Advances in Intelligent Systems and Computing 865

Jezreel Mejia
Mirna Muñoz
Álvaro Rocha
Adriana Peña
Marco Pérez-Cisneros *Editors*

Trends and Applications in Software Engineering

Proceedings of the 7th International Conference on Software Process Improvement (CIMPS 2018)

Springer

Advances in Intelligent Systems and Computing

Volume 865

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Jezreel Mejia · Mirna Muñoz
Álvaro Rocha · Adriana Peña
Marco Pérez-Cisneros
Editors

Trends and Applications in Software Engineering

Proceedings of the 7th International Conference on Software Process Improvement (CIMPS 2018)

Springer

Editors
Jezreel Mejia
Unidad Zacatecas
Centro de Investigación en Matemáticas A. C.
Zacatecas, Zacatecas, Mexico

Mirna Muñoz
Unidad Zacatecas
Centro de Investigación en Matemáticas A. C.
Zacatecas, Zacatecas, Mexico

Álvaro Rocha
Departamento de Engenharia Informática
Universidade de Coimbra
Coimbra, Portugal

Adriana Peña
Departamento de Ciencias Computacionales
Universidad de Guadalajara
Guadalajara, Jalisco, Mexico

Marco Pérez-Cisneros
Centro Universitario de Ciencias Exactas
Universidad de Guadalajara
Guadalajara, Jalisco, Mexico

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-030-01170-3 ISBN 978-3-030-01171-0 (eBook)
https://doi.org/10.1007/978-3-030-01171-0

Library of Congress Control Number: 2018955572

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Introduction

This book contains a selection of papers accepted for presentation and discussion at the 2018 International Conference on Software Process Improvement (CIMPS'18). This conference had the support of the CIMAT A.C. (Mathematics Research Center/Centro de Investigación en Matemáticas), CUCEI (Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara, Jalisco, México), AISTI (Iberian Association for Information Systems and Technologies/Associação Ibérica de Sistemas e Tecnologas de Informação), ReCIBE (Revista electrónica de Computación, Informática, Biomédica y Electrónica), ROPRIN (Red de Optimización de Procesos Industriales), ORACLE, and IBM. It took place at CUCEI, Guadalajara, Jalisco, México, on October 17–19, 2018.

The International Conference on Software Process Improvement (CIMPS) is a global forum for researchers and practitioners that present and discuss the most recent innovations, trends, results, experiences, and concerns in the several perspectives of software engineering with clear relationship but not limited to software processes, security in information and communication technology, and big data field. One of its main aims is to strengthen the drive toward a holistic symbiosis among academy, society, industry, government, and business community, promoting the creation of networks by disseminating the results of recent research in order to align their needs. CIMPS'18 was built on the successes of CIMPS'12, CIMPS'13, CIMPS'14, which took place on Zacatecas, Zac; CIMPS'15 which took place on Mazatlán, Sinaloa; CIMPS'16 which took place on Aguascalientes, Aguascalientes, México; and the last edition CIMPS'17 which took place again on Zacatecas, Zac, México.

The program committee of CIMPS'18 was composed of a multidisciplinary group of experts and those who are intimately concerned with software engineering and information systems and technologies. They have had the responsibility for evaluating, in a 'blind review' process, the papers received for each of the main themes proposed for the conference: organizational models, standards and methodologies; knowledge management; software systems, applications and tools; information and communication technologies and processes in non-software

domains (mining, automotive, aerospace, business, health care, manufacturing, etc.) with a demonstrated relationship to software engineering challenges.

CIMPS'18 received contributions from several countries around the world. The papers accepted for presentation and discussion at the conference are published by Springer (this book), and extended versions of best selected papers will be published in relevant journals, including SCI/SSCI and Scopus-indexed journals.

We acknowledge all those who contributed to the staging of CIMPS'18 (authors, committees, and sponsors); their involvement and support are very much appreciated.

October 2018

Jezreel Mejia
Mirna Muñoz
Álvaro Rocha
Adriana Peña
Marco Pérez-Cisneros

Organization

Conference

General Chairs

Jezreel Mejía	Mathematics Research Center, Research Unit Zacatecas, Mexico
Mirna Muñoz	Mathematics Research Center, Research Unit Zacatecas, Mexico

The general chairs and co-chair are researchers in Computer Science at the Research Center in Mathematics, Zacatecas, México. Their research field is software engineering, which focuses on process improvement, multi-model environment, project management, acquisition and outsourcing process, solicitation and supplier agreement development, agile methodologies, metrics, validation and verification, and information technology security. They have published several technical papers on acquisition process improvement, project management, TSPi, CMMI, multi-model environment. They have been members of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish.

General Support

CIMPS General Support represents centers, organizations, or networks. These members collaborate with different European, Latin American, and North American organizations. The following people have been members of the CIMPS conference since its foundation for the last 7 years.

Cuauhtémoc Lemus Olalde	Head of CIMAT Unit Zacatecas, Mexico
Gonzalo Cuevas Agustín	Polytechnical University of Madrid, Spain
Jose A. Calvo-Manzano Villalón	Polytechnical University of Madrid, Spain

Tomas San Feliu Gilabert	Polytechnical University of Madrid, Spain
Álvaro Rocha	Universidade de Coimbra, Portugal

Local Committee

CIMPS established a local committee from the University of Guadalajara, Jalisco, Mexico, and the Mathematics Research Center; Research Unit Zacatecas, Mexico. The list below comprises the local committee members.

