

Technical Debt in Software Development: A Multi-Perspective Investigation

Abstract

Software products need to be constantly maintained and updated to keep being useful and satisfying companies' and users' needs. Developers are often required to perform software maintenance and evolution activities in the shortest possible time in order to make the changes available as soon as possible. As a result, they do not have the possibility to apply ideal development practices, thus introducing the so-called technical debt, i.e., the application of a quick and low-quality solution instead of a better one that would take longer. This will cause a decrease in software quality and require significant maintenance effort in the future. For this reason, identifying the symptoms of technical debt in advance is of fundamental importance for software companies. However, such symptoms could appear in different forms and at different stages of development, making harder their identification. In the context of this thesis, we face this challenge from several perspectives.

First, we focus on bad code smells, poor design or implementation choices applied in the source code by developers that have been associated with maintainability and understandability degradation. Over the last years, several researchers have been devising tools and techniques for the automatic detection of these design flaws. However, unfortunately, all the proposed detectors appear to be still too limited and inadequate to be applied in real industrial contexts. The first part of this thesis focuses on experimenting with the suitability of machine learning-based code smell detection techniques. Preliminary results demonstrate that machine learning-based techniques still have limited performance for automatic code smell detection, due to several limitations such as (i) the strongly unbalanced nature of the problem, (ii) the subjectivity of the results, and (iii) the limited set of metrics considered so far. This thesis investigates these three limitations separately, proposing specific solutions to overcome them. However, although some advantages have been reported, machine learning techniques still require more improvement to provide reliable detection of code smells.

Other than studying technical debt in production code, we also consider its presence, as well as its harmfulness, in test code. Testing activities seem to receive way lower attention during software development: tests are often developed without applying proper programming principles or automatically generated with the support of specific tools. Therefore, resulting test suites are often characterized by a low quality that could also reduce their effectiveness in bug discovery. This thesis faces this challenge by presenting a large-scale analysis of test code quality and effectiveness both in traditional systems and in mobile applications in order to understand the test-related factors that are most related to technical issues in production code. The main results confirm that test suites are characterized by a very low code quality and effectiveness, particularly with respect to mobile applications. Moreover, differently from what was previously stated in the literature, some of the quality aspects considered (e.g., size, test smells) have been shown to have a stronger correlation with production code defects as compared to traditional and widely-adopted coverage metrics.

Finally, we also include a discussion on the main lessons learnt and open issues together with some indications about further research directions.