al. [29] also identified quality attributes to define sustainability. Interestingly, Albertao et al. [28] formulated software project sustainability characteristics into development-related features (modifiability, reusability, portability, and supportability), usage-related attributes (performance, dependability, usability, and accessibility), and process-related attributes (predictability, efficiency, and project's footprint). Kern et al. [29] endorsed a quality model for sustainable software which constructs sustainability criteria into three categories: common quality criteria which are well-known and standardized issues (such as efficiency, reusability, modifiability, and usability); directly related benchmark (such as energy efficiency, framework entropy, functional types, hardware obsolescence, adaptability, feasibility, accessibility, usability, and organization's sustainability); and indirectly related yardstick that demonstrate the effects of software on other products and services and cover the effects of use as well as systemic effects, such as the fit for aim, elegance, and reflectivity.

11.2.2 Sustainability in SE Curricula

Sustainable software engineering is getting limelight among professionals and researchers [13, 30]. However, researchers have noted that sustainability is underrepresented in the curricula [11], hence the need to include the concept of sustainability in the university curriculum of computer science, software engineering, and information systems. Mann et al. [31] presented a framework for educators to design sustainability-centered education while Sammalisto and Lindhqvist [32] observed on the integration of sustainability in higher education based on different sustainability dimensions like environmental, economic, social, and technical. Gibson et al. [22] studied the significance of requirements engineering in ensuring sustainability in software development in the UK. Groher and Weinreich [33] studied how sustainability is perceived by software professionals in projects and found that professionals mainly linked it to maintainability and extensibility of software. Renzel et al. [34] contributed a detailed strategy for projects in sustainable software engineering.

Chitchyan et al. [35] reviewed sustainability related with Software Product Line Engineering (SPLE) and suggested the focus be on technical and social sustainability issues along with social sustainability related to organizations. Lutz et al. [36] also specified characteristics of sustainability in SPLE. Mohankumar and Anand Kumar [37] proposed a green-based model for sustainable software engineering. Recently Penzenstadler et al. [38] proposed a blueprint for a course on software engineering for sustainability.

11.2.3 Key Competencies in Sustainability

The major challenges in the incorporation of sustainability in the university education are in the field of teaching [39–42]. Therefore identifying key competences in sustainability may be the first step towards sustainability inclusion in higher education [43]. Wiek et al. [44] defines competence as a functionally linked complex of knowledge, skill, and attitude that enables successful task performance and problem solving. Competence is a quality developed through practice and not an end state [45].

In 2002, the Organisation for Economic Co-operation and Development (OECD) identified key competencies needed for an individual to lead an overall successful and responsible life and for contemporary society to face present and future challenges [46]. The OECD key competencies are divided into three categories: subject and methodological, social, and personal. The OECD study on key competencies and comprehensive educational objectives reveals sustainability's significance for the future [47]. Subsequently, multiple studies have introduced key competencies for education for sustainable development in the formal education sector to help assess the learning outcomes of pupils, and an overview is provided in the Table 11.1.

This study will assess proposed units integrated in SE courses based on the approach by Giangrande et al. [52] and Wiek et al. [44]. Wiek et al. [44] proposed a framework of key competencies in sustainability by categorizing competencies into clusters, which was found to be useful by Giangrande et al. [52] who further extended the framework.

11.3 Sustainable Software Engineering Curricula Outline

Sustainability knowledge should be integrated in a curriculum by linking the concept of sustainability to a particular field of study [32] rather than offering separate courses on sustainability. Considering the suggestion, Fig. 11.1 presents an approach to integrate sustainability education in SE curriculum. First, the ACM/IEEE guidelines for SE curriculum development has been followed to include an initial set of courses SE students should take in order to later practice their profession successfully. ACM/IEEE guidelines 2014 for SE curriculum development consists of a set of SE competencies that every SE graduate must possess and provides guidance to academic institutions and accreditation agencies about the knowledge and skills

