On the Impact of API Change- and Fault-Proneness on the User Ratings of Android Apps —Additional Analyses—

Gabriele Bavota, Mario Linares-Vásquez, Carlos Bernal-Cárdenas,
Massimiliano Di Penta, Rocco Oliveto, Denys Poshyvanyk

Abstract—This document reports additional analyses made in the context of "Study I: Mining Software Repositories", presented in Section 2 of the paper: "On the Impact of API Change- and Fault-Proneness on the User Ratings of Android Apps". Indeed, as in all empirical studies different ways of analyzing the results might lead to different findings.

1 Using Different Thresholds to Classify Apps on the basis of their Rating

In our paper we grouped the 5,848 apps object of our study in three different sets on the basis of their average user rating (r_a). In particular, given $Q_1 = 3.667$ and $Q_3 = 4.395$ the first and the third quartile of the distribution of the average user ratings assigned to the 5,848 apps, we clustered the apps into the following three sets:

1) Apps having *high rating*: apps having $r_a > Q_3$.
2) Apps having *medium rating*: apps having $Q_3 \geq r_a > Q_1$.
3) Apps having *low rating*: apps having $r_a \leq Q_1$.

This choice may have had an impact on our findings. In this document, we report the results of the same analysis performed in our paper when clustering the apps differently, and in particular:

1) Apps having *high rating*: top 33% of apps as indicated by their average rating r_a. In particular, those are the apps having $r_a > 4.300$.
2) Apps having *medium rating*: middle 34% of apps as indicated by their average rating r_a. In particular, those are the apps having $4.300 \geq r_a > 3.833$.
3) Apps having *low rating*: bottom 33% of apps as indicated by their average rating r_a. In particular, those are the apps having $r_a \leq 3.833$.

Note that this classification is different from the one used in our paper, where we considered (i) the top 25% apps as those having *high rating*, (ii) the middle 50% apps as those having *medium rating*, and (iii) the bottom 25% apps as those having *low rating*.

Also, in our paper we analyzed in isolation the 50 most and the 50 least successful apps (in terms of achieved average user rating). One may wonder what happens if we enlarge such sets of 50 apps each. For this reason, in this document we report results for the most 100 and the least 100 successful apps.

1.1 RQ$_1$ Results – Does the fault-proneness of APIs affect the rating of Android Apps?

Boxplots in Figure 1 show the distribution of average number of bug fixes in API classes used by apps having different levels of rating (i.e., *high*, *medium*, and *low* rating). As done in Section 2 of the paper, we set 30 as a limit for the y-axis (i.e., average number of bug fixes in API classes) for readability purposes.

The boxplots reported in Figure 1 highlight that apps having a higher rating use APIs having a lower bug-proneness. In particular, apps having a *high* rating use APIs with 6.5 bug-fixes on average. This number grows up to 9.9 (+53%) for apps having a *medium* rating and reaches 12.5 (+93%) for apps having a *low* rating. The difference in terms of APIs fault-proneness between apps having different levels of rating is very

- *G. Bavota, University of Sannio, Benevento, Italy. E-mail: gbavota@unisannio.it*
- *M. Linares-Vásquez, The College of William and Mary, Williamsburg, VA 23185, USA. E-mail: mlinarev@cs.wm.edu*
- *C. Bernal-Cárdenas, The College of William and Mary, Williamsburg, VA 23185. E-mail: cebernal@cs.wm.edu*
- *M. Di Penta, University of Sannio, Benevento, Italy. E-mail: dipenta@unisannio.it*
- *R. Oliveto, University of Molise, Pesche (IS), Italy. E-mail: rocco.oliveto@unimol.it*
- *D. Poshyvanyk, The College of William and Mary, Williamsburg, VA 23185, USA. E-mail: denys@cs.wm.edu*

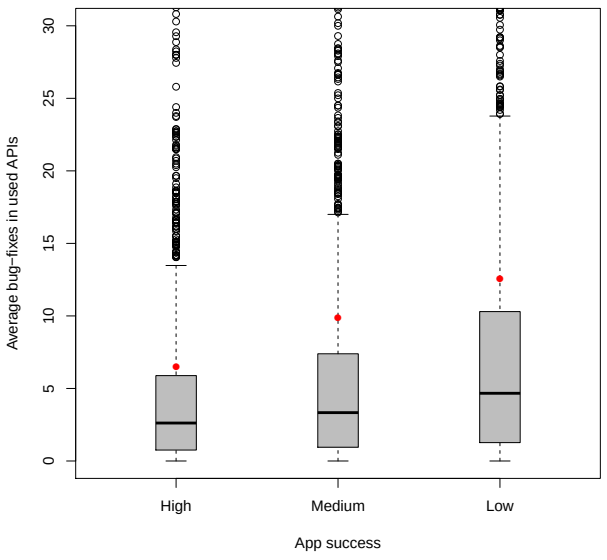

Fig. 1: Boxplots of average number of bug fixes in API classes used by apps having different levels of rating. The red dot indicates the mean.

clear by looking to the distributions depicted in Figure 1.

