Noname manuscript No.
(will be inserted by the editor)

On the Relationship between the Number of Ad Libraries in an Android App and its Rating

**Israel J. Mojica Ruiz, Meiyappan Nagappan,
Bram Adams, Thorsten Berger, Steffen Dienst
and Ahmed E. Hassan**

Received: date / Accepted: date

Abstract One of the most popular ways to monetize a free app is by including advertisements in the app. There are several advertising (ad) companies that provide these ads to the app developers through ad libraries that need to be integrated in the app. However, the demand for ads far exceeds the supply for them. This obstacle may lead app developers to integrate several ad libraries from different ad companies in their app to assure receiving an ad when requested. However, there is no empirical evidence so far about how many ad libraries are integrated in an app. Additionally, there is no current research to examine if integrating many different ad libraries has an impact on the ratings of an app. In this paper, we examine these two issues, by empirically examining thousands of Android apps. We find that there are apps with as many as 28 ad libraries. We find no evidence that the number of ad libraries in an app is related to the ratings that an app can get. However, integrating certain specific ad libraries can negatively impact the rating of an app.

Keywords Mobile apps · Ad libraries · Android · Software economics · Ad maintenance

Israel J. Mojica Ruiz, Meiyappan Nagappan, Ahmed E. Hassan
Software Analysis and Intelligence Lab
Queen's University, Kingston, Canada.
E-mail: {mojica,mei,ahmed}@cs.queensu.ca

Bram Adams
Lab on Maintenance, Construction and Intelligence
École Polytechnique de Montréal, Canada.
E-mail: bram.adams@polymtl.ca

Thorsten Berger
Generative Software Development Lab
University of Waterloo, Canada.
E-mail: tberger@gsd.uwaterloo.ca

Steffen Dienst
Chair of Business Information Systems
University of Leipzig, Germany.
E-mail: sdienst@informatik.uni-leipzig.de

1 Introduction

Mobile apps are software applications for mobile devices, such as smartphones, tablets, and other personal digital assistants. The growing demand for mobile apps has led to rapidly increasing downloads—from 7 billion apps in 2009 to an estimated number of 56 billion apps across all platforms in 2013. This rapid growth is attracting many amateur and professional developers, who strive to gain profit from developing apps. A major way for developers to monetize their apps is through displaying advertisements to the end users. These ads are provided by several advertising (ad) companies such as Google's googleads and Flurry's AppSpot.

The primary way to integrate ads in a mobile app is through *ad libraries*, provided by ad companies. An app requests ads from an ad company through the corresponding ad library. However, the success rate in receiving an ad from an ad company when an ad is requested (*Fill Rate*) is quite low when it comes to ads in mobile apps. In the first half of 2011, the average fill rate for the world's top 40 ad networks was less than 18%. This low fill rate is mainly because the number of ads being requested by apps is increasing faster than the number of ads available in the market for serving [3]. Hence, in order to achieve a high fill rate, app developers can integrate ad libraries from different ad companies. This is because developers are not restricted to one ad company. Connecting with a large number of ad companies helps ensure a higher fill rate, and hence, higher revenues.

In this paper, we empirically examine if there exists a relationship between the number of ad libraries integrated in an app and the average rating of the app, as assigned by it users. Our work is along similar lines as the work by Godfrey and German [8], who looked at the relationship between software economics and software quality as influenced by software evolution, and Platzer [14], who examined how to price a particular app. In our work, we strive to give actionable recommendations to app developers about the effective use of ad libraries in apps, i.e., maximizing revenues without affecting the rating of an app in the app market.

Our contributions in this paper are three-fold. We find that:

1. app developers indeed integrate more than one ad library in their apps (as many as 28 ad libraries).
2. the number of ad libraries in an app is not related to the rating of an app.
3. integrating certain specific ad libraries can negatively impact the rating of an app.

2 Collecting Data for the Study

Similar to other work on mobile apps [7][9][13], we analyze apps from the popular Android platform. In particular, we analyze 519,739 app versions distributed in 236,245 different Android apps, which is a superset of the apps used in one of our prior studies (208,601 apps) on reuse in Android apps [13]. Our dataset covers 27 Google Play categories, with the Entertainment category having the most apps (over 20K), and the weather category having the lowest number of apps (still over 1K). We describe the study design of our work from the collection of the data to the preprocessing of it, so others can replicate our work.