Table 11.1 Key competencies for education for sustainable development

	Key competencies
De Haan [48]	Foresighted thinking Interdisciplinary work Cosmopolitan perception, transcultural understanding and cooperation Participatory skills Planning and implementation skills Empathy, compassion and solidarity Self-motivation and motivating others Distanced reflection on individual and cultural models
Barth [49]	Self-motivation Capacity for empathy, compassion and solidarity Reflection on individual Motivating others Participatory skills Foresighted thinking Interdisciplinary work Cosmopolitan perception, transcultural understanding and cooperation Reflection on cultural models Planning and implementation
Sleurs [50]	Values and ethics Emotions Systems thinking Knowledge Action
Roorda [51]	Responsibility Emotional intelligence Systems orientation Future orientation Personal involvement Action skills
Wiek [44]	Interpersonal Anticipatory Systemic working Normative Strategic
Giangrande [52]	Intrapersonal Interpersonal Future thinking Systems thinking Disciplinary and interdisciplinary Normative and cultural Strategic

fundamental to software engineering education [53]. Subsequently, additional courses have been included to reflect current advancements in SE education. Furthermore, the final set of courses are organized in different categories based on the structure of academic programs in major universities: fundamental courses of sustainability, core SE courses, technical electives, nontechnical electives, project-based courses, and industrial practice. Finally, the information gained from literature

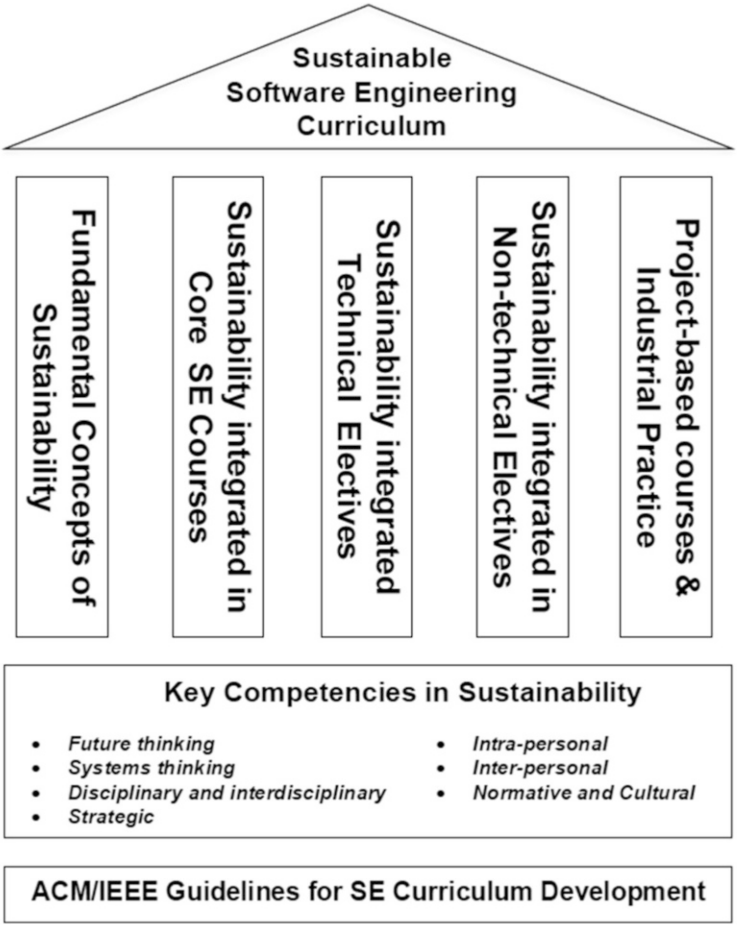
Sustainable
Software Engineering
Curriculum
Fundamental Concepts of Sustainability
Sustainability integrated in Core SE Courses
Sustainability integrated Technical Electives
Sustainability integrated in Non-technical Electives
Project-based courses & Industrial Practice
Key Competencies in Sustainability
Future thinking
Systems thinking
Disciplinary and interdisciplinary
Strategic
Intra-personal
Inter-personal
Normative and Cultural
ACM/IEEE Guidelines for SE Curriculum Development

Fig. 11.1 An approach to sustainability inclusion in the Software Engineering Curricula

review along with the authors' long academic experience in SE discipline facilitated the inclusion of sustainability competence in the form of flexible units within existing courses in the SE curricula.

The program should include the following units in the existing courses of the SE curricula so that students can get sufficient exposure to different components of sustainability issues in the software development life cycle.

11.3.1 Fundamental Concepts of Sustainability

Sustainability Theory Understanding the concept of sustainability in software and its various parts so as to be able to apply it in different stages of software development and deployment and operations stages in the organization.

Sustainability Analysis Understanding of rigorous analyses of sustainability issues in software development, from cost estimation to project management, software maintenance, and evolution. It should include, in general, software systems to be developed from a comprehensive perspective to sustainability and long-term consequences on the environment and society.