Table 1 reports the results of the Mann-Whitney test (p-value) and the Cliff's d effect size. As in Section 2 of the paper, we compared each set of apps (grouped by level of rating) with all other sets having a lower rating (e.g., *high rating vs.* the other). As it can be seen from the table, apps having a higher rating than others always exhibit a statistically significant lower number of bug fixes in the used APIs than apps having a lower rating (p-value always < 0.0001). Also, consistently with what observed in Section 2 of the paper, the Cliff's d is small (0.10) when comparing apps having a *high* rating and apps having a *medium* rating, and medium (0.37) when the comparison is performed between apps having a *high* rating and apps having a *low* rating.

When comparing the difference in terms of API bugs between the 100 most and the 100 least successful apps, the average number of bug fixes in the APIs used by the 100 most successful is 5.7, while for the 100 least successful we measured an average of 22.0 bug fixes in the used APIs (+290%). The Mann-Whitney test reports a statistically significant difference (p-value < 0.0001) with a large effect size (0.59).

Thus, despite the different thresholds used to classify the apps, the answer to our $\mathbf{RQ}_1$ does not change: *APIs used by apps having high users ratings are, on average, significantly less fault-prone than APIs used by low rated apps.*

TABLE 1: Use of fault-prone APIs by apps having different levels of rating: Mann-Whitney test (adj. p-value) and Cliff's Delta (d).

Test	adj. p-value	d
high rating *vs* medium rating	<0.0001	0.12 (Small)
high rating *vs* low rating	<0.0001	0.33 (Medium)
medium rating *vs* low rating	<0.0001	0.13 (Small)

1.2 $\mathbf{RQ}_2$ Results – Does the change-proneness of APIs affect the rating of Android Apps?

Boxplots in Figure 2 show the change-proneness of APIs used by the three different sets of apps. In particular, Figures 2-(a) and 2-(b) report the overall number of method changes and the overall number of changes in the method signatures, respectively, while Figures 2-(c) and 2-(d) show the same data by considering the APIs' public methods only.

As already observed in our paper, Figure 2 suggests that apps having a higher rating generally use more stable APIs, i.e., APIs having a lower change-proneness. In particular, the APIs used by apps having a *high* rating underwent, on average, 26 method changes, as opposed to the 37 changes in the APIs used by apps having a *medium* rating (+42%) and to the 46 (+77%) of the apps having a *low* rating—see Figure 2-(a). Also, the three quartiles indicate a continuous upward-trend of the number of changes as the app success decreases.

The trend is almost the same if considering public methods only: an average of 16 method changes for APIs used by top apps having a *high* rating, 22 for those having a *medium* rating (+38%), and 26 for APIs used by apps having a *low* rating (+63%)—Figure 2-(c). Again, boxplots confirm that apps having a *low* rating generally use more change-prone APIs as compared to apps having a *high* rating. Also for changes involving method signatures (Figure 2-(b,d)), results highlight that highly rated apps are generally built using stable APIs.

Table 2 reports the results of the Mann-Whitney test and the Cliff's d when comparing the change-proneness of APIs used by apps belonging to different groups of average user ratings. Table 2 shows that: (i) there is statistically significant difference (p-value $<$ 0.0001) when comparing apps having a higher rating with those having a lower one, and (ii) Cliff's delta is small for all comparison.

When comparing the top 100 and the least 100 successful apps (i) the p-value is confirmed < 0.0001, and (ii) we get a *large* Cliff's d (≥ 0.474) for all change types.

Thus, also for our $\mathbf{RQ}_2$, the main finding reported in Section 2 of the paper is confirmed: *APIs used by apps having high user ratings are, on average, less prone to changes occurred to API signatures and implementation than APIs used by low rated apps.*

