IDENTIFYING SELF-ADMITTED TECHNICAL DEBT USING NATURAL LANGUAGE PROCESSING TECHNIQUES

Everton da S. Maldonado

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science in Software Engineering
Concordia University
Montréal, Québec, Canada

July 2016

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Everton da S. Maldonado**
Entitled: **Identifying Self-Admitted Technical Debt Using Natural Language Processing Techniques**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Software Engineering

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

______________________________ Chair
______________________________ Examiner
______________________________ Examiner
______________________________ Examiner
Dr. Emad Shihab______________________ Supervisor

Approved ______________________________
Chair of Department or Graduate Program Director

____________ 20 ______ ______________________________
Dean
Faculty of Engineering and Computer Science

Abstract

Identifying Self-Admitted Technical Debt Using Natural Language Processing Techniques

Everton da S. Maldonado

Technical debt is a metaphor coined to express the trade off between productivity and quality, e.g., when developers take shortcuts or perform quick hacks during the development of software projects. These non optimal solutions are often implemented to allow the project to move faster in the short term, at the cost of increased maintenance in the future. The accumulation of technical debt during the ever changing life-cycle of a project is unavoidable, and if not properly managed can severely hinder the development of the project. To help alleviate the impact of technical debt, a number of studies focused on the detection of technical debt. However, a recent study has shown that one possible source to detect technical debt is using source code comments, also referred to as self-admitted technical debt. Therefore, in this dissertation we use empirical studies and NLP techniques to propose an approach to automatically identify self-admitted technical debt.

First, we examine source code comments to determine the different types of technical debt, and we propose four simple filtering heuristics to eliminate comments that are not likely to contain technical debt. Then, we read through more than 33K comments, and we find that self-admitted technical debt can be classified into five main types - design debt, defect debt, documentation debt, requirement debt and test debt. In addition, the most common type of self-admitted technical debt is design debt, making up between 42% to 84% of the classified comments. We also make the dataset used in this study publicly available to support future research in the area.

Second, we leverage the knowledge obtained in our first study to present an approach to automatically identify design and requirement self-admitted technical debt using Natural Language Processing (NLP). We study 10 open source projects: Ant, ArgoUML, Columba, EMF, Hibernate, JEdit, JFreeChart, Jmeter, JRuby and SQuirrel SQL and find that 1) we are able to effectively identify self-admitted technical debt, significantly outperforming state-of-the-art techniques; 2) that words related to sloppy or mediocre source code are the best indicators of design debt, whereas for requirement debt, words related to enhancing or completing tasks are the best indicators; and 3) we can effectively identify self-admitted technical debt using as little a small training dataset of 1,444 comments for design debt comments and 380 comments to detect requirement debt.

Acknowledgments

At the conclusion of a stage, you must look back and take the time to thank and to be grateful for those who have been by our side, because happiness is meaningless without someone to share. In particular, I am grateful to God who gives me strength, guidance and above all the opportunity to pursue my masters degree.

I would like to express my sincere gratitude to my supervisor Dr. Emad Shihab for his support and dedication during this last two years. Thank you Emad for believing in my potential, for teaching me how to overcome my limitations and that, through hard work, everything is achievable. I must tell you that it has been quite an enjoyable adventure, and that a learned a lot from you.

I also would like to thank all my lab mates Moiz Arif, Davood Mazinanian, Ahmad Hassan, Samuel Donadelli, Shahriar Rostami, Rabe Abdalkareem and everyone else that I had the opportunity to work with. All of you are very special for me, and made this experience so much more enjoyable.

Contents

List of Figures

List of Tables