Adriana Peña Pérez Negrón (Local Chair)	CUCEI
Graciela Lara López (Local Co-chair)	CUCEI
María Elena Romero Gastelu (Public Relations)	CUCEI
María Patricia Ventura Núñez (Local Staff)	CUCEI
Carlos Alberto López Franc (Local Staff)	CUCEI
Elsa Estrada Guzmán (Local Staff)	CUCEI
Marco Antonio Pérez Cisneros (Academic Support)	CUCEI
Himer Ávila George (Local Staff)	CUVALLES
José Antonio Cervantes Álvarez (Local Staff)	CUVALLES
Sonia López Ruiz (Local Staff)	CUVALLES
Miguel Ángel De la Torre Gómora (Local Staff)	CUVALLES
Isaac Rodríguez Maldonado (Support)	CIMAT Unit Zacatecas, Mexico
Saúl Ibarra Luevano (Support)	CIMAT Unit Zacatecas, Mexico
Ana Patricia Montoya Méndez (Support)	CIMAT Unit Zacatecas, Mexico
Héctor Octavio Girón Bobadilla (Support)	CIMAT Unit Zacatecas, Mexico

Scientific Program Committee

CIMPS established an international committee of selected well-known experts in software engineering who are willing to be mentioned in the program and to review a set of papers each year. The list below comprises the scientific program committee members.

Adriana Peña Pérez-Negrón	University of Guadalajara, Mexico
Alejandro Calderón	University of Cádiz, Spain
Alejandro Rodríguez González	Polytechnical University of Madrid, Spain
Alejandra García Hernández	Autonomous University of Zacatecas, Mexico
Álvaro Rocha	Universidade de Coimbra, Portugal
Ángel M. García Pedrero	Polytechnical University of Madrid, Spain
Antoni Lluis Mesquida Calafat	University of Islas Baleares, Spain
Antonio de Amescua Seco	University Carlos III of Madrid, Spain
Benjamín Ojeda Magaña	University of Guadalajara, Mexico
Carla Pacheco	Technological University of Mixteca, Oaxaca, Mexico
Carlos Abraham Carballo Monsivais	CIMAT Unit Zacatecas, Mexico
Carlos Lara Álvarez	CIMAT Unit Zacatecas, Mexico
Edgar Oswaldo Díaz	INEGI, Mexico
Edrisi Muñoz Mata	CIMAT Unit Zacatecas, Mexico
Elisabet Cápon	Swiss Federal Institute of Technology, Zürich (ETHZ), Switzerland
Eleazar Aguirre Anaya	National Polytechnical Institute, Mexico
Fernando Moreira	University of Portucalense, Portugal
Gabriel A. García Mireles	University of Sonora, Mexico
Giner Alor Hernández	Technological University of Orizaba, Mexico
Gloria P. Gasca Hurtado	University of Medellin, Colombia
Graciela Lara López	University of Guadalajara, Mexico
Gonzalo Cuevas Agustín	Polytechnical University of Madrid, Spain
Gonzalo Luzardo	Higher Polytechnic School of Litoral, Ecuador
Gustavo Illescas	National University of Central Buenos Aires Province, Argentina
Himer Ávila George	University of Guadalajara, Mexico
Hugo Arnoldo Mitre	CIMAT Unit Zacatecas, Mexico
Hugo O. Alejandrez-Sánchez	National Center for Research and Technological Development, CENIDET, Mexico
Iván García Pacheco	Technological University of Mixteca, Oaxaca, Mexico
Jezreel Mejía Miranda	CIMAT Unit Zacatecas, Mexico
José Alberto Benítez Andrades	University of Lion, Spain
Jose A. Calvo-Manzano Villalón	Polytechnical University of Madrid, Spain
José Antonio Cervantes Álvarez	University of Guadalajara, Mexico
José Luis Sánchez Cervantes	Technological University of Orizaba, Mexico

Juan Manuel Toloza	National University of Central Buenos Aires Province, Argentina
Leopoldo Gómez Barba	University of Guadalajara, Mexico
Lohana Lema Moreta	University of the Holy Spirit, EC
Luis Casillas	University of Guadalajara, CUCEI, Mexico
Luis J. Domínguez Pérez	CIMAT Unit Zacatecas, Mexico
Magdalena Arcilla Cobián	National Distance Education University, Spain
Manuel Pérez Cota	University of Vigo, Spain
María de León Sigg	Autonomous University of Zacatecas, Mexico
Mary Luz Sánchez-Gordón	Østfold University College, Norway
Miguel Ángel De la Torre Gómora	University of Guadalajara, CUCEI, Mexico
Miguel Hidalgo Reyes	Institute of Ecology, Mexico
Mirna Muñoz Mata	CIMAT Unit Zacatecas, Mexico
Omar S. Gómez	Higher Polytechnic School of Chimborazo, Ecuador
Perla Velasco-Elizondo	Autonomous University of Zacatecas, Mexico
Ramiro Goncalves	University Tras-os Montes, Portugal
Raúl Aguilar Vera	Autonomous University of Yucatán, Mexico
Ricardo Colomo Palacios	Østfold University College, Norway
Lisbeth Rodríguez Mazahua	Technological University of Orizaba, Mexico
Rafael Valencia-García	University of Murcia, Spain
Rory O'Connor	Dublin City University, Ireland
Santiago Matalonga	University of the West, Scotland
Sergio Galván Cruz	Autonomous University of Aguascalientes, Mexico
Sodel Vázquez Reyes	Autonomous University of Zacatecas, Mexico
Sonia López Ruiz	University of Guadalajara, Mexico
Tomas San Feliu Gilabert	Polytechnical University of Madrid, Spain
Ulises Juárez Martínez	Technological University of Orizaba, Mexico
Vianca Vega	Catholic University of North Chile, Chile
Víctor Flores	Catholic University of the North, Chile
Víctor Saquicela	University of Cuenca, Ecuador
Viviana Y. Rosales Morales	Technological University of Orizaba, Mexico
Yadira Quiñonez	Autonomous University of Sinaloa, Mexico
Yilmaz Murat	Çankaya University, Turkey