2.1 Crawling Google Play

In our study, we use a dataset that we crawled from the official Google Play app store once a day during the first half of 2011, and twice a day during the second half of 2011. As a result, we obtained a set of 625,067 app versions distributed across 281,079 mobile apps. Google Play classifies apps into 27 different categories (including subcategories in "Games"). See a technical note [6] for more details. For each app version that was free to download, our dataset contained:

– **Android Packages (APK)**: The binary, in the Android-specific packaging format (APK).
– **Metadata**: App store specific information for each app version. For the present work, we only use the user rating (from one to five stars) of each app.

2.2 Extracting App Bytecode

App developers integrate the ad libraries into the APK, in order to use the APIs (Application Programming Interfaces) of the ad libraries. For our study, we need to identify which ad libraries are integrated in each app. Given that apps are packaged in the APK format, we only have access to the Dalvik bytecode of the apps, containing both application and library classes. We first use an open source tool [1] to extract the Java bytecode from the APKs. Then, we use the *Apache bcel* library [2] to extract the fully qualified class names (package or namespace in which a class is contained and the class name) and their corresponding set of method names (APIs) for each class, in each app. We drop apps with obfuscated class names. After this process, we end up with 519,739 app versions of 236,245 different apps. We store the fully qualified class names and their corresponding methods in a database (henceforth called **DB**). We use this data to identify the ad libraries in an app.

2.3 Identifying the Ad Libraries in each App

We filter the fully qualified class names of all apps looking for the regular expression [aA][dD] (*e.g.*, *com.package**Ad**libraryName.**Ad**className*) across all the apps in the **DB** created in Subsection 2.2. Notice that the regular expression [aA][dD], is very simple. Hence, it matched many class names across **DB**, even class names that were not necessarily a part of an ad library.

We group and sort the fully qualified classes according to their popularity in the **DB**. The most frequent fully qualified class is *com.google.ads.AdActivity* with 149,321 repetitions. We performed the following manual process to find the ad libraries:

(a) For each fully qualified class name, we perform a web search of the package name in order to find the website of the ad library provider (the ad company).

[1] dex2jar: `http://code.google.com/p/dex2jar`
[2] Apache bcel library: `http://commons.apache.org/bcel/`

We look for the ad library website in order to verify that the name of the library found was in fact from a real ad company. We expect that information about the ad library should exist if this is a real and trusted ad library because an app developer has to sign a contract with the ad company in order to receive ads, and later payments.

(b) If we find an ad library website, we add the package name to our set of ad libraries, otherwise we discard it.
(c) We filter out from the **DB** all the class names that belong to the package name identified in (a).
(d) We repeat this process for each remaining class name.

The process was repeated until a class name was repeated no more than 200 times in **DB**. This means that any library not integrated would only occur in less than 200 apps (0.08% of the studied apps). We stopped at this point, because there were still thousands (802,012) of fully qualified class names to verify that would not be necessarily part of an ad library. If they were, the presence of such an ad library would be in a minuscule set of apps.

Over half (51.21%) of the studied Android apps (236,245) have at least one ad library. We identify 72 different ad libraries that are going to be used in this study, and obtain the unique package name that identifies each one of the 72 ad libraries.

3 Do developers integrate more than one ad library in their apps?

Motivation: Recall that the fill rate (defined in Section 1) for ads in mobile apps is very low [3]. This situation may drive developers to integrate more than one ad library in an app to achieve a higher fill rate, and thus higher revenues. Consequently, we first determine the number of ad libraries that are integrated in each app in our dataset.

Results: We find that most apps have only one ad library in them. However, at least 42,206 (34.88%) of the apps with ad libraries in them have two or more ad libraries. Fig. 1 shows a breakdown of the number of apps based on the number of ad libraries bundled in the apps. We even find eight apps with as much as 28 ad libraries. More details on these eight apps can be found at the end of Section 4.