11.3.2 Core SE Courses

Software Requirements Engineering Sustainability inclusion in requirements elicitation and analysis process is crucial. Therefore, it is important to understand how to include sustainability during requirements elicitation process. Stakeholder modelling, goal modelling, and system modelling can assist in this part.

Software Architecture and Design How to apply sustainability in different kinds of software architecture and design issues, for instance database, human computer interaction, and modules interconnection, and in software architecture development.

Human-Computer Interaction Design Human-Computer Interaction (HCI) is part of many information technology and software applications. Therefore, sustainability issues should be included as a component in this course. Nyström and Mustaquim [54] suggested that persuasive system design can influence users to behave and live more sustainably and should be related to the sustainability of the environment. Sustainable HCI should address WCED's (World Commission on Environment and Development) sustainability view "... that it meets the needs of the present without compromising the ability of future generations to meet their own needs" [55]. Sustainable system design principles can be included in HCI, software system design, and industrial software development project curriculum.

Software Modelling and Analysis It is important that system modelling for complex software systems should be done from a sustainability perspective by using available tools. Software modelling and analysis (using UML diagrams) can assist in understanding how to incorporate sustainability into stakeholders' requirement scenarios. It should also include trade-offs and conflict resolution in the requirements of different stakeholders from a sustainability view.

Software Process Software process improvement should include sustainable software engineering processes along with Agile and DevOps approaches. It should also, knowledge of applicable tools, methods, and technologies to facilitate the sustainable software engineering processes. Energy and resource utilization are the main components that impact sustainability. Therefore, these should be determined from the initiation of the process. Eco-design of digital services to ensure reducing environmental impacts to develop digital services that are more sustainable, consume less resources and energy, and produce less waste. Further knowledge of relevant tools, methods, and technologies should be introduced to facilitate the sustainable software development process.

Software Verification and Validation An optimized approach in ensuring sustainability in software engineering is the software verification and validation process, including different types of testing and operation of the software product. Specification systems and automated verification tools can be helpful in this regard.

Software Quality Assurance Sustainability issue should be part of software process improvement and quality assurance process. Configuration management tools and software inspection tools can be complementary in this regard. Knowledge of standards of eco-design (ISO 14006, ISO 14062) should be imparted.

Software Project Management It includes the planning and controlling phases of sustainability activities along with sustainability policies to ensure an efficient process. Appropriate project management tools and agile methods management tools can facilitate in ensuring sustainability practices in software project management. Eco-design of digital services towards ensuring reduced environmental impacts to develop digital services that are more sustainable, consume less resources and energy, and produce less waste.

Software Construction and Evolution Software evolution is a continuous process. Refactoring tools, automated testing tools, and configuration management tools along with project management tools aid in ensuring sustainability in software construction and evolution.

Software Security Security and safety during the development of complex software systems is crucial. So security and safety are now an integral part of the set of nonfunctional requirements which lead to software quality. ISO/IEC 25010:2011 included safety as an explicit characteristic in software while ISO/IEC 9126-1:2001 ensures security in software. Safety and security are called out and treated specifically because they are significant characteristics. Penzenstadler et al. [56] supported that the same is true for sustainability, specifically the dimension of environmental sustainability, and there is a need to find suitable means to analyze, support, verify, and validate sustainability requirements in software engineering.

11.3.3 Technical Elective Courses

Internet of Things (IoT) IoT has the ability to combat climate change towards green environment. It could impact sustainability in different areas, such as water use and energy efficiency. According to the World Economic Forum, IoT could be a game changer for sustainability [57]. IoT helps in applying waste management strategies and in circular economy. IoT deployments can help in addressing many of the Sustainable Development Goals (SDGs) of the UN. IoT technology can provide tangible benefits to sustainability [57]. Many IoT initiatives may help accomplish sustainability in the future [58]. Therefore students should be made aware of such IoT applications that can be applied to achieve sustainability by including relevant case studies, white paper, discussion, seminar, etc. in the curriculum.

Cloud Computing Cloud computing provides more efficient use of computing power and is advantageous for environmental sustainability. Application of cloud computing ensures social, business, and environmental sustainability. It can include discussions, case studies, seminars, projects, company visits, etc.

