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 (2016)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. (Schulze, 2016)

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

Abstract . . . iii

List of Figures . . . vii

List of Tables . . . ix

List of Abbreviations . . . x

1. Introduction . . . 1

2. Background . . . 3
2.1. Sustainability and Sustainable Software Engineering . . . 3
2.2. Related work in Sustainable Software Engineering . . . 3
2.2.1. Principles and Manifestos . . . 3
2.2.2. Approaches / different stages . . . 3
2.2.3. Models and metrics . . . 3
2.2.4. The SECoMo approach . . . 3

3. Contributions . . . 5
3.1. Why sustainable software engineering is not yet used in practice(?!) 5
3.2. SeCoMo in the context of Sustainable SW Engineering . . . 5

4. Findings(?) . . . 7

5. Conclusion . . . 9

Bibliography . . . 11

Appendix 13

A. First class of appendices 15
A.1. Some appendix 15

List of Figures

List of Tables

1. Introduction

2. Background

2.1. Sustainability and Sustainable Software Engineering

2.2. Related work in Sustainable Software Engineering

2.2.1. Principles and Manifestos

2.2.2. Approaches / different stages

2.2.3. Models and metrics

2.2.4. 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 [1]. 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 [1]. Thus, SECoMo represents a concrete estimation approach for the impact a software system has regarding ecological sustainability during its usage phase.

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 [1]. 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 [1]. 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 [1].
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 [1]. 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 [1]. 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 [1].
Against this background, the SECoMo approach is intended to be used in the early stages of software engineering projects to create estimates about the ecological impact of a software system, so as to enable transparency about the sustainability aspect right from the start [1]. 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. [1]

3. Contributions

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

3.2. SeCoMo in the context of Sustainable SW Engineering

4. Findings(?)

5. Conclusion

Bibliography

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

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 9, 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 9, 2017 Unterschrift