Contents

Knowledge Management

Software Systems, Applications and Tools

Organizational Models, Standards and Methodologies

Extending ISO/IEC 29110 with Sustainability Tasks

Gabriel Alberto García-Mireles[1(✉)] and Miguel Ehécatl Morales-Trujillo[2]

[1] Departamento de Matemáticas, Universidad de Sonora, Hermosillo, Sonora, Mexico
mireles@mat.uson.mx
[2] Computer Science and Software Engineering Department, University of Canterbury, Christchurch, New Zealand
miguel.morales@canterbury.ac.nz

Abstract. Sustainability is an aspect to be considered by software development organizations because it is believed that software systems are means to support sustainable development. Currently, few processes, methods and tools exist to practice sustainability design within software engineering, and they are scattered across different application domains and life cycle stages. With the purpose of providing a set of practices in order to address sustainability goals during software development, this paper proposes a set of sustainability tasks to be implemented by very small software organizations. The tasks were derived from literature containing sustainability practices. The latter were organized as tasks and integrated into the ISO/IEC 29110 Basic Profile processes: project management and software implementation. The proposed sustainability tasks were initially validated against two previously reported studies on environmentally sustainable software development.

Keywords: Environmental sustainability · Green software
Software process improvement · Sustainability tasks · ISO/IEC 29110

1 Introduction

Within software engineering (SE) community, sustainability is a topic of recent interest as literature shows work in diverse domains and throughout software development life cycle stages [1]. In order to address sustainability in software projects, researchers have suggested that software developers should consider its dimensions (economic, social, environment, at least) as well as the direct and indirect impacts on sustainable development [2, 3].

Sustainable software is defined as follows: "software, whose impacts on economy, society, human being, and environment that result from development, deployment and usage of the software are minimal and/or which has a positive effect on sustainable development" [4]. Although the SE sustainability research community suggests focusing on all the sustainability dimensions [5], the majority of research work has been conducted in the environmental dimension [6]. Environmental issues can also be

J. Mejia et al. (Eds.): CIMPS 2018, AISC 865, pp. 3–13, 2019.
https://doi.org/10.1007/978-3-030-01171-0_1

treated as the green dimension of sustainable development [7]. In this dimension, the strategies to follow are minimizing the use of resources as well as reducing energy consumption.

Research in sustainability from a software engineering perspective is in its initial stage regarding both its definition and related software development practices. On the first issue, sustainability is treated as a quality goal [8], but it is difficult to identify its components in order to establish appropriate indicators and measures [9]. On the issue of sustainability practices, Chitchyan et al. [10] pointed out that software practitioners need methodologies to carry out sustainable design within SE. Similarly, Manotas et al. [11] mentioned that practitioners lack the necessary information and support infrastructure to develop green software. In addition, there are multiple factors that influence the extent to which the sustainability can be achieved in a software product [12]. Indeed, sustainability is a systemic aspect that has multiple dimensions and requires actions on multiple levels [5].

Considering the need of developing a systemic approach to address sustainability during a software development life cycle, this paper suggests a set of sustainability tasks, i.e., tasks aiming to achieve sustainability goals, based on sustainability practices discussed in surveys, literature reviews, and papers addressing sustainability practices with a software process focus. The sustainability tasks are presented as an extension of the ISO/IEC 29110 Basic Profile process activities [13]. A preliminary validation was carried out through an attempt to categorize practices from studies on green software development by means of our proposal.

The paper is structured as follows. Section 2 describes the relevant literature about sustainability and software processes. Section 3 briefly presents the methodology followed in this research. Section 4 describes our proposal focusing on sustainability tasks integrated into the ISO/IEC 29110 processes, while Sect. 5 presents an initial validation. Discussion of results is described in Sect. 6. Finally, conclusions and further work are addressed in Sect. 7.

2 Related Work

This section provides a sustainability in SE background and briefly presents ISO/IEC 29110 to support the proposal presented herein. In this paper, the term ‘practices’ refers to actions carried out by practitioners or those reported by case studies. ‘Activity’ and ‘task’ terms are used as defined in the ISO/IEC 29110 [13]. The former term refers to a set of tasks while the latter describes a recommended action intended to accomplish a process objective.

2.1 Background

SE and sustainability literature reviews reflect trends in methods and practices. In the period from 2006 to 2012, few papers addressed this topic and researchers founding little methodological guidance to support sustainability [14]. In a follow up study, researchers found sustainability studies in the areas of software process, software

design and software quality [1]. However, the authors concluded that there is little evidence of sustainability practices validation in industrial settings [1].

In another literature review about software process and environmental sustainability [15], the author reported few theoretical papers. In general, the selected papers focused on identifying software life cycle stages that should address sustainability aspects [16, 17] and on developing sustainability processes based on the ISO/IEC 12207 [18].

