Teaching and Assessing Sustainability Based on the Karlskrona Manifesto

Dr. Ing. Ivan Cabezas, Universidad de San Buenaventura

Ivan Cabezas was born in Colombia in 1973. He received the B. Eng. in Computer Science and the Engineering Ph. D. degrees from Universidad del Valle, in 2004 and 2013, respectively. He is a member of IEEE and ASEE. Engineering education and sustainability concerns during the software engineering design process are among his research interests. He has been working as a full-time professor in the Software Systems Engineering program at the Engineering School of the Universidad de San Buenaventura - Cali, in Colombia, since 2014.

Eileen Webb, Accreditation Preparation

Eileen Webb is president of Accreditation Preparation which has helped over 100 programs at over 20 universities with their ABET accreditation in the United States, Mexico, Colombia and Portugal since 2012. She is also president of Streamline Consulting, an industrial engineering firm serving manufacturers, casinos and government clients. Former employers include Texas Instruments, Raytheon, Procter and Gamble, Shedd's Food Products, Weyerhaeuser Paper, ABB, and others. She has been an invited speaker at the ABET Symposium, World Engineering Education Forum, LACCEI (Latin America and Caribbean Consortium of Engineering Institutions) and Simposium Assessment in Barranquilla, Colombia Her bachelor of chemical engineering is from Georgia Institute of Technology (Georgia Tech.)

Teaching and assessing sustainability based on the Karlskrona Manifesto

Abstract

Sustainability is not a new concept. Over the last few decades the Brundtland Commission Report and the United Nations have emphasized the importance of sustainability and defined key concepts. Understanding and seeking sustainability is not only a must but also a challenge for today's engineers. Incorporating sustainability into design helps students build their engineering judgement beyond the short-term, technical issues that they tend to focus on to better prepare them for the ambiguities of professional practice. Consequently, engineering curricula and faculty should prepare students with the required knowledge, skills, and behaviors to address it. However, this preparation is not a simple task due to the little guidance available to achieve it on a daily basis. On the one hand, some approaches around it may generate ambiguities and misconceptions arising during the engineering design process. On the other hand, a concise but narrow perception of sustainability framing it as an environmental issue or being able to maintain a business activity over time introduce bias to a proper engineering design process aimed at sustainable development. In fact, it can be argued that there is not a single definition of sustainability suited to all engineering disciplines. However, the main elements of a sustainability model are general enough to be applied to most engineering disciplines.
In this paper, sustainability is addressed as a software engineering design concern that involves sustainability principles and multiple dimensions at different moments in time. The presented experience is aimed as a guide for teaching and assessing sustainability during a software engineering capstone design. It is based on the Karlskrona Manifesto for sustainability design, involving societal, individual, environmental, economic, technical dimensions, and considering, short, medium and long-term effects of engineering solutions. A sustainability matrix was used as a tool for analyzing and comparing different software systems designs. Based on the conducted experience, undergraduate students faced a challenge for identifying the impacts of software systems beyond a short-term time window, while graduate students were better able to identify potential impacts beyond first-order – short term time horizons. Learned lessons are shared for the sake of repeatability.

Tags: sustainability, Karlskrona Manifesto, Software Engineering, capstone design.

