DOI: https://doi.org/10.31410/ERAZ.2022.317

Sustainability in Software Engineering: A Design Science Research Approach

Clara Silveira[1]
Leonilde Reis[2]

Received: June 30, 2022
Accepted: August 23, 2022
Published: December 30, 2022

Keywords:
Artefact;
Sustainability dimensions;
Sustainable development goals;
Software process;
Methodology

Abstract: *In the current global context, with so many challenges to be faced, it is important to see people's increased interest in sustainability issues as an opportunity for change. Sustainable Software Engineering, as a recent research area, incorporates sustainability principles and dimensions in the software development process. On the other hand, the Design Science Research methodology has become a well-received research paradigm in Information Systems in general and in Software Engineering in particular. The paper presents a Sustainable Software Engineering approach integrated into the Design Science Research methodology. The concepts of sustainability in software development, namely the principles of the Karlskrona Manifesto, the principles of Green Software Engineering and the Sustainable Development Goals are integrated into the approach. Preliminary results from applying the approach indicate that the iterative process of the Design Science Research methodology allows for the integration of multidisciplinary sustainability artefacts during the software process.*

1. INTRODUCTION

Software Engineering is an area of knowledge in constant evolution facing new challenges and sustainability is now seen as a central concern. The use of Information and Communication Technologies (ICT) can enhance the transformation of Society by incorporating sustainability concerns, including the Sustainable Development Goals (SDG).

The paper presents an approach that integrates the concepts of sustainable Software Engineering in the phases of the Design Science Research methodology for the construction of the artefact. These concepts of sustainability in the software process include the principles and commitments of the Karlskrona Manifesto, the principles of Green Software Engineering, and the 17 United Nations Sustainable Development Goals. The principles of the Manifesto for Energy-aware Software were also considered.

The work developed in this paper is structured in six sections: introduction to the problem under study, the research methodology, the main topics of the background research, approach design to sustainability software engineering, future research directions and conclusion.

2. METHODOLOGY

The research methodology adopted is Design Science Research. The DSR methodology is a tool to improve methods in software engineering research and to offer specific guidelines for the development and evaluation of artefacts. With reference to the characteristics of the five types/genres (Peffers, Tuunanen, & Niehaves, 2018) of research approaches identified in the literature, the Design Science Research methodology is used from Peffers' perspective, as it includes the

[1] Polytechnic Institute of Guarda, Guarda, Portugal
[2] Polytechnic Institute of Guarda, Guarda, Portugal

necessary characteristics to be applied in Software Engineering, namely: focus on the development of practical artefacts; flexible and iterative process; the role of theory: generalizability, reasoned argument that an artefact can work; evaluation: outcome-oriented, practical.

The process of the DSR methodology is shown in Figure 1. The Design Science Research methodology includes six steps (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007): problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication.

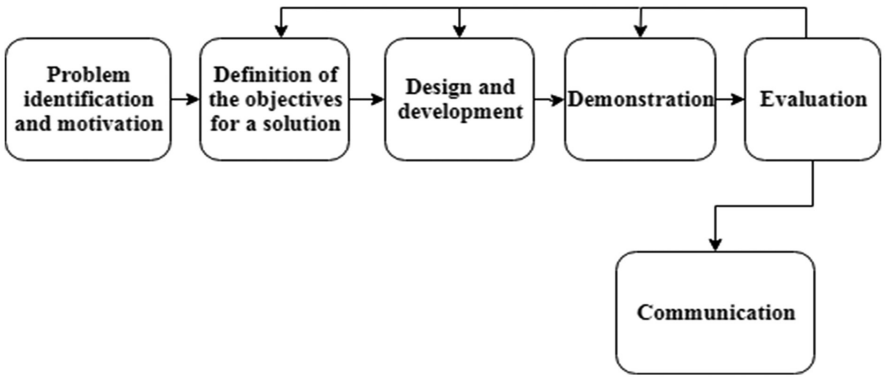

Figure 1. Design Science Research Process
Source: Adapted from (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

3. BACKGROUND RESEARCH

The study of sustainability in the field of software development is very important in order to promote the integration of the 17 SDGs outlined by the United Nations. On 25 September 2015, the United Nations General Assembly adopted the 2030 Agenda for Sustainable Development (UNDP, 2022), setting out 17 goals and 169 targets, covering social, economic and environmental dimensions around the world (Figure 2).