With regard to software life cycle proposals, Naumann et al. [16] described the GreenSoft Model whose goals are oriented towards achieving better sustainable development through software. This model considers both direct and indirect impacts on environmental, social, and economic dimensions. The product life cycle consists of development, usage, and end of life stages; however, no detailed structure of each of them is provided. Also, the model includes an example of a software process that fits the development stage. The description is carried out through: sustainability reviews, process assessment, a sustainability journal (containing a log of improvement effects) and a sustainability retrospective [16].

Taking into account the characteristics of the GreenSoft Model, Mahmoud and Ahmad [17] proposed a software life cycle model, which involves requirements and testing stages, for addressing green computing. In addition, each life cycle stage consists of actions and recommendations. On the other hand, Lami et al. [18] defined three processes based on the ISO/IEC 12207 [19]. They are: a sustainability management process, a sustainability engineering process and a sustainability qualification process. However, these processes are discussed at a high level of abstraction in terms of process purposes and outcomes.

A methodological support for addressing sustainability is scattered across software development life cycle stages and until now, there are few validated proposals in industrial settings [1, 20]. Indeed, sustainability in SE is still in an immature stage [6]. Thus, in order to organize practices addressing sustainability, we decided to present them as an extension to ISO/IEC 29110 [13]. The sustainability should be addressed by all software organizations and considered in all software projects [5]. Therefore, the proposal presented herein contributes to addressing sustainability within very small organizations. In addition, given the immature state of sustainability in SE, we believe that the ISO/IEC 29110 Basic Profile [13] is an appropriate framework to organize current sustainability practices.

2.2 ISO/IEC 29110 Basic Profile

The ISO/IEC 29110 Basic Profile [13] consists of two processes: project management and software implementation, which are derived from the ISO/IEC 12207 software life cycle process [19] taking into account very limited resources of software organizations composed of up to 25 persons. The expected benefits of implementing ISO/IEC 29110 are an increase of customer satisfaction and product quality together with a minimization of development costs.

The purpose of the project management process is to establish a plan to support software development activities in order to fulfill expected project goals. It is consists

of the following activities: planning, executing, assessment and control, and closure. The purpose of the software implementation process is to systematically develop a software product through the activities of software implementation initiation, software requirements analysis, software architectural and detailed design, software construction, software integration and tests, and product delivery. While the project management process consists of four activities and 26 tasks, the software implementation process comprises six activities and 41 tasks.

3 Approach to Define a Sustainability Extension

In order to develop this proposal, the following steps were carried out:

1. Identifying relevant papers addressing sustainability related practices. In order to provide a full view of the software process and to identify practices instead of techniques, the papers can be gathered from systematic literature reviews, surveys conducted in industrial settings as well as papers that address sustainability practices for more than one software life cycle stage. The selected papers are presented in Sect. 4.
2. Mapping identified sustainability practices to ISO/IEC 29110. A template that considers both the ISO/IEC 29110 [13] activities and tasks was developed to guide the mapping process. Each selected paper was reviewed and text fragments addressing a practice or recommendation were extracted. The template with process activities was filled in with the related text fragments. The mapping was carried out by the first author and verified by the second author. Based on the identified sustainability practices, *sustainability tasks* were proposed (Tables 1 and 2).
3. Defining a sustainability related role and products. A sustainability expert role is needed to support sustainability related practices. Their responsibilities are described in Sect. 4. The ISO/IEC 29110 [13] products were reviewed and an enhanced proposal is presented to include sustainability aspects (Table 3).
4. Carrying out an initial validation. The initial validation is focused on assessing to what extent the *sustainability tasks* are useful for categorizing sustainability practices used in case studies or experience-based suggestions. Two papers addressing sustainability practices during software development were selected. The first paper provides a framework of recommended practices for developing green software, mainly targeted for software design and construction stages [21]. The second paper describes a case study which addresses practices for eliciting sustainability requirements [22]. The mapping procedure was similar to the described in bullet 2, where text fragments extracted from these two papers were labeled with corresponding activities from ISO/IEC 29110. The second author verified the identification and classification of text fragments. Issues were resolved through a virtual peer debriefing.

Table 1. Sustainability tasks for project management process

Activity	Sustainability task	References
Project planning	1. Review business and software product goals considering sustainability 2. Determine sustainability scope in terms of dimensions, stakeholders, system boundaries, and time span 3. Identify and document sustainability risks 5. Define sustainability roles, responsibilities and specific tasks 6. Identify sustainability training needs 7. Review sustainability goals with customer 8. Establish a sustainability repository	[10–12, 17, 23, 24]
Project plan execution	9. Analyze change requests for sustainability goals 10. Review with development team sustainability goals' issues and mitigation actions 11. Review with customer and other stakeholders changes on sustainability goals	[13, 20]
Project assessment and control	12. Evaluate project progress considering sustainability goals 13. Establish corrective actions when sustainability risks emerge	[17, 24]
Project closure	14. Inform customer on the extent sustainability goals were addressed in software product	[13, 24]

Table 2. Sustainability tasks for software implementation process

Activity	Sustainability task	Reference
Software implementation initiation	1. Review with work team sustainability goals in current project 2. Set up the implementation environment considering means to support sustainability	[10, 11, 13, 23]
Software requirements analysis	3. Identify and document sustainability requirements considering sustainability goals 4. Analyze sustainability requirements considering their feasibility, risk, and potential interdependencies among dimensions	[11, 17, 20, 23]