Web and Mobile Systems Sustainability and page speed are correlated. When your website runs more efficiently it consumes less processing power thus less energy and leaves a lower carbon footprint [59]. Also, a sustainable design is more efficient and accessible. Sustainable mobile apps and their users may contribute to achieving environmental goals, and mobile devices are enablers of sustainable actions due to their huge potential for scalability [60]. Mobile applications that have even a little effect on resource efficiency or the reduction of greenhouse gas emissions could result in a greater impact as these are used every day [60]. This can include concepts on social software interface with sustainability issues, green software development and usage practices, and promotion of technologies, development frameworks, and tools which facilitate sustainability in web and mobile systems development. These could be included as real-life projects, cases studies, and seminars and lectures from industry practitioners.

Sustainable Data Center Green data centers or sustainable data centers help in reducing carbon footprint, design and deployment of data store, and applications to operate in energy-efficient ways. Therefore, the course should include real-life case studies, seminars, and discussions on how sustainability can be incorporated in this regard.

Tools for Software Sustainability Tools must be introduced to assist different stages of software development (requirements, design, testing, configuration management, etc.) towards ensuring sustainability. This can be a part of a sustainable or green software engineering laboratory program.

11.3.4 Nontechnical Elective Courses

Global Professional Practice/Social Responsibility Students should be aware how carbon footprint, CO_2 emissions, global warming is a matter of concern. Therefore global professional practice should include environmental issues arising from software engineering products and their use. These can be included as case studies, seminars, and group discussions to analyze environmental degradation cases and to explore mitigation plans, global environmental challenges, sustainable software, energy management, and green computing standards in the context of software applications.

11.3.5 Project-Based Courses and Industrial Practice/ Internships

Project-based courses Most universities have final-year projects or thesis for students to explore real-world challenges. Universities sometimes also require their students to do industrial internships of 1–2 months so that students can gain experience in professional projects in real-world settings. Sustainability can be included as a learning outcome for such courses. Projects involving sustainability in software engineering during summer internships or such mini projects should be part of the course.

Table 11.2 presents how the integration of these units into current curriculum will help SE professionals to acquire key competencies in sustainability.

11.4 Discussion

Sustainable software engineering is an emerging paradigm and significant for society in terms of the environment. Sammalisto and Lindhqvist [32] argued that a proper feedback system is required between educators and university administrators to show the value and significance of the integration of sustainability. Torre et al. [11] observed a top 10 universities curriculum analysis that none of the engineering courses explicitly addresses sustainable software engineering or the status of green sustainable software engineering. The vast majority of the survey respondents (97%) expressed there is need for more courses related to sustainability.

The present industrial production and increasing use of ICT may endanger prospective sustainability and lead to environmental problems [1, 2]. In a recent study, Salam and Khan [61] classified 20 success elements towards the evolution of green and sustainable software. Out of these, green software design and efficient coding was found to be the most significant factor (71%) followed by power-saving software methods (70%). Mahmoud and Ahmad [20] proposed green model for sustainable software engineering. Naumann et al. [7] proposed sustainable software engineering process and quality models and suggested nine successive stages: Requirements, Design, Unit Testing, Implementation, System Testing, Green Analysis, Usage, Maintenance, and Disposal. Lami et al. [62] found that sustainability-related processes are missing in ISO/IEC 12207 and proposed three processes:

Table 11.2 Assessment of key competencies in sustainability with respect to proposed units in SE curriculum

	Key competencies in sustainability [44, 52]						
Units	Intrapersonal	Interpersonal	Future thinking	Systems thinking	Disciplinary and interdisciplinary	Normative and cultural	Strategic
Sustainability Theory	X		X	X	X		X
Sustainability Analysis	X	X	X	X	X		X
Software Requirements Engineering	X	X	X	X	X		X
Software Architecture and Design		X	X	X	X		X
Human-computer interaction design		X	X	X	X	X	X
Software Modelling and Analysis		X	X	X	X		X
Software Process		X	X	X	X		X
Software Verification and Validation		X	X	X			X
Software Quality Assurance		X	X	X			X
Software Project Management		X	X	X			X
Software Construction and Evolution		X	X	X	X		X
Software Security		X	X	X			X
Internet of Things (IoT)			X	X	X		X
Cloud Computing			X	X	X		X
Web and Mobile Systems		X	X	X	X		X
Sustainable Data Center			X	X			X
Tools for Software Sustainability		X	X	X	X	X	X
Global Professional Practice/Social Responsibility	X	X	X	X	X	X	X
Project-Based Courses		X	X	X	X	X	X

Sustainability Management Processes, Sustainability Engineering Process, and Sustainability Qualification Process.