Introduction and paper goal

At 1987, The Brundtland Commission Report [1] defined sustainable development as meeting the needs of the present without compromising the ability of future generations to meet their own needs. The 1992 United Nations Conference on Environment and Development relate sustained development with simultaneous environmental protection, plus economic and social development [2]. These three factors were recognized as the interdependent and mutually reinforcing pillars for achieving sustainable development by the United Nations World Summit on Sustainable Development in 2002 [3]. More than thirty years later, a balance among the above-mentioned factors is widely adopted and promoted as a way to achieve sustainable management and development, as they are a foundation of the United Nations 17 Sustainable Development Goals of the 2030 Agenda for Sustainable Development [4]. The United Nations 17 Sustainable Development Goals aim to transform our world and to improve people's lives and prosperity on a healthy planet. It comprises 169 broad and interdependent goals. There is not an explicit mention of software in those goals. Information and communications technology and technological development are addressed by few of them, mainly targeting developing countries. Thus a software engineer might conclude that his or her responsibility regarding sustainability is narrow on a daily basis, or it is beyond software related activities. However, in practice, software plays a central role in sustainability, because of the extent to which software systems impact all aspects of our lives. In the context of software engineering, sustainable development can be understood as a way of how a software product is developed, so that negative and positive impacts on sustainability are captured, documented, and optimized over the whole product's life cycle [5]. Sustainable software was defined in [6] as software whose direct and indirect negative impacts on economy, society, human beings, and the environment resulting from the development, deployment, and use of the software is minimal or has a positive effect on sustainable development. In fact, the Brundtland Commission Report, the multiple United Nations initiatives, and several available works on literature do a good job of emphasizing the importance of sustainability and defining many key concepts. However, in practice, there is a gap between understanding sustainability and achieving it based solely in a conceptual ideal. This gap can be tackled during the engineering design process [7] since it is an iterative, creative, decision-making process for devising a system, component, or process fulfilling desired needs and specifications within a set of constraints [8]. Thus, the engineering design process requires involving sustainability principles during decision-making in order to obtain a sustainable engineering solution.

The goal of this paper is to present an experience aimed to incorporate the three main elements of a sustainability model – the Karlskrona Manifesto – during the engineering design process: sustainability principles, sustainability dimensions and order effects [9]. The Karlskrona Manifesto allows practitioners and researchers to understand the major role that software engineering plays on sustainability. It arose as the outcome of a cross-disciplinary initiative to create a common ground and develop a focal point of reference for the global community of research and practice allowing a deeper thinking on software engineering and sustainability. It has proven its usefulness for effectively communicating key issues, goals, values, and principles of sustainable design. Moreover, it has been adopted by researchers addressing a wide variety of subjects in software engineering (e.g. software requirements, software engineering curricula, systems thinking, design thinking, engineering ethics, and multidisciplinary engineering, among others) [10].

Background

Sustainable software approaches and models

A systematic mapping study on sustainability in software engineering is presented in [11]. Among the conclusions, authors highlight that in spite of the increasing attention sustainability has received in software engineering during the last years, it is still not clearly defined nor understood. In practice, software sustainability issues arise due to main factors: firstly, the time-to-market pressure during software development projects on which sustainability is an afterthought, and secondly, the software engineers' lack of education and skills for applying sustainability-improvement techniques [12].

Three main approaches for classifying software sustainability works are as follows [5]: (i) considering sustainability as a part of software quality [13], (ii) using quality attributes and metrics supporting sustainability [14], and (iii) adopting a global vision of sustainability [9]. From our perspective, the last approach offers advantages over the two formers. Such a claim can be supported by using a theoretical framework on sustainability. In fact, the Karlskrona Manifesto can be used to create a global view of sustainability. In this paper, three aspects of the Manifesto are highlighted for the sake of clarity: sustainability principles, sustainability dimensions, and order effects [9].

Sustainability principles

A sustainability principle is a perspective from which sustainability is assumed and adopted. The Karlskrona Manifesto entails nine principles and a pragmatic perception about sustainability.

- Sustainability is a challenge to be addressed, not a problem to be solved.
- Sustainability is a concern independent of the purpose of the system.
- Sustainability applies to both a system and its wider contexts.
- Sustainability requires actions on multiple levels.
- System visibility is a necessary precondition and enabler for sustainability design.
- Sustainability is systemic.
- Sustainability has multiple dimensions.
- Sustainability requires long-term thinking.
- Sustainability transcends multiple disciplines.
- It is possible to meet the needs of future generations without sacrificing the prosperity of the current generation.

Sustainability dimensions

Dimensions allow understanding the nature of sustainability in any given situation. These are summarized as follows.

- Environmental: focused on the long term effects of human activities on natural systems. It includes the use and stewardship of natural resources, such as energy consumption, waste production, and the impact on climate change, among others.