(*continued*)

Table 2. (*continued*)

Activity	Sustainability task	Reference
	5. Review that sustainability requirements are consistent with product description and are testable	
Software architectural and detailed design	6. Based on sustainability dimensions addressed, determine sustainability criteria to define the software architecture 7. Document software architecture based on sustainability criteria 8. Verify that software design meets sustainability criteria and requirements 9. Develop test cases for sustainability goals	[11, 17, 20, 23, 24]
Software construction	10. Construct software components considering sustainability criteria 11. Test software components considering sustainability requirements	[11, 17, 20]
Software integration and tests	12. Execute sustainability test cases 13. Determine to what extent the system achieves sustainability goals (e.g.: develop energy profiles)	[11, 17]
Product delivery	14. Document, for maintenance activities, tools and procedures used to achieve sustainability goals	[17, 24]

Table 3. Sustainability items to be added to existing work products

Product	Sustainability related action
4. Maintenance documentation	Add a section to describe the environment (compilers configuration, static analysis tools, energy profiler tools, among others) for defining sustainability requirements, energy profiles and testing
6. Product operation guide	The product describes criteria for operational use and it could include sustainability scenarios and recommended operation modes that contribute to energy efficiency or other sustainability goals
7. Progress status record	Include a record of actual results about sustainability goals against planned goals
8. Project plan	Include sustainability objectives of the project. Deliverables and tasks with which sustainability goals are addressed should be defined. Resources to address sustainability goals as well as the required training should be described. Identify a sustainability related repository. Identify project risks considering sustainability in their both direct and indirect impacts as well as in the sustainability dimensions
11. Requirements specification	Add a section to describe sustainability requirements considering their impacts and dimensions. Include potential interaction among sustainability requirements and nonfunctional requirements

(*continued*)

Table 3. (*continued*)

Product	Sustainability related action
15. Software design	Include a description of the extent sustainability goals and requirements to guide software architecture definition. Within software components, describe sustainability aspects they support
16. Software user documentation	In the operational environment section, it should include a description of the way sustainability goals can be achieved with current software version. It could include sustainability risks, warnings and notes
17. Statement of work	It should include sustainability objectives of the project
18. Test cases and test procedures	Include in test cases and procedures the means to identify sustainability related issues. They could consider energy profiles

4 Sustainability Tasks Integrated into ISO/IEC 29110

The articles considered to identify sustainability practices can be classified into surveys conducted among practitioners [10, 11, 23], proposals focused on software processes [17, 24] and models derived from literature reviews [12, 20]. They are referenced in Tables 1 and 2 grouped by process activities from ISO/IEC 29110 [13].

In the surveys group, Groher and Weinreich [23] presented a qualitative interview study, targeted to software project team leaders, with the aim of understanding how sustainability is currently managed in software development projects. The authors identified influencing factors, problems and measures they took to improve sustainability. In another qualitative study, Manotas et al. [11] conducted a survey targeted to 464 practitioners, with the purpose of identifying current practices used by software engineers to address green SE. They provided a suggestion to develop suitable software. On the other hand, Chitchyan et al. [10] explored the perceptions and attitudes of requirement engineers towards sustainability. They identified concerns practitioners faced and mitigation strategies based on sustainability design principles.

In the software process related proposals, Dick et al. [24] proposed an agile extension to software development process to integrate sustainability practices both into software process and software product. The proposed practices are complemented by some tools and guidelines. Mahmoud and Ahmad [17] proposed sustainability practices that could be addressed at requirements, design, coding and testing stages. In addition, the authors proposed a green analysis stage to determine the greenness of each increment of the system under development [17].

Based on literature reviews, Chitchyan et al. [12] analyzed practices in software product line domain to understand sustainability concerns software developers deal with. The authors grouped the factors that influence sustainability according to dimensions and analyzed both the cross-dimensional dependencies and the influence of stakeholders [12]. Finally, a mapping study about interactions between software quality and environmental sustainability [20] organizes methods found in SE knowledge areas.

In addition to the roles specified in ISO/IEC 29110 [13], this proposal suggests that a sustainability expert role is needed. This role has gathered knowledge about sustainability in SE and they have the abilities to apply it in software projects. In addition,

he or she is responsible for supporting the development team and project manager as well as for keeping up the sustainability repository.

Table 1 depicts the sustainability tasks for the project management process. The first column displays an activity from ISO/IEC 29110 while the second column shows proposed sustainability tasks. The third column presents references. For instance, the 'Task 6. Identify sustainability training needs' was derived from references such as [23] who established that "training of personnel was regarded as very important. Training multiple persons on the topics decreases the key person risk". Similarly, [10] suggested that "it is necessary to educate … practitioners on the subject of sustainability design through formal education …, practice guidelines, demonstrative examples/case studies, and alike."

Table 2 presents proposed sustainability tasks for software implementation process. Task 4 "Analyze sustainability requirements considering their feasibility, risk, and potential interdependencies among dimensions" is based on Mahmoud and Ahmad model [17] who recommended performing risk analysis by taking into account energy efficiency and suggesting that feasibility analysis can be of help in identifying benefits related to improving energy efficiency. To support different sustainability goals, García-Mireles et al. [20] reported the usage of modeling languages to identify and analyze sustainability goals as well as the prioritization of sustainability requirements when conflicts among sustainability goals emerge.