Figure 2. 17 Sustainable Development Goals
Source: https://sdgs.un.org/goals

The SDGs focus on people, human rights and responding to growing social inequalities, as well as core concerns such as peace, security and climate change. The SDG fundamentally targets the "5P": People, Prosperity, Partnership, Peace and Planet.

In the study (Wu, Guo, Huang, Liu, & Xiang, 2018) on the correlations between the SDG and ICT, several gaps were identified in the three pillars of sustainable development: social, economic and environmental perspectives. There is therefore a need to raise awareness and draw attention on how to innovate in ICT to better help all nations achieve the SDG by 2030. In this sense, ICT can enhance the transformation of Society by incorporating sustainability concerns, including the SDG. In this sense, it becomes essential to include sustainability in software development.

P#	Principles descriptions
P1	**Sustainability is systemic:** Sustainability is never an isolated property. Systems thinking has to be the starting point for the transdisciplinary common ground of sustainability.
P2	**Sustainability has multiple dimensions:** We have to include those dimensions into our analysis if we are to understand the nature of sustainability in any given situation.
P3	**Sustainability transcends multiple disciplines:** Working in sustainability means working with people from across many disciplines, addressing the challenges from multiple perspectives.
P4	**Sustainability is a concern independent of the purpose of the system:** Sustainability has to be considered even if the primary focus of the system under design is not sustainability.
P5	**Sustainability applies to both a system and its wider contexts**: There are at least two spheres to consider in system design: the sustainability of the system itself and how it affects sustainability of the wider system of which it will be part.
P6	**Sustainability requires action on multiple levels:** Some interventions have more leverage on a system than others. Whenever we take action towards sustainability, we should consider opportunity costs: action at other levels may offer more effective forms of intervention.
P7	**System visibility is a necessary precondition and enabler for sustainability design**: The status of the system and its context should be visible at different levels of abstraction and perspectives to enable participation and informed responsible choice.
P8	**It is possible to meet the needs of future generations without sacrificing the prosperity of the current generation:** Innovation in sustainability can play out as decoupling present and future needs. By moving away from the language of conflict and the trade-off mindset, we can identify and enact choices that benefit both present and future.
P9	**Sustainability requires long-term thinking**: We should assess benefits and impacts on multiple timescales, and include longer-term indicators in assessment and decisions.

Figure 3. Principles of KarlskronaManifesto

Source: Adapted from (Becker et al., 2015)

The concept of sustainability in Software Engineering is often referred to following the Karlskrona Manifesto (Becker et al., 2015). This manifesto serves as a guide for designing and developing more sustainable software systems. Figure 3 shows the nine principles of the Manifesto.

Sustainable software engineering is a means of developing sustainable software in which the process activities balance the various dimensions of sustainability, namely, the individual, social, economic, environmental, and technical dimensions. Figure 4 illustrates these five dimensions from the perspective of the Karlskrona Manifesto (Becker et al., 2015).

It is considered that, concerning the dimensions of sustainability (Figure 4), ICT can enhance the incorporation of these dimensions to include concerns in the field of software reuse, open-source, cloud computing, virtualisation, process dematerialisation, digital transformation and information security (Reis, Carvalho, Silveira, Marques, & Russo, 2021).

The authors (Fonseca, Kazman, & Lago, 2019), advocate that this may be an opportunity for the R&D community to apply the nine principles of the "Manifesto for energy-aware software" to foster energy awareness as an ICT sub-discipline, thus enhancing software engineering practices. In this sense, it is considered that successful application may depend on three fundamentals: awareness; the creation of a body of engineering knowledge; education and training.

Sustainability dimensions

- **Individual**: refers to the well-being of humans as individuals. This includes mental and physical well-being, education, self-respect, skills, mobility, etc.;
- **Social**: concerned with societal communities (groups of people, organizations) and the factors that erode trust in society. This dimension includes social equity, justice, employment, democracy, etc.;
- **Economic**: focused on assets, capital and added value. This includes wealth creation, prosperity, profitability, capital investment, income, etc.
- **Environmental**: concerned with the long term effects of human activities on natural systems. This dimension includes ecosystems, raw resources, climate change, food production, water, pollution, waste, etc.;
- **Technical** : refers to longevity of information, systems, and infrastructure and their adequate evolution with changing surrounding conditions.

