

Breaking Bad? Semantic Versioning and Impact of Breaking Changes in Maven Central

Lina Ochoa¹, Thomas Degueule², Jean-Rémy Falleri^{2,3} and Jurgen Vinju^{1,4}

¹ Eindhoven University of Technology, Eindhoven, The Netherlands

² University of Bordeaux, Bordeaux INP, CNRS, LaBRI, Bordeaux, France

³ Institut Universitaire de France, Paris, France

⁴ Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

l.m.ochoa.venegas@tue.nl, thomas.degueule@labri.fr, falleri@labri.fr, and j.j.vinju@tue.nl

Abstract—This document contains the main information of the article *Breaking Bad? Semantic Versioning and Impact of Breaking Changes in Maven Central*, published on the journal of *Empirical Software Engineering* on March 17th, 2022. This article is now being submitted to the *Journal-First Papers track* at the *International Conference on Software Engineering 2023*.

I. ORIGINAL ABSTRACT

Just like any software, libraries evolve to incorporate new features, bug fixes, security patches, and refactorings. However, when a library evolves, it may break the contract previously established with its clients by introducing Breaking Changes (BCs) in its API. These changes might trigger compile-time, link-time, or run-time errors in client code. As a result, clients may hesitate to upgrade their dependencies, raising security concerns and making future upgrades even more difficult. Understanding how libraries evolve helps client developers to know which changes to expect and where to expect them, and library developers to understand how they might impact their clients. In the most extensive study to date, Raemaekers et al. investigate to what extent developers of Java libraries hosted on the Maven Central Repository (MCR) follow semantic versioning conventions to signal the introduction of BCs and how these changes impact client projects. Their results suggest that BCs are widespread without regard for semantic versioning, with a significant impact on clients. In this paper, we conduct an external and differentiated replication study of their work. We identify and address some limitations of the original protocol and expand the analysis to a new corpus spanning seven more years of the MCR. We also present a novel static analysis tool for Java bytecode, Maracas, which provides us with: (i) the set of all BCs between two versions of a library, and; (ii) the set of locations in client code impacted by individual BCs. Our key findings, derived from the analysis of 119,879 library upgrades and 293,817 clients, contrast with the original study and show that 83.4% of these upgrades do comply with semantic versioning. Furthermore, we observe that the tendency to comply with semantic versioning has significantly increased over time. Finally, we find that most BCs affect code that is not used

by any client, and that only 7.9% of all clients are affected by BCs. These findings should help (i) library developers to understand and anticipate the impact of their changes; (ii) library users to estimate library upgrading effort and to pick libraries that are less likely to break, and; (iii) researchers to better understand the dynamics of library-client co-evolution in Java.

II. ARTICLE'S LINK

The original article can be found at <https://link.springer.com/article/10.1007/s10664-021-10052-y>.