Understanding Asset Degradation in Software Engineering

Ehsan Zabardast

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

Objective: The main objective of this thesis is: (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. The collected data is from peer-reviewed literature and collaboration with five companies that included extracting archival data from over 20 million LOC and archival data from open-source repositories.

Results: The first contribution of this thesis is defining the concept of assets and asset degradation in a position paper. We aim to provide an understanding of software assets and asset degradation and its impact on software development.

Additionally, a taxonomy of assets is created using academic and industrial input. The taxonomy includes 57 assets and their categories.

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 alignment between ownership and contribution to code.

Conclusion: The results can help organisations identify and understand the relevant software assets and characterise their quality degradation. Understanding how assets degrade and which factors might impact their faster accumulation is the first step to conducting 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 prioritising mitigation strategies and activities (focusing on recently introducing TD items).

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