The purpose of SSE is to curtail the energy footprint of computers as well as minimize other environmental impacts related to software systems. Software is now a pervasive part of the society as even mobile phone and social media users are in billions. It is the responsibility of software engineering educators to prepare SE professionals by equipping them with skills to meet the expectations of the software industry [63]. Therefore, it is significant to include sustainability in courses for future software engineers so that it can be achieved while developing, deploying, and maintaining all kinds of software in the future. Professional practices should be part of the SE curriculum [64], which can include a sustainability component. Moreover, sustainability has the potential to attract more students to the SE discipline due to its indispensable significance for the future [65]. Programs that address environmental sustainability in information technology are sometimes also referred to as green information technology. Green IT refers to information technology and system initiatives and programs that address environmental sustainability [66] and manage energy consumption as well as waste associated with the use of hardware and software, which tend to have a direct and positive impact on sustainability [67].

The proposed curriculum development can be easily customized and introduced as part of an undergraduate- or graduate-level software engineering curriculum. Since only a limited number of undergraduate and graduate programs on sustainability have been introduced in the last decade in certain institutions, the curricula proposed here can be a useful contribution to the body of knowledge for software engineering educators. As requirement specifications are the base input for software architecture and design, they have an impact on sustainability. With increasing global concern regarding climate change, the time has come to include “Sustainability” as a nonfunctional requirement towards quality software for future generations.

11.5 Conclusion and Outlook

Due to climate changes in the last decade and proliferation of information technology, software, and apps in daily life, there is a crucial need to develop and deploy green software. Therefore, there is a need to train future software engineers in such a manner that they will be able to include sustainability in each stage of the software development life cycle. Here, important units of sustainability inclusion in software engineering curricula have been described according to the recent ACM/IEEE curriculum guidelines for SE curriculum along with literature review on sustainable software engineering approaches, concepts, and tools. Software engineering undergraduate and graduate programs should include at least one foundation course on sustainability in their curriculum. This chapter also included appraisal of key competencies in sustainability for proposed units in SE curriculum.

This work can be extended by a survey and interviewing software engineering professionals to know in a more detailed manner how the SSE course can be developed and improved in the future into a more practice-oriented approach so that future software engineers will be able to produce eco-friendly and sustainable software.

References

1. Cai S, Chen X, Bose I (2013) Exploring the role of IT for environmental sustainability in China: an empirical analysis. Int J Prod Econ s146(2):491–500
2. Sissa G (2010) Green software. UPGRADE: Eur J Inf Prof 11:53–63
3. Agarwal S, Nath A, Chowdhury D (2012) Sustainable approaches and good practices in green software engineering. Int J Res Rev Comput Sci 3(1):1425
4. Dick M, Naumann S (2010) Enhancing software engineering processes towards sustainable software product design. In: EnviroInfo. pp 706–715
5. Ray S (2013) Green software engineering process: moving towards sustainable software product design. J Glob Res Comput Sci 4(1):25–29
6. Brooks S, Wang X, Sarker S (2012) Unpacking green IS: a review of the existing literature and directions for the future. In: Green business process management. Springer, pp 15–37
7. Naumann S, Dick M, Kern E, Johann T (2011) The greensoft model: a reference model for green and sustainable software and its engineering. Sustain Comput Inf Syst 1(4):294–304
8. Erdelyi K (2013) Special factors of development of green software supporting eco sustainability. In: 2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, pp 337–340
9. Becker C et al (2015) Requirements: The key to sustainability. IEEE Softw 33(1):56–65
10. Penzenstadler B (2013) Towards a definition of sustainability in and for software engineering. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. pp 1183–1185
11. Torre D, Procaccianti G, Fucci D, Lutovac S, Scanniello G (2017) On the presence of green and sustainable software engineering in higher education curricula. In: 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM). IEEE, pp 54–60
12. Lago P, Damian D (2015) Software engineering in society at ICSE. STC Sustain Computing Newsl 4(1)
13. Lago P, Kazman R, Meyer N, Morisio M, Müller HA, Paulisch F (2013) Exploring initial challenges for green software engineering: summary of the first GREENS workshop, at ICSE 2012. ACM SIGSOFT Softw Eng Notes 38(1):31–33
14. Pang C, Hindle A, Adams B, Hassan AE (2015) What do programmers know about software energy consumption? IEEE Softw 33(3):83–89
15. Penzenstadler B, Fleischmann A (2011) Teach sustainability in software engineering? In: 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp 454–458
16. Manotas I et al (2016) An empirical study of practitioners' perspectives on green software engineering. In: 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE). IEEE, pp 237–248
17. Durdik Z, Klatt B, Koziolek H, Krogmann K, Stammel J, Weiss R (2012) Sustainability guidelines for long-living software systems. In: 2012 28th IEEE International Conference on Software Maintenance (ICSM). IEEE, pp 517–526
18. Raisian K, Yahaya J, Deraman A (2016) Current challenges and conceptual model of green and sustainable software engineering. J Theor Appl Inf Technol 94:428–443
19. Betz S, Caporale T (2014) Sustainable software system engineering. In: 2014 IEEE Fourth International Conference on Big Data and Cloud Computing. IEEE, pp 612–619