> *34.88% of the studied apps have more than one ad library.*

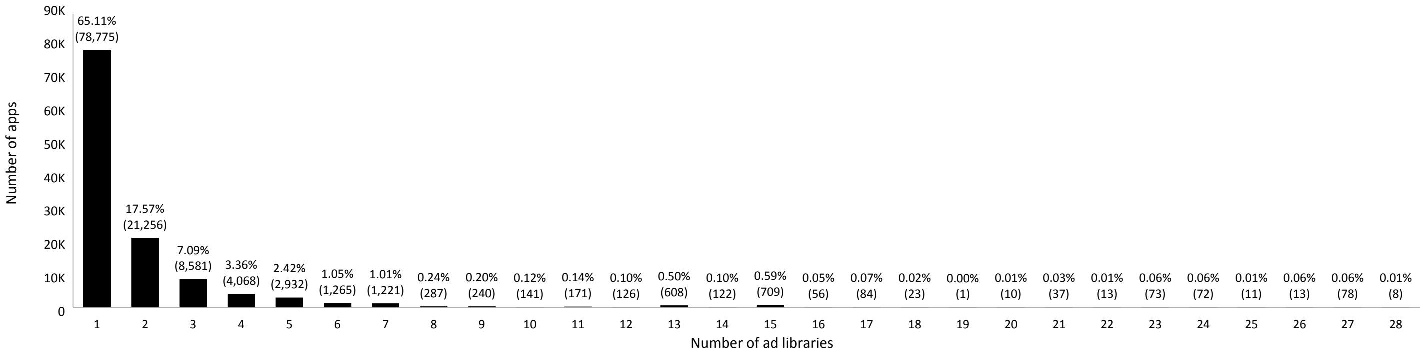

Fig. 1 Breakdown of the percentage of apps with ad libraries (y-axis) based on the number of ad libraries bundled in them (x-axis).

4 Is there a relationship between the number of ad libraries in an app and the rating received by the app?

Motivation: One key distinction between ad libraries and other libraries (like utility libraries or interface libraries), is that ad libraries are not required for the correct functioning of an app. They are exclusively for monetizing the app. From the previous section, we know that thousands of apps integrate more than one ad library. Integrating more ad libraries in an app increases the maintenance (as does adding any other piece of code) effort for the developers. Thus, the quality of the app can be impacted by the increased number of ad libraries. Consequently, we examine the relationship between the number of ad libraries in an app and the rating of an app. We use the rating of an app as an indicator of the quality of an app, since in past research, ratings have been shown to be highly correlated to the number of downloads of an app [10], which is a concrete measure of success.

Approach: In Google Play, app users can rate an app from 1 to 5 "stars" (5 is the highest value). Since the rating of an app is subject to rater bias [4], the number of raters is an important factor to consider. Hence, in order to minimize this rating bias, we decided to limit our analysis to ad-supported app versions with at least 10 raters. We also chose to study apps with at least two versions in 2011, since they represent apps that are actively maintained. Our study criteria of apps with at least one ad library, having more than one version, and at least 10 raters per version, results in 13,983 versions distributed across 5,937 different apps.

Finally, given that some apps have more versions than others, if we use every version of an app our results can be affected by apps with a high number of versions. Since each app should have the same importance, we use only one version per app. In our case, we decided to use the last version of each app. We present the results of the rating of an app based on the number of ad libraries within each app.

Results: The Spearman rank correlation between the number of ad libraries in an app and the rating of the app is 0.016. Such a weak correlation illustrates that there is no relationship between the number of ad libraries in an app and the rating of the app. In Fig. 2 we break down the data and show more details by means of a bean plot of the rating of apps grouped by the number of integrated ad libraries. Bean plots show both the median value for the rating (indicated by the solid horizontal line), and the actual distribution (the width of the curve at any point in each bean plot indicates the number of apps with a particular rating). The bean plots are ordered by the number of ad libraries that each group of apps has (X-axis). The Y-axis indicates the rating of an app. We can see that the median rating for all apps taken together is more than 4 stars. Apps with a higher number of ad libraries (the bean plots at the end) tend to have only a slightly lower median rating, except for apps having 23, 27, and 28 ad libraries, which have a rating higher than the median. However, note that these groups represent only a small percentage of all apps. Most app versions with between seven and 21 ad libraries have a rating below the median.