Figure 4. Sustainability Dimensions

Source: Adapted from (Becker et al., 2015)

Green Software Engineering is an emerging discipline that is based on the eight Green Software Engineering Principles (Figure 5) in order to list the necessary skills to define, build and execute applications interconnecting software, hardware and data center design practices and architectures, among other areas (Microsoft, 2022). These principles include concerns about creating and optimizing applications that are carbon and energy efficient, reducing the amount of data and the distance it must travel across the network.

Principles of Green Software Engineering

1. Carbon: Build applications that are carbon efficient.
2. Electricity: Build applications that are energy efficient.
3. Carbon Intensity: Consume electricity with the lowest carbon intensity.
4. Embodied Carbon: Build applications that are hardware efficient.
5. Energy Proportionality: Maximize the energy efficiency of hardware.
6. Networking: Reduce the amount of data and distance it must travel across the network.
7. Demand Shaping: Build carbon-aware applications.
8. Optimization: Focus on step-by-step optimizations that increase the overall carbon efficiency.

Figure 5. Green Software Engineering Principles

Source: Microsoft, 2022.

4. APPROACH DESIGN

The concepts of sustainability in software development, namely the principles and commitments of the Karlskrona Manifesto, the principles of Green Software Engineering, the principles of the Manifesto for Energy-aware Software, and the 17 United Nations Sustainable Development Goals are considered in the steps of the DSR methodology for the construction of the artefact. Table 1 presents the phases of the DSR methodology with the incorporation of the principles of the Karlskrona Manifesto described in Figure 3.

Table 1. Karlskrona manifest principles applied in the DSR phases

DSR Phase	Karlskrona Manifesto principles that apply
Problem identification	P1, P3 and P5- Ensuring that sustainability is considered from the moment the problem is identified, making it possible to develop a solution with systemic value. The analysis of the problem involves elements from several disciplines with transversal knowledge of the processes that must incorporate sustainability concerns. Sustainability applies to both a system and its wider contexts: assessing the impacts arising from the definition of the problem and the environment in which the solution will operate.
Definition of the objectives	P1, P2 and P3- Sustainability is systemic and has multiple disciplines and dimensions: Software sustainability has different disciplines/dimensions that must be considered when defining the objectives of a solution.
Design and development	P2 and P4- Sustainability has multiple dimensions and is independent of the purpose of the system: during the creation of the artefact, consider the different dimensions of sustainability (technical, social, environmental, economic and individual) to determine the desired functionality.
Demonstration	P2, P6 and P7- When demonstrating the artefact, it must have implicit actions that promote the inclusion of the various dimensions of sustainability, namely: technical, economic, social, environmental and human. This way it will allow the visibility of sustainability actions at different levels of abstraction.
Evaluation and communication	P6, P8 and P9- Involve the community in evaluating the adequacy of the artefact in solving the problem and scientifically disseminating the added value of the solution.

Table 1 can be extended to incorporate the SDG, the principles of Green Software Engineering and the principles of the Manifesto for Energy-aware Software. In this perspective, this approach proposes the application of the phases of the DSR methodology incorporating the principles and dimensions of sustainability in order to create artefacts that assertively contribute to solving the various problems, given the current challenges of sustainability (Figure 6).

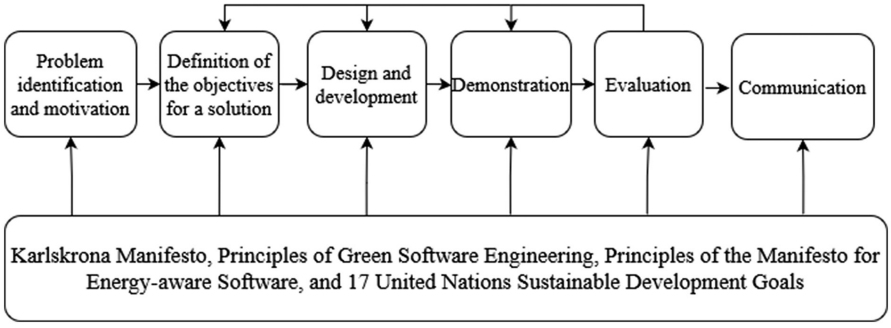

Figure 6. DSR methodology incorporating the principles and dimensions of sustainability
Source: Adapted from (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

The DSR methodology incorporating the principles and dimensions of sustainability is a tool that offers specific guidelines for the development and evaluation of artifacts (namely software products).