- Economic: focused on assets, capital, and added value. It includes wealth creation, prosperity, profitability, capital investment, and return on investment, among others.
- Societal: focused on the relationships between individuals and groups. It covers social equity, justice, employment, democracy, as well as the structures of mutual trust and communication in a social system and the balance between conflicting interests, among others.
- Individual: refers to the wellness of human beings. It includes individual freedom, human dignity, individuals' ability to thrive and exercise their rights, among others.
- Technical: refers to longevity of information, systems, and infrastructure and their adequate resilience and evolution with changing surrounding conditions. It includes maintenance, obsolescence, data integrity, and system transitions, among others.

Impacts of software systems
For evaluating the sustainability of a software system, three orders of effects need to be considered.

- Direct or first-order: are the immediate opportunities and effects created by the physical existence of software technology and the processes involved in its design and production.
- Indirect or second-order: are the opportunities and effects arising from the application and usage of software.
- Systemic or third-order: are the effects and opportunities that are caused by large numbers of people using software over time.

As we said before, it is possible to identify three main approaches for classifying software sustainability. The nature of approaches relating sustainability to quality models or to quality attributes and metrics, includes technical, environmental, economic, and –partially– societal dimensions. Nevertheless, by definition, these are focused on the technical dimension and in direct and indirect order effects related to the software development process, software product release and technical maintenance. In this way, the societal dimension is oriented to the software development team. Consequently, as a global vision of sustainability approach, the Karlskrona Manifesto provides a more holistic and comprehensive theoretical framework for addressing, understanding and discussing sustainability.

Sustainability analysis matrix

Although the Karlskrona Manifesto is focused on principles and values of sustainability, instead of current techniques, specific models, and suggested approaches, it can be used as the foundation of specific techniques. An adaptation of the Attribute-Driven Design method [15] was proposed in [16], as a research product of a master's in software engineering project at the Universidad de San Buenaventura – Cali. The adapted method is termed as Sustainable Attribute-Driven Design and includes sustainability drivers as one of the inputs with higher priority, aiming to achieve a sustainable architectural design. In the adapted method, an analysis of sustainability is documented based on a multidimensional impact matrix of the architectural components considered during the design phase by stakeholders. In this way, the adaptation introduced by the Sustainable Attribute-Driven Design method consists of analyzing and documenting sustainability impacts and opportunities of each architectural component regarding the five dimensions of sustainability against the three order effects. The multidimensional impact matrix is illustrated in Figure 1. In it, system architects and stakeholders can document impact and opportunities introduced by a specific design element regarding each dimension and considering direct, indirect and systemic effects. Impacts are associated with negative effects (i.e. threats, drawbacks, risks, waste, technical debt, among others), while opportunities are associated with positive leverages points (i.e. advantages, improvements, costs reductions, profits, among others) in terms of sustainability. Sample text of each dimension and order effect is shown in the examples from students work in the conducted experience section. A decision-making process on which designers choose among possible alternatives is based on such documented analysis. The contribution of the multidimensional matrix relies on providing to the software architects a tool to conduct a very complex, challenging and abstract task: an engineering design process promoting sustainability.

Teaching sustainability in the software engineering context

On the one hand, although international standards and curriculum guidelines such as the Software Engineering Body of Knowledge V 3.0 [17] and the SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [18] consider knowledge areas related to sustainability, there is not an explicit relationship between sustainability principles or concepts and such knowledge areas. Something similar happens with the Software Engineering Code of Ethics [19]. In fact, there is not wide guidance or a well-adopted approach on how to incorporate sustainability into a software engineering curriculum [10].

SUSTAINABILITY IMPACT MATRIX OF <<DESIGN ELEMENT>>						
<<DESIGN ELEMENT>>	First Order (Development and Design)		Second Order (Release and Use)		Third Order (Maintenance and Extended Use)	
Dimension	Impacts	Opportunities	Impacts	Opportunities	Impacts	Opportunities
Environmental	✗	✓	✗	✓	✗	✓
Societal						
Technical						
Economic						
Individual						

Figure 1. Multidimensional impact matrix for analyzing design sustainability.

