Scopus

Document details

< Back to results | 1 of 1

Export Download Print E-mail Save to PDF Add to List More... >

Full Text View at Publisher

Journal of Computer Science Open Access
Volume 15, Issue 1, 2019, Pages 92-107

A static analysis of android source code for lifecycle development usage patterns (Article) (Open Access)

Hoshieah, N.[a], Zein, S.[b], Salleh, N.[c], Grundy, J.[d]

[a]Software Engineering Master Program, Birzeit University, Birzeit, Palestine
[b]Department of Computer Science, Birzeit University, Birzeit, Palestine
[c]Department of Computer Science, International Islamic University, Kuala Lumpur, Malaysia

View additional affiliations

Abstract

View references (23)

Building robust Android apps is a non-trivial task that requires skilled developers to understand various Android platform peculiarities. However, among the Android developers community, a large fractions are considered to be novice and inexperienced developers. One of the main peculiarities in the Android app development is the activity lifecycle model. A developer needs to have deep understanding of the different lifecycle states and callback methods that an Android activity can go through during its runtime. These callback methods are called by the system whenever an app activity changes its state. The developer needs to override appropriate callback methods correctly to avoid app memory leaks and data loss or other phone resource compromise. Detailed static analysis of software applications provides actionable insights and helps us to understand how applications are actually built. Although there have been many studies focusing on static analysis of Android apps in the areas of testing, quality, design, privacy and security; no studies to date focus on lifecycle development practices and usage patterns thus far. In this paper, we analyzed 842 open- source Android apps containing 5577 activities to explore and understand how Android developers actually comply with best practices regarding the Android activity lifecycle model. We developed a tool named SAALC that is capable of analyzing Android activities and extracting valuable information about lifecycle callback methods usage . Our results show, which callback methods are implemented and the nature of the code they contain. The results also show incorrect implementation of the callback methods and incorrect acquiring and releasing of system resources in many Android apps and we argue that a relatively large fraction of Android developers do not sufficiently well understand the app lifecycle model. We also discuss our results in comparison to the Android app lifecycle model best practices. © 2019 Noura Hoshieah, Samer Zein, Norsaremah Salleh and John Grundy.

SciVal Topic Prominence

Topic: Testing | Graphical user interfaces | Manual testing

Prominence percentile: 95.031

Author keywords

Activity lifecycle | Android | Application | Mobile Apps | Static analysis

ISSN: 15493636
Source Type: Journal
Original language: English

DOI: 10.3844/jcssp.2019.92.107
Document Type: Article
Publisher: Science Publications

Metrics

0	Citations in Scopus
0	Field-Weighted Citation Impact

PlumX Metrics

Usage, Captures, Mentions, Social Media and Citations beyond Scopus.

Cited by 0 documents

Inform me when this document is cited in Scopus:

Set citation alert >

Set citation feed >

Related documents

Andro Lyze: A Distributed Framework for Efficient Android App Analysis

Baumgartner, L. , Graubner, P. , Schmidt, N.
(2015) Proceedings - 2015 IEEE 3rd International Conference on Mobile Services, MS 2015

An approach for using mobile devices in industrial safety-critical embedded systems

Armoush, A. , Franke, D. , Kalkov, I.
(2014) Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST

Developer mistakes in writing android manifests: An empirical study of configuration errors

Jha, A.K. , Lee, S. , Lee, W.J.
(2017) IEEE International Working Conference on Mining Software Repositories