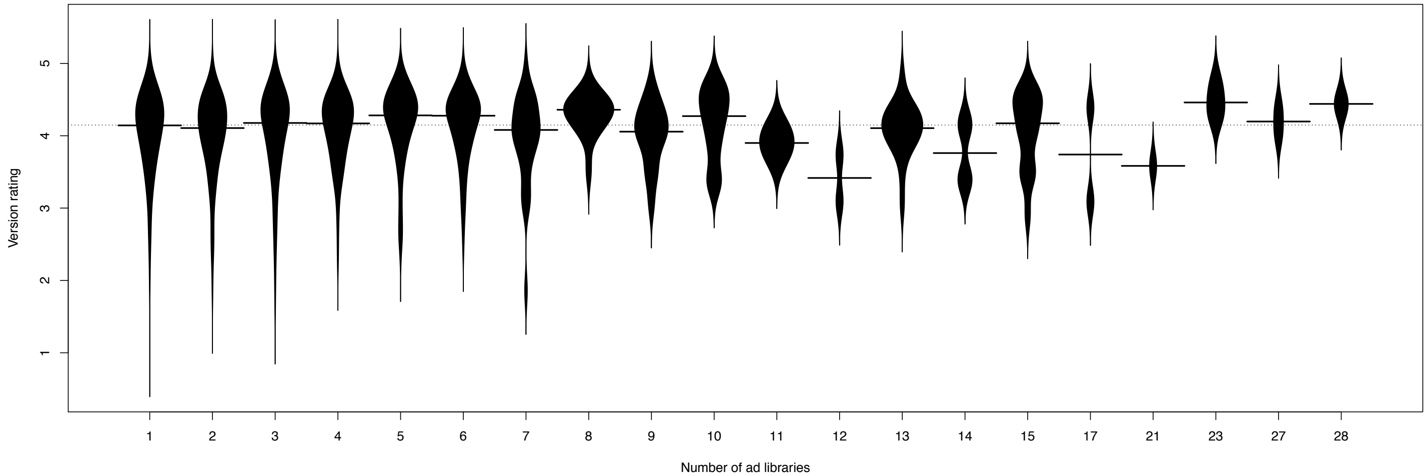

Fig. 2 Beanplots showing the relation between the number of ad libraries (*x*-axis) and the corresponding version rating (*y*-axis). Beanplots are sorted descending by the number of ad libraries. The horizontal lines represent the median of the app version rating.

Next, we examine the eight apps that have 28 ad libraries. Five out of these eight apps are developed by the same app developer, and the other three by two different app developers. Three of the apps were removed from Google Play for undisclosed reasons. We verified the rating and rater numbers for the remaining five apps in order to verify if the users of these apps are unhappy because of the high number of ad libraries. Surprisingly, those apps still have similar ratings (a high average rating), and continue accumulating positive ratings. For example, the app with the largest number of raters of these five apps has 1,489 raters, and an average rating of 4.3 out of 5, with 946 users giving a 5-star rating. Interestingly, the number of users keeps increasing in spite of having a large number of ad libraries. Note that these apps specialize in displaying pictures, and are classified as "mature only for 18+ viewing".

One possible explanation for our finding that the number of ad libraries does not impact the perceived quality of an app, is that in practice a high number of ad libraries does not mean that an app displays more than one ad at the same time. Instead the use of a large number of ad libraries is simply a monetization strategy that is used to to achieve a high fill rate.

The number of ad libraries does not affect the rating of an app.

5 Is there a relationship between a specific ad library and the ratings?

Motivation: A key goal in advertising is to attract the attention of a user. As such, ads are often part of the Graphical User Interface (GUI) of an app. Hence, app developers also have to consider how app users react to the displayed ads in order to keep their app users happy (otherwise app users will rate their apps poorly). App developers might have to be cautious about the specific ad libraries that they integrate in their apps. Thus, we now study the relationship between particular ad libraries and the rating of apps.