In this proposal, we use the same ISO/IEC 29110 [13] products to address sustainability. Table 3 describes the products that need to include sustainability goals and specifications. The product number corresponds to the number described in ISO/IEC 29110 [13]. Thus, the ISO/IEC 29110 task list tables can be used to relate products with proposed sustainability tasks.

5 Initial Validation

To validate the feasibility of the proposed set of tasks, two papers (presented in Sect. 3, bullet 4) were reviewed in order to identify recommended practices to address sustainability and map them to proposed sustainability tasks. In the first paper [21] we identified 30 text segments that were categorized among activities such as software architecture and detailed design, software construction, among others (Table 4, Case A). The recommended practices were categorized in eight sustainability tasks from

Table 4. Software implementation process activities addressed by text fragments

Case study	Activity	Number of text fragments
Case A	Software architecture and detailed design	17
Case A	Software construction	8
Case A	Software implementation initiation	3
Case A	Software integration and test	2
Case B	Software requirements analysis	20

Table 2 (tasks number 2, 6, 7, 8, 9, 10, 11, and 13). The tasks with more text fragments were Task 6 (7 text fragments) and Task 10 (6 text fragments). The former task is centered on defining the software architecture whereas the latter refers to constructing software. An example of a text fragment for task 10 is as follows [21]: "Clean up useless code and data. …writing to never-read variables and other useless routines (such as repeated conditionals) might consume power purposelessly."

In the second paper [22] we identified 20 text segments that belong to software requirements analysis activity (Table 4, Case B). The text segments were categorized in Task 3 (15 text segments) and Task 4 (5 text segments). Task 3 is related to the identification and documentation of sustainability requirements considering sustainability goals. Task 4 focuses on analyzing sustainability requirements with regard to their feasibility and risk, and potential interdependencies among dimensions. An example of a Task 3 text fragment is as follows [22]: "Online questionnaire…Our goal was to elicit quantitative information about important aspects of sustainability requirements from a sample of involved people which is larger than the number of meal planners interviewed."

6 Discussion

This study proposed a sustainability extension to the ISO/IEC 29110 Basic Profile [13]. The sustainability extension addresses 14 tasks to be included in the project management process and the same number of tasks for addressing sustainability in the software implementation process. In comparison with the ISO/IEC 29110 Basic Profile which consists of 67 tasks, the 28 proposed tasks represent around 42% of new tasks. However, our proposal gives visibility to sustainability aspects during software development.

The theoretical proposal was validated by identifying sustainability practices in an existing case study and a framework for developing green software. The 50 text fragments were successfully categorized within 10 of the 14 sustainability tasks of the software implementation process. However, we have not found any reference to sustainability tasks related to the software project management process. In the literature, it is difficult to find empirical evidence for sustainability in software project management [1, 20]. Thus, we need to conduct empirical studies to validate the task extension proposed herein.

Considering internal validity threats, the definition of sustainability tasks was derived from at least two literature sources addressing similar practices. The proposed tasks are described on an abstraction level appropriate to be included into ISO/IEC 29110 [13]. Furthermore, to mitigate the misinterpretation of the proposed sustainability tasks, the second author classified text fragments. As a result, we achieved 86% of classification agreement, and all the inconsistencies were resolved.

7 Conclusions

Based on literature sources, this paper proposes a set of 28 sustainability tasks, which were mapped to the ISO/IEC 29110 Basic Profile process activities. An initial validation showed that our proposal can be used to map sustainability practices. However, this work is in an initial stage.

As future work, it is necessary to validate the sustainability tasks with research experts in sustainability and software processes. Afterwards, the sustainability extension should be applied in case studies to assess its practical feasibility and the required effort to implement the proposed tasks.

References

1. Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H., Franch, X.: Systematic mapping Study on Software Engineering for Sustainability (SE4S). In: ACM International Conference Proceeding Series, pp. 1–10 (2014)
2. Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S.M., Penzenstadler, B., Seyff, N., Venters, C.C.: Requirements: The key to sustainability. IEEE Softw. **33**(1), 56–65 (2016)
3. Penzenstadler, B., Femmer, H.: A generic model for sustainability with process- and product-specific instances. In: GIBSE 2013 - Proceedings of the 2013 Workshop on Green in Software Engineering, Green by Software Engineering, pp. 3–7 (2013)
4. Dick, M., Naumann, S.: Enhancing software engineering processes towards sustainable software product design. In: EnviroInfo, pp. 706–715 (2010)
5. Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., Venters, C.C.: Sustainability Design and Software: The Karlskrona Manifesto. In: Proceedings - International Conference on Software Engineering, pp. 467–476 (2015)
6. Calero, C., Piattini, M.: Puzzling out software sustainability. Sustain. Comput.: Inform. Syst. **16**, 117–124 (2017)
7. Calero, C., Piattini, M.: Introduction to green in software engineering. In: Calero, C., Piattini, M., (eds.). Green in Software Engineering, pp. 3–27 (2015)
8. Lago, P., Koçak, S.A., Crnkovic, I., Penzenstadler, B.: Framing sustainability as a property of software quality. Commun. ACM **58**(10), 70–78 (2015)
9. Venters, C.C., Jay, C., Lau, L.M.S., Griffiths, M.K., Holmes, V., Ward, R.R., Austin, J., Dibsdale, C.E., Xu, J.: Software sustainability: the modern tower of babel. In: CEUR Workshop Proceedings, pp. 7–12. (2014)
10. Chitchyan, R., Becker, C., Betz, S., Duboc, L., Penzenstadler, B., Seyff, N., Venters, C.C.: Sustainability design in requirements engineering: state of practice. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp. 533–542. ACM (2016)
11. Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., Pollock, L., Clause, J.: An empirical study of practitioners' perspectives on green software engineering. In: Proceedings of the 38th International Conference on Software Engineering, pp. 237–248. ACM, Austin (2016)
12. Chitchyan, R., Groher, I., Noppen, J.: Uncovering sustainability concerns in software product lines. J. Softw.: Evol. Process. **29**(2), 1–20 (2017)

