Abstract

Background: As software is everywhere, and almost every company has nowadays a dependency on software, designing and developing software-intensive products or services has become significantly challenging and time-consuming. The challenges are due to the continuous growth of size and complexity of software and the fast pace of change. It is important that software-developing organisations' engineering practises adapt to the rising challenges by adopting well-engineered development activities.

Organisations deal with a plethora of software artefacts; some of which are more valuable for the organisation. We define Software Assets as software artefacts that are intended to be used more than once. Given the continuous and evolutionary aspect of software development, the assets involved degrade over time. In order to exercise quality control over software assets, organisations need to understand what assets are relevant and how they degrade. Asset degradation is inevitable, and it may manifest in different ways.

Objective: The main objective of this thesis is two-fold: (i) to contribute to the software engineering body of knowledge by providing an understanding of what assets are and how they degrade; and (ii) to gather empirical evidence regarding asset degradation and different factors that might impact it on industrial settings.

Method: To achieve the thesis goals, several studies have been conducted, including a literature review, industrial studies, a field study, a sample study using archival analysis and focus groups as data collection methods. The collected data is from peer-reviewed literature and collaboration with five company partners that included extracting archival data from over 20 million lines of code, as well as archival data from open-source repositories from the Apache Foundation.

Results: The first contribution of this thesis is defining the concept of assets and asset degradation in a position paper. Our aim is to provide an

understanding of software assets and asset degradation and its impact on the development of software-intensive products or services. In addition, a taxonomy of assets has been created using academic and industrial input. The taxonomy includes 57 assets and their categories.

In order to further investigate the concept of asset degradation, we have conducted in-depth analyses of multiple industrial case studies on selected assets. This thesis presents results to provide evidence on the impact of different factors on asset degradation including: i) how the accumulation of technical debt is affected by different development activities; ii) how degradation 'survives'; and iii) how working from home or the misalignment between ownership and contribution impacts the faster accumulation of asset degradation. Additionally, we created a model to calculate the degree of the misalignment between ownership and contribution to code.

Conclusion: The results can help organizations at identifying and understanding the relevant software assets and characterise their quality degradation. Understanding how assets degrade and which factors might impact its faster accumulation is the first step to conduct sufficient and practical asset management activities. For example, by engaging i) proactively in preventing uncontrolled growth of degradation (e.g., aligning ownership and contribution); and ii) reactively in prioritizing mitigation strategies and activities (focusing on recently introducing TD items).

Keywords: *Assets in Software Engineering, Asset Management, Asset Degradation, Technical Debt*