Approach: We use the same dataset as in the previous question. We query for the rating of each app in its last version in the ad supported multi app version dataset grouped by the type of ad libraries that it uses. For example, given a 3-star app X that integrates $ad1$ and $ad2$ ad libraries, a 4-star app Y that integrates $ad1$ and $ad3$ ad libraries, and a 3.5-star app Z that integrates $ad1$, $ad2$ and $ad3$ ad libraries. Then, the ratings for each ad library are $ad1$={3.0, 4.0, 3.5}, $ad2$={3.0, 3.5}, and $ad3$={4.0, 3.5}. We show the beanplots of the ratings for each of the ad library in our dataset.

Results: Fig. 3 presents beanplots for 70 out of the 72 ad libraries in our studied dataset (the *adhubs* and *mobus* ad libraries are not integrated in the last version of any of the studied apps). The beanplots present the rating for all the apps that integrate each specific ad library, sorted in descending order by the median rating. The long dotted line and the line across each beanplot represent the median rating of all the apps and the median rating of the apps that integrate a particular ad library, respectively. The beanplots show a slight decrease depending on the type of ad libraries. Most of the set of apps with a specific ad library are higher or slightly lower than the overall median rating (4.15).

We examine three ad libraries from Fig. 3 that have apps with lower ratings in order to find possible causes of their low rating. *Wooboo* [3] is an ad network based in China. This company also develops its own apps. We found that past research has flagged the Wooboo ad library as spyware [7][9]. One app user complains about an app with this ad library is: "*This is a direct copy of another app, it displays ads and now your password belongs to someone in china*".

Leadbolt [4] is an ad network based in Australia. This ad library exhibits an intrusive behaviour called "push notification", where the ad library pushes ads into the notification bar. This results in ads being displayed even when the app is not running [12]. In 2012, Leadbolt added a new type of ad called "app icon". An "app icon" ad installs new icons on the Android device even without the app user's authorization. Additionally, app stores have problems with this ad library because it sends the device ID in plain text [7].

Airpush [5] is an ad network based in the USA. Similar to *leadbolt*, this ad library uses a 'push notification' and the 'app icon' to serve advertisements. This ad library has become quite controversial among app users for its intrusive behaviour [11].

We observe that these apps with a highly intrusive behaviour result in complaints from app users, and in some occasions ruin the app users' perceived quality of the app. We find comments such as: "*Good game, bad ads. I was loving the game until I noticed it put a new shortcut called 'Apps' on my launcher. Sorry, but if your idea of advertising is putting sh*t in launcher pages or notifications then I'm not interested. Keep ads INSIDE the app.*". "*Good game, bad ads. This game is awesome but the ads...embarrising (sic). the ads totally ruin it and there isn't a way to take them off :(*".

[3] Wooboo:`http://www.wooboo.com.cn/`
[4] Leadbolt:`http://www.leadbolt.com/`
[5] Airpush:`http://www.airpush.com/`