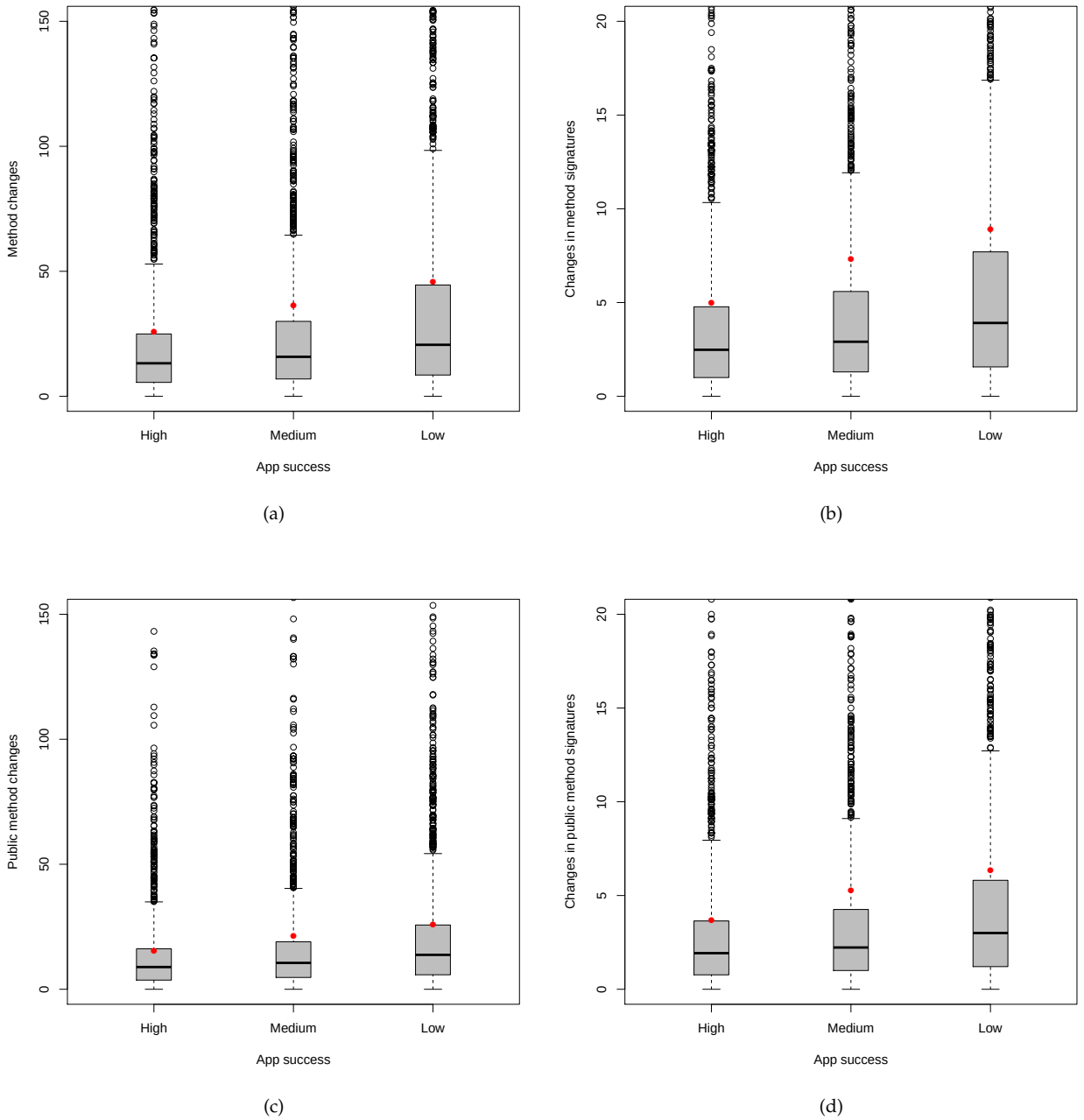

Fig. 2: Boxplots of change-proneness in API classes used by apps having different levels of rating. The red dot indicates the mean.

2 Focusing the Analysis on the 1,000 Most Popular Apps

The set of 5,858 apps considered in our study has been randomly selected from the Google Play Market. Thus, it includes very popular apps (e.g., the CNN official app) as well as apps that are not so popular. It is interesting to verify if the findings reported in Section 2 of the paper about the relationship between change and fault-proneness of used APIs and app's rating are still valid when just considering very popular apps on the Google Play Market since those are the ones users expect to exhibit a higher quality. A good proxy to estimate the apps' popularity would be the number of times an app has been downloaded. However, the Google Play Market does not provide such an information (none of the mobile markets lists the number of downloads). Indeed, it just shows the number of times an app has been installed in ranges (e.g., from 100.000 to 500.000). Such information is not enough precise for our purposes. For this reason, we used the number of reviews received by an app

TABLE 2: Change-proneness of APIs for apps having different levels of rating: Mann-Whitney test (p-value) and Cliff's delta (d).

