How social debt in software development affects your motivation in the graduation

Leonardo Lana Violin Oliveira,
oriented by: Alfredo Goldman and
Damian A. Tamburri

October 30, 2017

Introduction

The research has as its goal to verify if technical debts, that rise as progress towards the end of the assignment generates a social debt. Students not always implement their programs in an organized way or in a way that let the code's reading easy, which can generate a technical debt. With that in mind, I intend to study if these situations have an effect lowering students' enjoyment, well-being and attention for the assignments and even the course itself, therefore getting into a sub-optimal routine when developing the next assignments.

Methodology

The data gathering will be done through Github's private repositories, online forms and interviews with the students, who are attending to MAC0216 (**Programming Techniques I**) and agree with their data being collected.

The interviews will be made with each group in the following weeks after the deadline of assignment, and the Github's repositories will be checked weekly.

What motivated the choice of MAC0216 for the project was the fact that it will be the first time that first-year students will have contact with group assignments, therefore it is an environment prone to technical debts (bad modularized functions and files, etc.). Furthermore, I'm the teacher's assistant of this subject, so if necessary I will provide a *git* course to the students.

Form

The form in question is hosted at: `goo.gl/forms/xJYaQqsyvahguv8J3`

The form was divided into three parts: identification of the student (name and USP identification number), creating the student's profile (assertive questions) and final questions (discursive questions).

Assertive questions

This questions have the objective of mounting the student's profile, they ask:

1. Previous level of experience of programming
2. How organized is the code
3. If the code has commentaries
4. If there is a plan before programming
5. How much time was spent in refactoring
6. In which time frame the assignment is completed
7. How the assignment is developed

The goal of the first question is to better understand how the students perceive their experience, so we can take caution when inferring which basic programming techniques they know (commentaries structure, modularization, version control, etc.).

Questions 2, 4, 6 and 7 seek to understand how organized the student is, for example, if the student completes the assignment near the deadline and develop it in peaks, these answers may suggest that he/she is organized in the coding aspect as well as scheduling aspect.

Questions 2, 3, and 5 seek to understand how the student codes, if him/her is organized within the code's realm, he/she documents the code properly with commentaries and if in the sight of design problems, he/she applies refactoring.

Updates

I will post every other week an update, with the current data analysis and conclusions in a form of a **.pdf** file in this website: `leolanavo.github.io/MAC0215/`