Consequently, there is not an agreement on which subjects and material should be addressed in a sustainability course in the context of software engineering. Moreover, there is a lack of or reusable learning objects for this effort. Some approaches for integrating sustainability into the educational path can be described as follows [10]: (i) developing courses covering selected sustainability topics, (ii) developing modules or projects addressing sustainability issues to be plugged into existing courses and (iii) transforming existing courses aiming a sustainability awareness. However, it still not clear which one or combination of these is most effective. On the other hand, there are clear promoters for preparing software engineering students in sustainability topics at knowledge, comprehension and application levels. The definition of the engineering design process and the student outcome number four provided by the ABET's EAC [8] are among them.

Assessing sustainability in capstone design projects

Designing for sustainability is not an alternative to traditional engineering design, but a more holistic design paradigm [7]. Thus, the ultimate goal of engineering education embracing sustainable principles is to train engineers to incorporate sustainability considerations into their professional practices. Among the many contributions for assessing sustainability, we highlight the proposals of [7] and [20]. Authors of [7] developed an analytical sustainable design rubric to aid quantifying students' abilities to incorporate sustainability into capstone design projects in the civil and environmental engineering context. It aimed to capture not only the extent to which students engage in sustainable design but also the influence of project sponsors and/or course instructors on sustainable design expectations. In [20] the above-mentioned rubric is reviewed and extended in order to broaden its applicability to engineering design projects outside of civil and environmental engineering. Nevertheless, the original and the reviewed rubrics only partially cover the five sustainability dimensions discussed in the previous section, focusing on environmental, economic and societal dimensions. Moreover, indirect and systemic effects are not explicitly addressed in the rubrics.

Conducted experience

Sustainability posture

The conducted experience reflects a posture around both sustainability and the engineering design fostering it. It is defined as follows.

- There is not a unique or general set of features allowing to declare as sustainable a software product in all scenarios and circumstances. Consequently, during an iterative decision-making process, the goal is analyzing, comparing and choosing by stakeholders,

among the different alternatives based on the inherent sustainability trade-offs associated with each engineering solution
- A software product incorporating sustainability concerns is the result of a sustainable development process on which the engineering design decision-making explicitly involves, documents, and analyzes sustainability principles, dimensions and short, mid and long term order effects of the software system where the software product is used.
- Seeking sustainability is an ongoing and continuous effort which requires iterative analysis at different moments in time, incorporating constraints imposed by a changing environment.

Experience development and designed rubric

The conducted experience was developed at the Universidad de San Buenaventura – Cali, in bachelor and master programs, in their software engineering projects.
In the bachelor program, the instruction around sustainability took place in the final semester during the major design experience. The theory of the Karlskrona manifesto was presented to students along with examples to help them understand its application to real projects. It was led by the course coordinator, in periodic meetings, in order the students would incorporate their sustainability analysis in the final technical report and present it as part of their projects defense. In that scenario, students were supported by their project advisor and were responsible to explain sustainability trade-offs of the proposed software solution to stakeholders.
In the master program, the instruction took place in a first-year mandatory course, devoted to ethics and sustainability. The course faculty presented several models and approaches for analyzing sustainability in software engineering, as well as related subjects to the lack of it such as technical debt [21]. Special attention is devoted to the Karlskrona Manifesto approach and to the multidimensional matrix, as a tool for documenting and analyzing sustainability trade-offs. Students apply the learned concepts to their first version of the research-project proposal solution during the course. The exercise is repeated, as their move forward in project development, and updated analysis are presented by students in follow-up progress meetings in front of the program faculty members committee. A final sustainability analysis is presented at the research-project defense. In practice, graduate students had more feedback than undergraduate students. Feedback was provided to all students during follow-up meetings. A rubric was developed to this end. It is shown in Figure 2.

