UNIVERSITY OF MANNHEIM

Enhancing Sustainable Software Engineering Approaches / Models through SECoMo

Seminar Thesis

submitted: May 2017

by: Natalie Buchner
nbuchner@mail.uni-mannheim.de
born July 12th 1993
in Bonn

Student ID Number: 1496726

University of Mannheim
Chair of Software Engineering
D – 68159 Mannheim
Phone: +49 621-181-3912, Fax +49 621-181-3909
Internet: `http://swt.informatik.uni-mannheim.de`

Abstract

Sustainability is a central topic governments, businesses and communities globally are dealing with today. Its most prominent aspect is ecological sustainability and the need to fight global warming, but it also concerns social and economic issues. Many factors come into play that have a negative impact on sustainability, for example an increase in energy consumption or pollution. The Information Technology (IT) sector contributes to these negative factors as well, a main reason being the growing energy consumption caused by IT hardware. But software can also have negative impacts on (mainly) ecological sustainability – directly and indirectly. Thus, it is equally important to consider how to increase the sustainability of software.

The growing field of sustainable software engineering deals with the questions of how to develop sustainable software and how to develop it in a sustainable way. It covers aspects in all life cycle phases of a software. Existing research proposes a number of sustainability metrics, measurement tools or process models, but despite this variety of approaches, it seems that sustainable software engineering is not yet well established in practice. Possible reasons are the very specific character of most existing tools and measures, and the rather abstract and general character of life cycle models, with concrete methods of calculating and reducing ecological costs missing.

The Software Eco-Costs Model (SECoMo) approach by Thomas Schulze [5] is a new estimation approach in this field which allows to estimate the ecological costs of software already from an early stage on in a software project and to represent those costs and their causes in a comprehensible and clear way. With this, it enables stakeholders to have an early understanding of the sustainability impact of a software and to make design decisions accordingly. [5]

The purpose of this seminar thesis is to consider how SECoMo can be integrated with other existing sustainable software engineering approaches and how it can contribute to improving sustainable software engineering in practice.

As SECoMo can be integrated in all development phases, especially the early

ones, it can help to enhance existing life cycle models with a specific method for understanding and improving ecological sustainability in the design and implementation phases of software engineering. In addition, with its new set of sustainability metrics, SECoMo offers new options for sustainability measurement in existing models and tools, as they which base on a general way of software specification that can reasonably be applied in practice.

Contents

List of Figures

List of Tables

1. Introduction

Under the slogan "‘Sustainable Development Goals - 17 goals to transform our world"’[3], the United Nations are raising awareness for Sustainability and the need for everybody's action in order to "‘shift the world on to a sustainable and resilient path"’ [6]. The 17 goals mentioned are a central part of the sustainable development agenda published by the United Nation in 2015 and are essential in order to achieve sustainability with the main topics of "‘end[ing] poverty, protect[ing] the planet and ensur[ing] prosperitiy for all"’[3].
These ambitious goals emphasize how important the topic of Sustainability has become for the world, which is facing global challenges such as extreme poverty, climate change or war and injustices. Despite the global relevance of these issues, the scope of these goals reveals that taking action for sustainability affects everybody: "‘governments, the private sector, civil society"’[3] and every private person.

This is especially true when considering the causes for these challenges, spanning for example corrupt governments that cause injustice in their countries, or the ever increasing pollution and carbon dioxide emissions caused by various industries in the private sector. Many individuals and groups have a negative impact on sustainability with their actions - but on the other hand, there is also a great potential for positive impacts, starting from there.
Embracing sustainability has become more important in many areas over the last decades . Accordingly, many industries in the private sector have started to consider how to reduce their impacts on sustainability, often with a focus on ecological sustainability.

The ICT sector is no exception: it contributes to high power consumption and carbon dioxide emission with its "‘products"’ - but in the last few years, many efforts have been made to counteract these negative impacts, for example by improving energy efficiency of hardware products or by creating software solutions that support more eco-friendly business processes.

However, the focus of these efforts is still mainly on reducing the negative impacts of hardware artifacts, during its production and, more importantly, usage phase. But software artifacts can have an impact on sustainability issues, too - not only by enabling people or processes using it to be more sustainable, but also as a product itself, which for example impacts energy consumption during usage, or even during its development process. Thus, efforts directed at improving software sustainability during its development and usage phases are equally important in terms of IT Sustainability. And the research area of Sustainable Software Engineering has actually /indeed started to grow significantly over the past five to six years. There is a growing number of papers covering ideas for example on sustainability guidelines for software, models for sustainable development life cycle models, abstract software sustainability metrics or specific measurement tools for certain types of applications.