13. ISO: ISO/IEC TR 29110-5-1-2 Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile, Geneva (2011)
14. Penzenstadler, B., Bauer, V., Calero, C., Franch, X.: Sustainability in software engineering: a systematic literature review. In: IET Seminar Digest, pp. 32–41 (2012)
15. García-Mireles, G.A.: Environmental Sustainability in Software Process Improvement: a Systematic Mapping Study. In: International Conference on Software Process Improvement, pp. 69–78. Springer (2016)
16. Naumann, S., Dick, M., Kern, E., Johann, T.: The GREENSOFT Model: a reference model for green and sustainable software and its engineering. Sustain. Comput.: Inform. Syst. **1**(4), 294–304 (2011)
17. Mahmoud, S.S., Ahmad, I.: A green model for sustainable software engineering. Int. J. Softw. Eng. Its Appl. **7**(4), 55–74 (2013)
18. Lami, G., Fabbrini, F., Fusani, M.: Software sustainability from a process-centric perspective. In: Winkler, D., O'connor, R.V., Messnarz, R. (eds.) Systems, software and services process improvement. European Conference on Software Process Improvement. 301 CCIS, pp. 97–108. (2012)
19. ISO/IEC: Systems and software engineering – software life cycle processes - Redline. ISO/IEC 12207:2008(E) IEEE Std 12207–2008 – Redline, pp. 1–195 (2008)
20. García-Mireles, G.A., Moraga, M.Á., García, F., Calero, C., Piattini, M.: Interactions between environmental sustainability goals and software product quality: a mapping study. Inf. Softw. Technol. **95**, 108–129 (2018)
21. Ardito, L., Procaccianti, G., Torchiano, M., Vetrò, A.: Understanding green software development: a conceptual framework. IT Prof. **17**(1), 44–50 (2015)
22. Huber, M.Z., Hilty, L.M., Glinz, M.: Uncovering sustainability requirements: an exploratory case study in canteens. In: CEUR Workshop Proceedings, pp. 35–44 (2015)
23. Groher, I., Weinreich, R.: An interview study on sustainability concerns in software development projects. In: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 350–358. IEEE (2017)
24. Dick, M., Drangmeister, J., Kern, E., Naumann, S.: Green software engineering with agile methods. In: 2013 2nd International Workshop on Green and Sustainable Software (GREENS), pp. 78–85. IEEE (2013)

Methodologies, Methods, Techniques and Tools Used on SLR Elaboration: A Mapping Study

Marco Palomino, Abraham Dávila(✉), and Karin Melendez

Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima, Peru
{palomino.marco, abraham.davila, kmelendez}@pucp.edu.pe

Abstract. The aim of this study is to perform a Systematic Literature Mapping (SLM) about methodologies, methods, techniques and tools used on the development of Systematic Literature Reviews (SLR). As a result, on the mapping, we expected to find and classify methodologies, methods, techniques and tools commonly used on SLR. In addition, we have considered other contexts such as, Medicine or Education with the purpose of getting multiple methodologies, methods, techniques and tools that allow performing SLR on efficient ways. It is mainly expected to identify techniques related to research questions formulation and the methods used for building search strings in order to get the higher number of studies associated to the research topic. In our study, we found multiple methodologies, methods, techniques and tools already implemented for performing SLRs On the mapping we describe some of them to highlight the most used and referenced studies.

Keywords: Systematic Literature Review · Methodologies · Methods Techniques

1 Introduction

The research, as well as other activities, should be supported by methodologies, methods, techniques and tools among others to obtain the expected results in a systematic way. In particular, the Systematic Literature Review (SLR) in Software Engineering domain is a case where research teams use methodologies, techniques and tools to help them on reducing time and effort in the research [1, 2]. The SLR is a type of secondary study [3] based on (i) the selection of primary studies of digital libraries and repositories; (ii) the analysis of the data or evidence collected and (iii) the preparation of the answers to the research questions.

Multiple studies have researched tools or search-strategies on a specific SLR activity [4–6]. In fact, the study [7] is a mapping about tools used in SLRs.

In our research, we expanded the previous studies finding also methodologies, tools, methods and techniques commonly used on SLR process. In addition, this approach will help future researchers to work on SLR efficiently due to it will focus on such activities that are manual and intensive such as Planning and Selection.

J. Mejia et al. (Eds.): CIMPS 2018, AISC 865, pp. 14–30, 2019.
https://doi.org/10.1007/978-3-030-01171-0_2

In this context, the purpose of our study was to identify the methodologies, methods, techniques and tools used in the preparation of an SLR, in different contexts such as medicine or education. Additionally, we wanted to identify the available techniques and tools that improve SLR performance. For this study a Systematic Mapping of the Literature (SML) was carried out in the relevant digital databases such as ACM Digital Library, IEEE Xplore, ProQuest, EBSCO, Emerald and Springer.