Study limitations and motivation

This practice has been applied for two years in the bachelor program and for one year in the master program. So far, we have focused on designing and developing the respective sustainability undergraduate module and the graduate course. We have not yet conducted surveys focused on student’s perceptions of sustainability issues, neither pre-course nor post-course tests.

Performance Indicators	Satisfactory	Developing	Unsatisfactory
Sustainability principles	All impacts and opportunities properly reflects an understanding and a perspective integrating multiple principles.	Most of impacts or opportunities are properly formulated and can be associated to at least one principle.	Formulation of several impacts or opportunities is ambiguous, or do not correspond to a real impact neither opportunity.
Order effects	Presented impacts and opportunities properly cover direct, indirect and systemic effects.	Most of presented impacts and opportunities correspond to direct order effects.	Most of impacts and opportunities do not correspond to the associated order.
Sustainability dimensions	All presented impacts and opportunities correspond properly to associated dimensions, and all dimensions include impacts or opportunities.	There are only impacts or opportunities properly related for a subset of dimensions.	Most of presented impacts or opportunities do not correspond properly to the associated dimension.

Figure 2. Rubric for assessing sustainability analysis matrix.

The student outcome of the designed and developed educational experiences is stated as follows: a capability to apply engineering judgment regarding sustainability issues in the context of software engineering. Regarding the undergraduate program, preparing the students for performing sustainability analysis from a software engineering perspective contributes to the educational objectives fulfillment. Taking into account the professional profile of master students and their active role on the local software industry, engaging them in explicitly incorporating sustainability aspects in their professional exercise is a matter of social and ethical responsibility by the faculty and the program, and a way to serve program constituents.

Student sample work and observed results

Two examples of matrices for a graduate and undergraduate student, respectively, are shown for the sake of illustration.

The matrix shown in Figure 3 corresponds to a master project introducing the software architecture for a Health Catalog Repository, which allows the storage, management, and custody of Electronic Health Records based on a regulated model of Personal Health Records, as a cloud service [22, 23]. The obtained sustainability analysis matrix is associated with strong student performance on the educational experience outcome. Impacts and opportunities were properly documented and supported, and properly located along the three orders. Moreover, impacts and opportunities correspond to the developed project specificities, properly viewed from each sustainability dimension.

SUSTAINABILITY IMPACT MATRIX OF HEALTH RECORD BANK IN A HEALTH CATALOGUE REPOSITORY						
Health Record Bank	First Order (Development and Design)		Second Order (Release and Use)		Third Order (Maintenance and Extended Use)	
Dimension	Impacts	Opportunities	Impacts	Opportunities	Impacts	Opportunities
Environmental		Avoids physical waste generation		Allows cost-effective use of natural resources		Allows a low carbon footprint by green cloud computing technology
Societal			Requires regulatory laws for use Requires service levels agreement on information handling	Allows accessible health information		Supports a proper health care attention by a secured and controlled information exchange
Technical	Requires cloud development capabilities Requires a user centered design approach	Allows distributed storage with reusable components and flexible services by demand	Require secure storage mechanisms	Allows timely access to consolidated personal health information Avoids duplicated health information		Avoids on site platform obsolence Allows durable health documents storage
Economic	Implies an expensive design process	Allows low investment on infrastructure		Allows the generation of new incomes by incorporated services Avoids the investment on storage licenses	Requires sustained financial support	Allows new business models
Individual		Allows privacy, confidentiality and health care information protection	Requires of basic technological skills for a proper use Requires a responsible use for keeping information security	Provides a high quality of experience for health care services Allows health information ubiquity		

Figure 3. Sustainability matrix for a software architecture developed during a master project.