What is lacking are concrete applications of these guidelines, models and metrics in practical projects and evaluations of the validity and practicality of these various approaches. Reasons for that might be the fact that it is a research field that has only emerged during the past couple of years and is still in its first steps . Another aspects though might be that there are still more concrete approaches and metrics missing, that are generic enough to be applied to various software engineering projects, but specific enough to be directly used in practice.

This paper aims at comparing a new approach in the field of Sustainable Software Engineering, SECoMo by Thomas Schulze (2016) , with the existing variety of models and approaches in the field. The goal is to work out how this approach for estimating the ecological costs of a software system can enhance existing approaches, models or metrics in order to enable sustainable software engineering to be more likely to be applied in practical projects.

The relevant background topics for this seminar thesis, Sustainability and the relationship between Sustainability and ICT / IT, are introduced in the first chapters including definitions. In the following, the related work from the field of Sustainable Software Engineering is described, covering the most important existing guidelines, methods, models and metrics. In addition, the SECoMo approach is introduced and described with a focus on its general purpose and motivation as well as its most relevant elements and how to apply SECoMo within a software development process.

Based on these foundations, SECoMo is then considered in comparison to existing methods and approaches in order to identify benefits of this approach for the field of Sustainable Software Engineering.

2. Background

2.1. Sustainability

Although it is clear that Sustainability is a concept of increasing importance in the world and it has wide-ranging influences, on the way countries are lead, companies are behaving/acting/strategically aligning themselves(?) or which causes societies are engaging for(?) for example, the definition of the term Sustainability itself is not as clear and not always unified / consistent / coherent?: There are various attempts at understanding and defining the concept of Sustainability, coming from different perspectives which influence the focus of the definition (e.g. the needs of humans or of nature [1]) and thus it can be hard to find a common understanding [2]. In order to successfully work towards a sustainable future, finding this common understanding is absolutely necessary, though [2].

However, most attempts at defining the concept of Sustainability start with the well-known(?) definition of the concept of *Sustainable Development* by the World Commission / UN , also known as the *Brundtland Report*: "'Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs"'[4] . In addition, this report(?) mentions (3 dimensions?)...

Based on this understanding of what Sustainable Development is and which dimensions it covers, the concept of Sustainability can be defined as "'a holistic concept that embraces environmental, social and economic factors which lead to a decent life for the current generation while maintaining natural, social and economic resources so that future generations are not limited in living the same decent life"'.This definition understands/recognizes(?) Sustainability as a concept that enables Sustainable Development which a focus on human needs, but not limited to it.

2.2. Sustainability and IT

As mentioned before , the Information and Communication Technology (ICT) sector/industry plays, as many industries do, an important role in achieving Sustainability. Even though this covers all three dimensions, the focus is often on ecological sustainability, leading to the frequent use of the term *Green IT* covering this aspect of IT Sustainability.

3. Related work in Sustainable Software Engineering

3.1. state of research

3.1.1. Principles, Guidelines, Best practices

3.1.2. Approaches / different stages

3.1.3. Models and metrics

3.1.4. case studies, concrete examples, practical adoption?

3.2. The SECoMo approach

A relatively new addition to the field of sustainable software engineering approaches is the **Software Eco-Cost Model** (SECoMo) Approach by Thomas Schulze [5]. This approach provides Software Engineers with generic models and metrics necessary to estimate and express the ecological costs a software system causes when it is used [5]. Thus, SECoMo represents a concrete estimation approach for the impact a software system has regarding ecological sustainability during its usage phase.