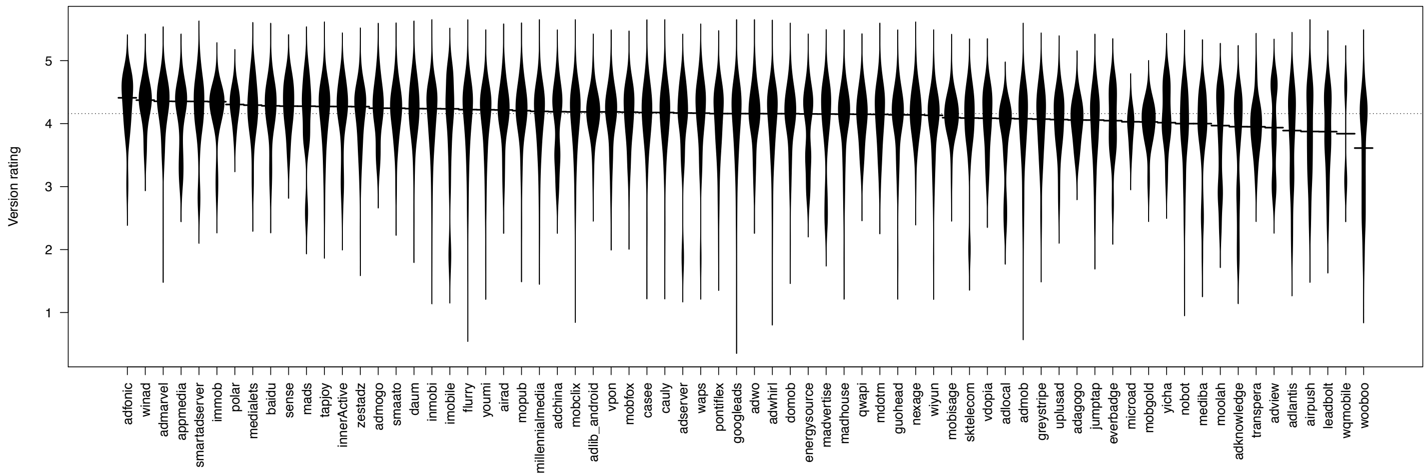

Fig. 3 Beanplots showing the distribution of the rating of the apps that contain the ad library (*y*-axis) for each of the ad libraries (*x*-axis). The beanplots are sorted descending by the median app rating. The horizontal line represents the median of the app ratings.

Ad libraries with security issues and/or intrusive behaviour may result in apps getting low ratings, disapproval from the app users [5], and even rejection from the different app stores. Recently, Google Play updated its policies to encourage the app developers to be more conscious with ad libraries that have highly intrusive behaviour. App developers need to perform tasks of ad maintenance in order to align their apps with the new policies in Google Play [2].

The behaviour of specific ad libraries negatively impacts the rating of an app.

6 Discussion

Our case study shows that app developers integrate many ad libraries in their apps, some of which can have a negative impact on the rating of the app. The negative impact does not seem to be related to the actual number of ad libraries in an app. Indeed, when considering possible motivation of app developers for adding certain ad libraries that could potentially hurt the ratings of their app, it is very obvious that although such libraries are very intrusive, they also provide the largest amount of payout for each ad clicked/viewed. For example, we found that ad libraries like Airpush promise to offer a larger amount of payout than Admob and other conventional ad networks [1]. Hence, for developers it becomes important to perform a thorough cost-benefit-analysis: 'Whether the increased revenue per click is worth the possible low rating (and possibly low future app sales)?'. In such a case, it might make sense to integrate the ad libraries discussed in Section 5. Otherwise, app developers should avoid those libraries.

7 Conclusions and Recommendations

Advertisements are a popular way to monetize free mobile apps. However, one problem with this business model is that the demand for ads is much larger than the supply, and hence apps often do not get an ad every time they request one. This (low fill rate) phenomenon leads to lower revenue generated from ads. Hence, developers tend to connect to more than one advertising company through their respective ad libraries. In this paper, we empirically examine the prevalence of apps using ad libraries (72 in our study) from multiple ad companies by examining 120,981 different apps with ad libraries. We find that almost a third of these apps have more than one ad library (and more than half have at least one ad library). We find that there is no relationship between the number of integrated ad libraries and the rating of an app. However, we find that certain specific ad libraries can in fact result in poor app ratings.

In summary, given a certain real estate space that the ads will occupy on the screen of a device (control variable), app developers can add as many ad libraries as needed to increase the fill rate without impacting their ratings. However, developers need to be careful and selective about the specific ad libraries that they integrate.