5. FUTURE RESEARCH DIRECTIONS

Previous studies on the integration of the principles and commitments of the Karlskrona Manifesto in software development (Ovelheiro, Silveira, & Reis, 2021; Silveira, & Reis, 2021), made it possible to explain to developers and stakeholders which principles of sustainability to apply in the implementation of the use case. Also (Reis, & Silveira, 2020) show that it is possible to assess the implementation of awareness-raising actions through the SDG.

In future work, we intend to continue teaching the incorporation of the principles and dimensions of sustainability in software development and include applying the approach to academic co-creation projects with organizations.

6. CONCLUSION

Sustainability in ICT, and more specifically in Software Engineering, was the main concern of this paper. Software Engineering can present itself as an interesting contribution with regard to the inclusion of sustainability, namely by incorporating sustainability principles and dimensions in the software development process. The developed approach integrates sustainable Software Engineering with the Design Science Research methodology, allowing the creation of artifacts that meet sustainability.

Preliminary results from applying the approach indicate that the iterative process of the Design Science Research methodology allows for the integration of multidisciplinary sustainability artefacts during the software development process.

Sustainability concepts in the software development process may include the principles and commitments of the Karlskrona Manifesto, the principles of Green Software Engineering, the principles of the Manifesto for Energy-aware Software, and the United Nations' 17 Sustainable Development Goals.

REFERENCES

Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., & Venters, C. (2015). Sustainability Design and Software: The Karlskrona Manifesto. *37th International Conference on Software Engineering (ICSE 15).*

Fonseca, A., Kazman, R., & Lago, P. (2019). A manifesto for energy-aware software. *IEEE Software*, *36*(6), 79-82.

Microsoft. (2022). Principles of Green Software Engineering, *Microsoft*, https://docs.microsoft.com/pt-pt/learn/modules/sustainable-software-engineering-overview/.

Ovelheiro, B., Silveira, C., & Reis, L. (2021). Sustainability Design Applied to the Digital Signature of Documents. Em *In Carvalho, L. C., Reis, L., Prata, A., & Pereira, R. (Ed.), Handbook of Research on Multidisciplinary Approaches to Entrepreneurship, Innovation, and ICTs* (pp. 349-374). IGI Global. http://doi:10.4018/978-1-7998-4099-2.ch016 .

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45-78.

Peffers, K., Tuunanen, T., & Niehaves, B. (2018). Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems, 27:2*, 129-139.

Reis, L., Carvalho, L., Silveira, C., Marques, A., & Russo, N. (2021). *Inovação e Sustentabilidade em TIC.* Silabo. ISBN 978-989-561-146-1.

Reis, L., & Silveira, C. (2020). Multidisciplinary Sustainability - The multiple perspectives of a Social Organization, In Proceedings *6th International Scientific-Business Conference Leadership, Innovation, Management and Economics: Integrated Politics of Research – LIMEN 2020*. DOI: https://doi.org/10.31410/LIMEN.2020.179

Silveira, C., & Reis, L. (2021). Sustainability in Information and Communication Technologies. Em L. C. Carvalho, *Handbook of Multidisciplinary Approach to Entrepreneurship, Innovation, and ICTs* (pp. 375-396). USA: IGI Global. doi: http://doi:10.4018/978-1-7998-4099-2.ch017

UNDP (2022). Sustainable Development Goals, *United Nations Development Programme*: www.undp.org/content/undp/en/home/sustainable-development-goals.html.

Wu, J., Guo, S., Huang, H., Liu, W., & Xiang, Y. (2018). Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives. *IEEE Communications Surveys Tutorials 20(3)*, 2389–2406.