Beyond this particular example, we also observed that some graduate students fail to properly address the specificity of the software project on hands while they rely on common places for some dimensions (e.g. user experience in the individual dimension, and identify the opportunity of using green-cloud infrastructure during software development in the environmental dimension). It was also clear that the periodic presentations of conducted sustainability analysis allow students to improve it and enhance it, prior to the final defense, besides that their long experience in the software industry and software projects gave them a broader perspective. Additionally, a more confident professional communication allowed them to be more effective in sharing their perspective with stakeholders. The matrix shown in Figure 4 corresponds to a bachelor project focused on a multimodal image retrieval model in the context of a smart-city safety. It tackles a face detection, comparison, indexing and recovery problem in the context of a very large image database [24]. While the project was technically successful, and properly fulfilled its objectives, the obtained sustainability analysis matrix is an example of weak student performance. The formulation of some impacts and opportunities were ambiguous and were not properly associated with the respective dimension. There are only impacts for a subset of dimensions. Long-term impacts and opportunities are only present for a single dimension. Beyond this example, we also observed that undergraduate students found a challenge for identifying impacts and opportunities, in most of the dimensions, beyond direct order effects. An unclear distinction between the societal and the individual dimensions was also commonly found in sustainability analysis. Moreover, undergraduate students tend to avoid identifying design impacts. We believe they were afraid that by doing so, were exposing in somehow design weaknesses, and it may hamper their projects assessment.

SUSTAINABILITY IMPACT MATRIX OF A FACE RECOVERY SYSTEM FOR A SMART CITY						
Face comparison and recovery	First Order (Development and Design)		Second Order (Release and Use)		Third Order (Maintenance and Extended Use)	
Dimension	Impacts	Opportunities	Impacts	Opportunities	Impacts	Opportunities
Environmental	There is not environmental impact			Increases the well being in the city		
Societal			Requires laws for people right protection	Increases safety for citizens		
Technical	There is not technical impact	Allows durable and elastic data storage Software was developed applying quality standards		Technological infrastructure properly support software use		
Economic	There is not economic impact					Safety will improve businesses opportunities Will allow business ideas and models
Individual		Considers user centered design for improving user experience	Improper use may affect individuals rights	Requires a low learning curve		

Figure 4. Sustainability matrix for a software system developed during an undergraduate project.

Additionally, undergraduate students were more susceptible to be influenced by stakeholders and even advisor perspectives, instead of presenting impacts and opportunities totally analyzed on their own. As a common factor, graduate and undergraduate students felt more confident identifying impacts and opportunities related to the technical dimension.

Conclusions

Sustainability reasoning about software products imposes particular challenges, as software is mainly concentrated in logical and abstract entities instead of concrete or physical artifacts. Most of the times, the origin of sustainability issues in software projects arises due to giving exclusive attention to the economic and the technical dimensions during the development phase, while only the direct impacts of the software systems are considered. Such a scenario on which sustainability is an afterthought can also be encouraged by the necessity of having working software and time-to-market pressures. Against this common practice, the research community in software engineering is still pursuing concise and widely accepted guidance for the multiple aspects of sustainability in our professional exercise. Such guidance should be supported by specific tools to be applied during the engineering design process. In this paper, the contributions are as follows. The multidimensional impact matrix is presented and discussed aiming to build in students engineering judgment around sustainability issues. It also introduces a rubric for assessing how properly students understand and apply in a concrete engineering design situation the three main elements of a sustainability model: sustainability dimensions, sustainability principles, and order effects. In this way, the proposed rubric is both specific and general enough to be used not only in a sustainability analysis based on the model introduced by the Karlskrona Manifesto, but also with other sustainability models. Challenges faced by students are discussed, giving some insights to faculty seeking to conduct a similar pedagogical experience. Currently we are working on adding more examples to clarify the difference between individual and societal dimensions, and also to better illustrate possible impacts and opportunities beyond first-order effects.

References

[1] G. Brundtland et al., “Our Common Future: Report of the World Commission on Environment & Development”, Oxford University Press, 1987.

[2] United Nations, “UN - Conference on Environment and Development”, [Online] http://www.un.org/geninfo/bp/enviro.html, Accessed January 31, 2019.

[3] United Nations, “UN - World Summit on Sustainable Development”, [Online] https://sustainabledevelopment.un.org/milesstones/wssd, Accessed January 31, 2019.