References

1. Airpush FAQ: How is Airpush different from admob and the other conventional mobile ad networks? (2014). URL `http://www.airpush.com/resources/faqs/`
2. Google play developer program policies (2014). URL `https://play.google.com/about/developer-content-policy.html`
3. Candeias, H.: Smaato releases q2 2011 mobile metrics report (2011). URL `http://www.smaato.com/metricsq22011-2/`
4. Chen, M., Singh, J.P.: Computing and using reputations for internet ratings. In: Proceedings of the 3rd ACM conference on Electronic Commerce, EC '01, pp. 154–162 (2001)
5. Chien, E.: Madware makers mixing up their code (2012). URL `http://www.mobilesecurity.com/articles/285-madware-makers-mixing-up-their-code`
6. Dienst, S., Berger, T.: Static analysis of app dependencies in android bytecode (2012). Tech. Note, available at `http://www.informatik.uni-leipzig.de/~berger/tr/2012-dienst.pdf`
7. Enck, W., Octeau, D., McDaniel, P., Chaudhuri, S.: A study of android application security. In: Proceedings of the 20th USENIX Conference on Security, SEC'11 (2011)
8. Godfrey, M., German, D.: The past, present, and future of software evolution. In: Frontiers of Software Maintenance, 2008. FoSM 2008., pp. 129–138 (2008)
9. Grace, M.C., Zhou, W., Jiang, X., Sadeghi, A.R.: Unsafe exposure analysis of mobile in-app advertisements. In: Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WISEC '12, pp. 101–112 (2012)
10. Harman, M., Jia, Y., Zhang, Y.: App store mining and analysis: Msr for app stores. In: Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on, MSR'12 (2012)
11. Ionescu, C.: Airpush begins obfuscating ad modules (2012). URL `http://www.symantec.com/connect/blogs/airpush-begins-obfuscating-ad-modules`
12. Marketing: Earn more with push notifications (2011). URL `http://www.leadbolt.com/blog/2011/10/earn-more-with-push-notifications/`
13. Mojica, I.J., Adams, B., Nagappan, M., Dienst, S., Berger, T., Hassan, A.E.: A large-scale empirical study on software reuse in mobile apps. IEEE Software **31**(2), 78–86 (2014)
14. Platzer, E.: Opportunities of automated motive-based user review analysis in the context of mobile app acceptance. Proc. CECIIS pp. 309–316 (2011)

Israel Mojica is a software engineer at McAfee. His research interests include mobile software and empirical software analysis in general. Mojica received an MS in computer science from Queens University, Canada. Contact him at Israel_Mojica@McAfee.com.

Meiyappan Nagappan is a postdoctoral fellow in the Software Analysis and Intelligence Lab (SAIL) at Queens University, Canada. His research interests include deriving solutions that encompass all the various stakeholders of software systems and using large-scale software engineering data to also address the concerns of soft- ware operators, build engineers, and project managers. Nagappan received a PhD in computer science from North Carolina State University. He received a best paper award at the International Working Conference on Mining Software Repositories (MSR 12). Contact him at mei@cs.queensu.ca.

Bram Adams is an assistant professor at the Ecole Polytechnique de Montreal, where he heads the MCIS (Maintenance, Construction, and Intelligence of Software) lab. His research interests include software release engineering, software integration, software build systems, software modularity, and software maintenance. Adams received a PhD in computer science engineering from Ghent University. He was an organizer of the First International Workshop on Release Engineering (RELENG 13) and is a member of IEEE. Contact him at bram.adams@polymtl.ca.

Thorsten Berger is a postdoctoral fellow in the Generative Software Develop- ment Lab at the University of Waterloo, Canada. His research interests include model-driven development, variability modeling for software product lines and software ecosystems, variability-aware static analyses of source code, and mining software repositories. Berger received a PhD (Dr. rer. nat.) in computer science from the University of Leipzig. Contact him at tberger@gsd.uwaterloo.ca.

Steffen Dienst is a PhD student in the Chair of Business Information Systems at the University of Leipzig, Germany. His main research topic is using machine- learning techniques to help monitor the operation of renewable power plants. Other interests range from reverse engineering to functional programming. Dienst received a M.Sc. (Dipl.-Inf.) in computer science from the University of Leipzig. Contact him at sdienst@informatik.uni-leipzig.de

Ahmed E. Hassan is the NSERC/Black- Berry Software Engineering Chair at the School of Computing at Queens University, Canada. His research interests include mining software repositories, empirical software engineering, load testing, and log mining. Hassan received a PhD in computer science from the University of Waterloo. He spearheaded the creation of the Mining Software Repositories (MSR) conference and its research community. Hassan also serves on the editorial boards of IEEE Trans- actions on Software Engineering, Springer Journal of Empirical Software Engineering, and Springer Journal of Computing. Contact him at ahmed@cs.queensu.ca.