A Case Study on the Impact of Technical Debt Management Efforts on Code Quality

Rafael Pinto  
University of Texas  
El Paso, Texas  
rafaelljpinto@gmail.com

Maliheh Zargaran  
University of Texas  
El Paso, Texas  
maliheh.zargaran@gmail.com

Omar Badreddin  
University of Texas  
El Paso, Texas  
obbadreddin@utep.edu

Abstract

Background. As software systems continue to grow in complexity, their maintenance effort becomes exponentially challenging. Software engineers, faced with tight schedules and constraints, may take technical shortcuts that can satisfy immediate objectives but are detrimental to the longevity of the software. Those technical shortcuts result in arbitrary complexities in the code and can make the code difficult to maintain in the future. Such avoidable complexities are often referred to as Technical Debt. Just like financial debt, technical debt accumulates interests and overtime if not addressed early in the software life cycle. Therefore, many organizations are actively managing Technical Debt by continuously monitoring code quality to ensure that Technical Debt remains manageable over time.

Study Design. The goal of this study is to understand the impact of active technical debt management. Towards that goal, we designed a study with the goal of assessing the impact of monitoring and managing technical debt in software projects. We also aim to understand how do software engineers perceive technical debt the effectiveness of continuously managing its growth throughout the project lifecycle. We selected four code repositories, two of which have been a subject to technical debt monitoring and management, and two were unmanaged. We measure technical debt over five-year period, and collect information about code smells in the subject repositories. We also designed a questionnaire and interviews with primary sources and software engineers working on the subject repositories.

Results and Analysis. The primary findings of the study suggest that technical debt management significantly reduce the accumulation of technical debt throughout the project lifecycle. The study also find that software engineers find that efforts to manage technical debt to be worthwhile. In the future, we will quantify the cost of technical debt management in hours, and quantify the impact on code maintainability.

Conclusion. Technical debt tends to grow uncontrollably over time and can be the main reason for failures, bugs, high maintainability costs and eventually the success or failure of the application and the company producing it. Monitoring technical debt can significantly reduce its growth and positively impact maintenance.