[4] United Nations, “UN - Sustainable Development Goals”, [Online] hhttps://www.un.org/sustainabledevelopment/sustainable-development-goals/, Accessed January 31, 2019.

[5] R. Amri and N. B. B. Saoud, "Towards a Generic Sustainable Software Model," Fourth International Conference on Advances in Computing and Communications, pp. 231-234, 2014.

[6] M. Dick, S. Naumann, and N. Kuhn, “A model and selected instances of green and sustainable software” What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience, IFIP Advances in Information and Communication Technology Vol. 328, pp. 248-259, 2010.

[7] M. Watson et al., “Development and Application of a Sustainable Design Rubric to Evaluate Student Abilities to Incorporate Sustainability into Capstone Design Projects”, 120th ASEE Annual Conference & Exposition, 2013.

[8] ABET, “EAC - Criteria for Accrediting Engineering Programs, 2019 – 2020”, [Online] https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/, Accessed January 31, 2019.

[9] C. Becker et al., “Sustainability Design and Software: The Karlskrona Manifesto”, IEEE/ACM 37th IEEE International Conference on Software Engineering - ICSE, pp. 467–476, 2015.

[10] B. Penzenstadler, et al., "Everything is INTERRELATED: teaching software engineering for sustainability", 40th International Conference on Software Engineering: Software Engineering Education and Training - ICSE-SEET, ACM, pp 153-162, 2018.

[11] N. Wolfram, P. Lago, and F. Osborne, "Sustainability in software engineering", Sustainable Internet and ICT for Sustainability (SustainIT) 2017.

[12] Z. Durdik, B. Klatt, H. Koziolek, K. Krogmann, J. Stammel, and R. Weiss, “Sustainability guidelines for long-living software systems,” in 28th Intl. Conf. on Software Maintenance, pp. 517–526, 2012.

[13] C. Calero, M. A. Moraga, and M. F. Bertoa, “Towards a software product sustainability model,” University of Castilla-La Mancha & University of Malaga - 2013, [Online] https://arxiv.org/abs/1309.1640 , Accessed January 31, 2019.

[14] F. Albertao et al., “Measuring the sustainability performance of software projects”, Proc. the IEEE 7th International Conference on e-Business Engineering - ICEBE, pp. 369-373, 2010.

[15] R. Wojcik, “Attribute-Driven Design Method – ADD”, [Online] http://www.sei.cmu.edu/architecture/tools/define/add.cfm, Accessed January 31, 2019.

[16] Villa, L., et al., “Towards a Sustainable Architectural Design by an Adaptation of the Architectural Driven Design Method”, pp. 71-86, ICCSA, 2016.

[17] IEEE Computer Society, “Software Engineering Body of Knowledge SWEBOK V3 Guide”,[Online] https://www.computer.org/education/bodies-of-knowledge/software-engineering/topics , Accessed January 31, 2019.

[18] ACM, “Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering", [Online] https://www.acm.org/education/curricula-recommendations, Accessed January 31, 2019.

[19] IEEE Computer Society and ACM, “Software Engineering Code of Ethics”, [Online], https://www.computer.org/education/code-of-ethics, Accessed January 31, 2019.
[20] C. Cowan, E. Barrella, and M. Watson, “Validating Content of a Sustainable Design Rubric Using Established Frameworks”, 124th ASEE Annual Conference & Exposition, 2017.
[21] R. Verdecchia, I. Malavolta and P. Lago, "Architectural Technical Debt Identification: The Research Landscape", IEEE/ACM International Conference on Technical Debt -TechDebt, pp. 11-20, 2018.
[22] Villa, L., et al.: Electronic health record as an eHaaS. In: 2015 10th Computing Colombian Conference, 10CCC, 2015.
[23] Black, A.S., Sahama, T.: eHealth-as-a-Service (eHaaS): the industrialization of health informatics, a practical approach. In: 2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 555–559, 2014.
[24] J. Cao, B. Wang, and D. Brown, "Similarity based leaf image retrieval using multiscale R-angle description", Information Sciences, Volume 374, pp. 51-64, 2016.