

1. Mining android crash fixes in the absence of issue- And change-tracking systems (Open Access)

Kong, Pingfan (1); Li, L. (2); Gao, Jun (1); Bissyandé, Tegawendé F. (1); Klein, Jacques (1)

Source: *ISSTA 2019 - Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, p 123-133, July 10, 2019, *ISSTA 2019 - Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*; **ISBN-13:** 9781450362245; **DOI:** 10.1145/3293882.3330572; **Conference:** 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, July 15, 2019 - July 19, 2019;

Sponsor: ACM SIGSOFT; **Publisher:** Association for Computing Machinery

Author affiliation: (1) University of Luxembourg, Luxembourg (2) Monash University, Australia

Abstract: Android apps are prone to crash. This often arises from the misuse of Android framework APIs, making it harder to debug since official Android documentation does not discuss thoroughly potential exceptions. Recently, the program repair community has also started to investigate the possibility to fix crashes automatically. Current results, however, apply to limited example cases. In both scenarios of repair, the main issue is the need for more example data to drive the fix processes due to the high cost in time and effort needed to collect and identify fix examples. We propose in this work a scalable approach, CraftDroid, to mine crash fixes by leveraging a set of 28 thousand carefully reconstructed app lineages from app markets, without the need for the app source code or issue reports. We developed a replicative testing approach that locates fixes among app versions which output different runtime logs with the exact same test inputs. Overall, we have mined 104 relevant crash fixes, further abstracted 17 fine-grained fix templates that are demonstrated to be effective for patching crashed apks. Finally, we release ReCBench, a benchmark consisting of 200 crashed apks and the crash replication scripts, which the community can explore for evaluating generated crash-inducing bug patches. © 2019 Association for Computing Machinery. (48 refs)

Main heading: Android (operating system)

Controlled terms: Accidents - Computer debugging - Digital storage - Program debugging - Program documentation - Software testing - Testing - Tracking (position)

Uncontrolled terms: Android - Android apps - Change tracking - Crash - Fine grained - Mining software repositories - Scalable approach - Source codes

Classification Code: 722.1 Data Storage, Equipment and Techniques - 723 Computer Software, Data Handling and Applications - 914.1 Accidents and Accident Prevention

Funding Details: Number: 15/IS/10449467, Acronym: FNR, Sponsor: Fonds National de la Recherche Luxembourg; Number: C17/IS/11693861, Acronym: FNR, Sponsor: Fonds National de la Recherche Luxembourg;

Funding text: This work was supported by the Fonds National de la Recherche (FNR), Luxembourg, under CORE projects CHARACTERIZE C17/IS/11693861 and RECOMMEND 15/IS/10449467, and AFR-PhD project 11620657.

Open Access type(s): All Open Access, Green

Database: Compendex

Data Provider: Engineering Village

Compilation and indexing terms, Copyright 2022 Elsevier Inc.