The present study is organized as follows: Sect. 2, SLR fundamentals are presented; Sect. 3, describes the methodology and the considerations used in the elaboration of the research questions; Sect. 4, results are presented associated with the research questions; and in Sect. 5, final discussion and future work are presented.

2 Systematic Literature Review Fundamentals

Systematic Literature Reviews (SLR) have recently been introduced to the context of Software Engineering [8, 9] as a structured methodology to perform literature reviews based on similar models from other contexts, such as education or medicine [10].

Evidence-Based Software Engineering (EBSE) guides that have been appearing in the academic field allow to systematize a literature review setting out a scientific model, which allows, to identify new areas of research, on deepen existing areas and generate knowledge bases [11, 12] that will help future researches in the Software Engineering area [13, 14].

Systematic Literature studies, such as mappings or revisions, have emerged as a way of synthesizing part of the vast scientific evidence that is increasing as new research in the area that occurs [8]. As it is mentioned on [15], these studies, allowing future researchers to take a point of reference and a conceptual framework for the development of their studies. In addition, the SLRs are useful to identify literature and research gaps relevant in a topic of interest [1, 9]. The three main phases of an SLR include planning, conducting and reporting [11].

3 Systematic Literature Mapping

This section presents the SLM fundamentals taken into account and its application in this study. Also, in this section, we describe the planning of the SLM based on guidelines proposed by Petersen described in [16], and recommendation from SLR [8, 11, 17].

3.1 Systematic Literature Mapping Fundamentals

A SLM study provides a structure of the research reports and results that have been published using visual summary of their results [17]. It often requires less effort while providing a more solid overview. Previously, studies of systematic mappings in Software Engineering have been recommended mainly in the areas of research where there is a need for relevant primary studies [11, 16].

In addition, in order to frame the research question and define the search string, it was used the PICO (Population, Intervention, Comparison, Outcome) criteria applied to Software Engineering [11]. In this study, we have elaborated eight research questions that are displayed in the Sect. 4. Finally, the Table 1 shows the principal keywords and the Table 2 indicates the search strings elaborated.

Table 1. Principal keywords used based on PICO criteria

Population	Systematic literature review
Intervention	Techniques, practices, tools, methodology
Comparison	None
Outcome	Evaluation, experiences, classification

Table 2. Search strings

Data source	Search string
IEEE Xplore	(“Systematic Literature” OR “Literature Review”) AND (“Methodology” OR “Protocol” OR “Guide”) AND (“Conduct*” OR “Perform”)
ProQuest	(“Systematic Literature” “Literature Review” “search engine”) AND (Methodology protocol guide) AND (“conduct” “search”) AND (tool*)
EBSCO	(“Systematic Literature” “Literature Review”) AND (Methodology protocol guide) AND (“conduct” “search”) AND (Tool)
ACM Digital Library Emerald	(“Systematic Literature” “Literature Review” “search engine”) AND (Methodology protocol) AND (conduct search) AND (tools) Abstract:(“literature review” AND “perform”) AND Anywhere(“methodology” AND “tool” AND “search”)
Springer	“Methodology” AND “Systematic Literature Review” AND (“tools” OR “technique” OR “Conduct” OR “Perform”)

The search strings were adapted to the specific syntax of data sources. Also, asterisks (*) were used in the strings to obtain the greatest number of occurrences without explicitly considering the plurals and words derived from the main concepts. These characters allow the engines of the databases to obtain the different possible combinations based on the base word that the asterisk refers.

3.2 SLM Protocol

A SLM protocol was defined to reduce the possibility of researcher bias. This protocol described in [16], which mention several steps that help to conduct a SLM, was structured in five main steps that included:

Step 1: Studies were selected regarding the execution of search string.
Step 2 and 3: Both steps were related to the process of including and excluding articles using the criteria defined. Then, we analyzed the titles and abstracts.
Step 4: Then, the review of article Introduction and Conclusion
Step 5: At the end of the review, the final articles were verified by peer review to evaluate their exclusion or inclusion in our research

The exclusion and inclusion criteria considered were:

- Inclusion criteria: We have considered academic articles with methodological support (controlled experiments, case studies, and systematic reviews, systematic mapping or others). In addition, we included studies extracted from the mentioned digital databases. Studies will be accepted if in their content propose techniques, guides or methods to conduct an SLR (mainly, the preparation of research questions and search strings). Studies that present cases of success in the use of alternative techniques or protocols were also included.
- Exclusion criteria: We have not considered duplicate studies. Additionally, articles that are contained in subsequent articles were also excluded. Also, we didn't consider studies that are not part of the following types of publications: journals, conferences and digital databases. In addition, studies that mention techniques or procedures that do not present concrete results are omitted. Studies whose title is irrelevant or outside the context of the SLR were excluded. Finally, tertiary studies and conference abstracts weren't considered.

3.3 Quality Assessment

Quality assessment of this SLM followed 11 criteria defined by [18] based on [19]. The following are the criteria used in the quality assessment:

- Is this study based on research?
- Is there a clear statement of the aims of the research?
- Is there an adequate description of the context?
- Was the study design appropriate to address the aims of the research?
- Was the selection strategy appropriate to the aims of the research?
- Was there a control group for comparing treatments?
- Was the data collected in a way that addressed the research aims?
- Was the data analysis rigorous enough?