3.2.1. What is SECoMo about?

The main motivation behind SECoMo is to provide an approach that allows to not only measure the ecological costs that are actually caused by a software system, but also to be able to estimate those costs upfront, for example already during the design phase of a software engineering project [5]. In order to achieve this, SECoMo offers a set of mathematical models which allow to precisely calculate eco-cost metrics, based on information that is already available in the design phase: specification models that describe the functionality, behavior and structure of the software system [5]. Furthermore, SECoMo is intended to be highly adaptable in order to allow the estimates to be calculated for different levels of details available - from an early level where only very general information about the software system is available, over an intermediate level with partly more detailed information, to an advanced level with very specified details that allow for more accurate estimates [5].
In addition to the mathematical models, the SECoMo approach also defines a set of eco-cost drivers in order to identify causes for certain ecological impacts a software system has and to better describe under which circumstances they occur [5]. The auxiliary models used in SECoMo which extend the specification models provide information about these cost drivers, but can also be used to express the estimated eco-costs of the software system [5]. This way, SECoMo additionally offers a possibility to communicate estimated or measured eco-costs to stakeholders of a project which can use this information to make improved decisions [5].
Against this background, the SECoMo approach is intended to be used in the early stages of software engineering projects to create estimates about the eco-

logical impact of a software system, so as to enable transparency about the sustainability aspect right from the start [5]. This again makes it possible for software engineers and other stakeholders to make decisions about changes to the software at the design stage which take the impact on ecological costs into account - be it to improve certain eco-cost critical aspects of the software because ecological sustainability is a major concern, or to at least be aware of the eco-cost trade-offs other decisions cause that might be motivated by other concerns, e.g. profitability. [5]
But SECoMo can also be used in the context of defining requirements for a software system, for example in terms of specifying upper bounds for the eco-cost metrics that must not be exceeded, or even to calculate exact eco-costs if enough details are given, e.g. for the specific usage scenario of a software system in a certain environment [5].

3.2.2. How can SECoMo be integrated in a software engineering project?

On a very high level, the SECoMo approach, when being used within a software engineering project in order to estimate the eco-costs, can be structured as follows:

1. calibrate the models so that the information about the hardware aspects is accurate and relates to a given situation
2. prepare the auxiliary models so that they include all information available (or that can be derived, like frequency on the advanced level)
3. calculate the estimates for the different eco-cost metrics based on the available information and the specified parameter values identifying

4. Contributions

4.1. Why sustainable software engineering is not yet used in practice(?!)

4.2. SeCoMo in the context of Sustainable SW Engineering

5. Findings(?)

6. Conclusion

Bibliography

[1] Thomas N. Gladwin, James J. Kennelly, and Tara-Shelomith Krause. Shifting paradigms for sustainable development: Implications for management theory and research. *Academy of management Review*, 20(4):874–907, 1995.

[2] Dale Jamieson. Sustainability and beyond. *Ecological Economics*, 24(2):183–192, 1998.

[3] Florencia Soto Nino. Sustainable development goals - United Nations, 2017.

[4] World Commission on Environment and Development. Our common future. also: Brundtland report, United Nations, 1987.

[5] Thomas Schulze. *A Cost model for Expressing and Estimating Ecological Costs of Software-Driven Systems*. PhD thesis, Universität Mannheim, Mannheim, 2016.

[6] United Nations. Transforming our world: the 2030 agenda for sustainable development. New York, 2015.

Appendix

A. First class of appendices

A.1. Some appendix

This is a sample appendix entry.

Eidesstattliche Erklärung

Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen.

Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann. Mir ist bekannt, dass von der Korrektur der Arbeit abgesehen werden kann, wenn die Erklärung nicht erteilt wird.

Mannheim, April 22, 2017 Unterschrift

Abtretungserklärung

Hinsichtlich meiner Studienarbeit/Bachelor-Abschlussarbeit/Diplomarbeit räume ich der Universität Mannheim/Lehrstuhl für Softwaretechnik, Prof. Dr. Colin Atkinson, umfassende, ausschließliche unbefristete und unbeschränkte Nutzungsrechte an den entstandenen Arbeitsergebnissen ein.

Die Abtretung umfasst das Recht auf Nutzung der Arbeitsergebnisse in Forschung und Lehre, das Recht der Vervielfältigung, Verbreitung und Übersetzung sowie das Recht zur Bearbeitung und Änderung inklusive Nutzung der dabei entstehenden Ergebnisse, sowie das Recht zur Weiterübertragung auf Dritte.

Solange von mir erstellte Ergebnisse in der ursprünglichen oder in überarbeiteter Form verwendet werden, werde ich nach Maßgabe des Urheberrechts als Co-Autor namentlich genannt. Eine gewerbliche Nutzung ist von dieser Abtretung nicht mit umfasst.

Mannheim, April 22, 2017 Unterschrift