20. Mahmoud SS, Ahmad I (2013) A green model for sustainable software engineering. Int J Softw Eng Applic 7(4):55–74
21. Amsel N, Ibrahim Z, Malik A, Tomlinson B (2011) Toward sustainable software engineering: NIER track. In: 2011 33rd international conference on software engineering (ICSE). IEEE, pp 976–979
22. Gibson ML et al (2017) Mind the chasm: a UK fisheye lens view of sustainable software engineering
23. Mishra A, Mishra D (2020) Sustainable software engineering education curricula development. Int J Inf Technol Secur 12(2):47–56
24. Chitchyan R et al (2016) Sustainability design in requirements engineering: state of practice. In: Proceedings of the 38th International Conference on Software Engineering Companion. pp 533–542
25. Amri R, Saoud NBB (2014) Towards a generic sustainable software model. In: 2014 Fourth International Conference on Advances in Computing and Communications. IEEE, pp 231–234
26. Calero C, Bertoa MF, Moraga MÁ (2013) Sustainability and quality: icing on the cake. In: RE4SuSy@ RE. Citeseer
27. Calero C (2013) Sustainability as a software quality factor. In: Proceedings of the IBM Conference Day
28. Albertao F, Xiao J, Tian C, Lu Y, Zhang KQ, Liu C (2010) Measuring the sustainability performance of software projects. In: 2010 IEEE 7th International Conference on E-Business Engineering. IEEE, pp 369–373
29. Kern E, Dick M, Naumann S, Guldner A, Johann T (2013) Green software and green software engineering–definitions, measurements, and quality aspects. In: First International Conference on Information and Communication Technologies for Sustainability (ICT4S2013), 2013b ETH Zurich. pp 87–91
30. Naumann S, Kern E, Dick M, Johann T (2015) Sustainable software engineering: process and quality models, life cycle, and social aspects. In: ICT innovations for sustainability. Springer, pp 191–205
31. Mann S, Muller L, Davis J, Roda C, Young A (2010) Computing and sustainability: evaluating resources for educators. ACM SIGCSE Bull 41(4):144–155
32. Sammalisto K, Lindhqvist T (2008) Integration of sustainability in higher education: a study with international perspectives. Innov High Educ 32(4):221–233
33. Groher I, Weinreich R (2017) An interview study on sustainability concerns in software development projects. In: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp 350–358
34. Renzel D, Koren I, Klamma R, Jarke M (2017) Preparing research projects for sustainable software engineering in society. In: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS). IEEE, pp 23–32
35. Chitchyan R, Groher I, Noppen J (2017) Uncovering sustainability concerns in software product lines. J Softw Evol Process 29(2):e1853
36. Lutz R, Weiss D, Krishnan S, Yang J (2010) Software product line engineering for long-lived, sustainable systems. In: International Conference on Software Product Lines. Springer, pp 430–434
37. Mohankumar M, Kumar MA (2016) Green based software development life cycle model for software engineering. Indian J Sci Technol 9(32):1–8
38. Penzenstadler B et al (2018) Blueprint and evaluation instruments for a course on software engineering for sustainability. arXiv preprint arXiv:1802.02517
39. Buckler C, Creech H (2014) Shaping the future we want: UN Decade of Education for Sustainable Development; final report. UNESCO
40. Lazzarini B, Perez-Foguet A, Boni A (2018) Key characteristics of academics promoting Sustainable Human Development within engineering studies. J Clean Prod 188:237–252
41. Mulder KF, Segalàs J, Ferrer-Balas D (2012) How to educate engineers for/in sustainable development. Int J Sustain Higher Educ

