

## 1. Watch out for this commit! A study of influential software changes

Daoyuan Li (1); Li Li (1); Dongsun Kim (1); Bissyande, T.F. (1); Lo, D. (2); Le Traon, Y. (1)

**Source:** *Journal of Software: Evolution and Process*, v 31, n 12, p e2181 (25 pp.), Dec. 2019; **ISSN:** 2047-7473; **DOI:** 10.1002/smr.2181; **Publisher:** Wiley, USA

**Author affiliation:** (1) University of Luxembourg, Interdisciplinary Centre for Security, Luxembourg (2) Singapore Management University, School of Information Systems, Singapore

**Abstract:** One single code change can significantly influence a wide range of software systems and their users. For example, (a) adding a new feature can spread defects in several modules, while (b) changing an API method can improve the performance of all client programs. Unfortunately, developers often may not clearly know whether code changes are influential at commit time. This paper investigates influential software changes and proposes an approach to identify them immediately when they are applied. Our goals are to (a) identify existing influential changes (ICs) in software projects, (b) understand their characteristics, and (c) build a classification model of ICs to help developers find and address them early. We first conduct a post-mortem analysis to discover existing influential changes by using intuitions (eg, changes referred by other changes). Then, we re-categorize all identified changes through an open-card sorting process. Subsequently, we conduct a survey with about 100 developers to finalize a taxonomy. Finally, from our ground truth, we extract features, including metrics such as the complexity of changes and file centrality in co-change graphs to build machine learning classifiers. The experiment results show that our classification model with random samples achieves 86.8% precision, 74% recall, and 80.4% F-measure, respectively. (0 refs)

**Inspec controlled terms:** application program interfaces - feature extraction - learning (artificial intelligence) - pattern classification - software maintenance - software metrics

**Uncontrolled terms:** feature extraction - API method - classification model - machine learning classifiers - open-card sorting process - co-change graphs - software projects - code changes - software systems - influential software changes

**Classification Code:** C6110B Software engineering techniques - C6110S Software metrics - C6130 Data handling techniques - C6170K Knowledge engineering techniques - C0310F Software management

**IPC Code:** G06F7/00 - G06F9/44 - G06N5/04 - G06N20/00

**Treatment:** Bibliography (BIB) - Practical (PRA)

**Database:** Inspec

**Data Provider:** Engineering Village

Copyright 2020, The Institution of Engineering and Technology