42. Wals AE (2014) Sustainability in higher education in the context of the UN DESD: a review of learning and institutionalization processes. J Clean Prod 62:8–15
43. Mishra D, Mishra A (2020) Sustainability inclusion in informatics curriculum development. Sustainability 12(4):5769
44. Wiek A, Withycombe L, Redman CL (2011) Key competencies in sustainability: a reference framework for academic program development. Sustain Sci 6(2):203–218
45. Vare P et al (2019) Devising a competence-based training program for educators of sustainable development: lessons learned. Sustainability 11(7):1890
46. Rychen DS, Salganik LH (2002) Definition and Selection of Competencies (DESECO): theoretical and conceptual foundations. Strategy paper. Swiss Federal Statistical Office, Neuchatel, Switzerland
47. De Haan G (2010) The development of ESD-related competencies in supportive institutional frameworks. Int Rev Educ 56(2–3):315–328
48. De Haan G (2006) The BLK '21'programme in Germany: a 'Gestaltungskompetenz'-based model for Education for Sustainable Development. Environ Educ Res 12(1):19–32
49. Barth M, Godemann J, Rieckmann M, Stoltenberg U (2007) Developing key competencies for sustainable development in higher education. Int J Sustain Higher Educ
50. Sleurs W (2008) Competencies for ESD (Education for Sustainable Development) teachers. A framework to integrate ESD in the curriculum of teacher training institutes. CSCT Project (Comenius 2.1 project 118277-CP-1-2004-BE-Comenius-C2.1), Brussels, Belgium
51. Roorda N (2010) Sailing on the winds of change: the Odyssey to sustainability of the universities of applied sciences in the Netherlands. Doctoral dissertation, Maastricht University
52. Giangrande N et al (2019) A competency framework to assess and activate education for sustainable development: addressing the UN sustainable development goals 4.7 challenge. Sustainability 11(10):2832
53. Ardis M, Budgen D, Hislop GW, Offutt J, Sebern M, Visser W (2015) SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering. Computer 11:106–109
54. Nyström T, Mustaquim MM (2014) Sustainable information system design and the role of sustainable HCI. In: Proceedings of the 18th International Academic MindTrek Conference: Media Business, Management, Content & Services. pp 66–73
55. B Commission (1987) Report of the World Commission on Environment and Development: our common future, vol 10. [Online]. https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf
56. Penzenstadler B, Raturi A, Richardson D, Tomlinson B (2014) Safety, security, now sustainability: the nonfunctional requirement for the 21st century. IEEE Softw 31(3):40–47
57. Arias R, Lueth K, Rastogi A (2018) The effect of the Internet of Things on sustainability. In: World Economic Forum. https://www.weforum.org/agenda/2018/01/effect-technology-sustainability-sdgs-internet-thingsiot/. Accessed 14 Mar 2019
58. Lazarevich K (2018) 10 IoT initiatives for a more sustainable future. [Online]. https://www.iotforall.com/10-iot-environment-initiatives-sustainable-future/
59. The building blocks of sustainable web design. https://sustainablewebdesign.org/. Accessed 3 Mar 2020
60. Brauer B, Ebermann C, Hildebrandt B, Remané G, Kolbe LM (2016) Green by app: the contribution of mobile applications to environmental sustainability. In: Pacific Asia Conference On Information Systems (PACIS). Association for Information System
61. Salam M, Khan SU (2016) Developing green and sustainable software: success factors for vendors. In: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE, pp 1059–1062
62. Lami G, Fabbrini F, Fusani M (2012) Software sustainability from a process-centric perspective. In: European Conference on Software Process Improvement. Springer, pp 97–108
63. Mishra A, Ercil Cagiltay N, Kilic O (2007) Software engineering education: some important dimensions. Eur J Eng Educ 32(3):349–361

64. Mishra A, Mishra D (2012) Industry oriented advanced software engineering education curriculum. Croat J Educ 14(3):595–624
65. Özkan B, Mishra A (2015) A curriculum on sustainable information communication technology. Problemy Ekorozwoju–Prob Sustain Dev 10(2):95–101
66. Mishra A, Akman I (2014) Green information technology (GIT) and gender diversity. Environ Eng Manag J 13(12)
67. Mishra A, Yazici A, Mishra D (2012) Green information technology/information system education: curriculum views. TTEMTechnics Technol Educ Manag 7(3):679–686