Test	adj. p-value	d
Overall Method Changes		
high rating *vs* medium rating	<0.0001	0.10 (Small)
high rating *vs* low rating	<0.0001	0.23 (Small)
medium rating *vs* low rating	<0.0001	0.13 (Small)
Changes to Public Methods		
high rating *vs* medium rating	<0.0001	0.10 (Small)
high rating *vs* low rating	<0.0001	0.23 (Small)
medium rating *vs* low rating	<0.0001	0.13 (Small)
Overall Changes in Method Signatures		
high rating *vs* medium rating	<0.0001	0.09 (Small)
high rating *vs* low rating	<0.0001	0.22 (Small)
medium rating *vs* low rating	<0.0001	0.13 (Small)
Changes in Public Method Signatures		
high rating *vs* medium rating	<0.0001	0.09 (Small)
high rating *vs* low rating	<0.0001	0.22 (Small)
medium rating *vs* low rating	<0.0001	0.13 (Small)

TABLE 3: Use of fault-prone APIs by apps having different levels of rating: Mann-Whitney test (adj. p-value) and Cliff's Delta (d). **Most 1,000 popular apps.**

Test	adj. p-value	d
high rating *vs* medium rating	0.008	0.11 (Small)
high rating *vs* low rating	0.012	0.12 (Small)
medium rating *vs* low rating	0.400	0.01 (Small)

as a proxy of its popularity. Our conjecture is that the higher the popularity of an app, the higher the number of times it will be downloaded, the higher the number of reviews it will receive.

We replicated our analyses just on the most 1,000 popular apps (i.e., those having received at least 1,237 reviews). We grouped the 1,000 apps in three different sets on the basis of their average user rating (r_a). In particular, given $Q_1 = 4.034$ and $Q_3 = 4.475$ the first and the third quartile of the distribution of the average user ratings assigned to the most 1,000 popular apps, we clustered the apps into the following three sets:

1) Apps having *high rating*: apps having $r_a > Q_3$.
2) Apps having *medium rating*: apps having $Q_3 \geq r_a > Q_1$.
3) Apps having *low rating*: apps having $r_a \leq Q_1$.

2.1 RQ_1 Results – Does the fault-proneness of APIs affect the rating of Android Apps?

Boxplots in Figure 3 show the distribution of average number of bug fixes in API classes used by apps having different levels of success (i.e., *high*, *medium*, and *low* rating). Figure 3 just considers the most 1,000 popular apps.

Also when just considering the most 1,000 popular apps, it is clear from Figure 1 that apps having a higher rating use APIs having a lower fault-proneness. In particular, apps having a *high* rating

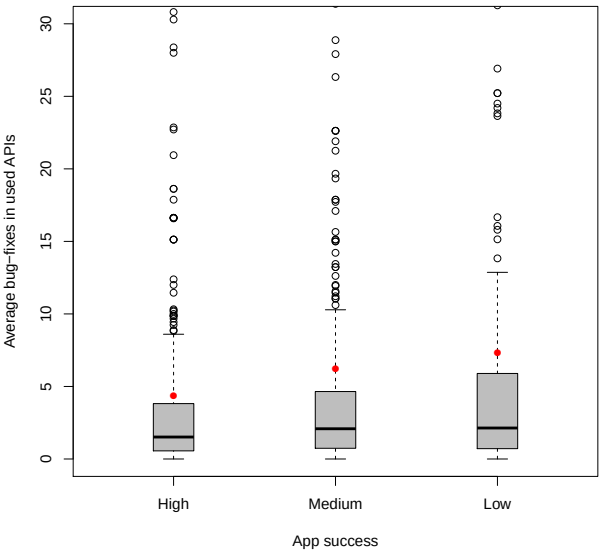

Fig. 3: Boxplots of average number of bug fixes in API classes used by apps having different levels of rating. The red dot indicates the mean. **Most 1,000 popular apps.**

use APIs with 4.4 bug-fixes on average. This number grows up to 6.2 (+41%) for apps having a *medium* rating and reaches 7.4 (+68%) for apps having a *low* rating.

Table 3 reports the results of the Mann-Whitney test (p-value) and the Cliff's d effect size. As in Section 2 of the paper, we compared each set of apps (grouped by level of rating) with all other sets having a lower rating (e.g., *high rating vs*. the other).

As we can notice from Table 3, there is a statistically significant difference between the fault-proneness of APIs used by apps having a *high rating* and the fault-proneness of APIs used by apps having a *medium* and a *low rating*. On the other side, there is no statistically significant difference when comparing *medium* and *low* rating apps in this dataset. This is likely due to the fact that the *low rating* apps in this dataset still have quite high ratings (their average rating is 3.8). This is an expected result, since the more an app receives positive reviews, the more it will be downloaded by other users, the more reviews it will receive. Thus, apps receiving a lot of reviews (i.e., popular apps) are also likely to exhibit a high average rating.

2.2 RQ_2 Results – Does the change-proneness of APIs affect the rating of Android Apps?

Boxplots in Figure 4 show the change-proneness of APIs used by the three different sets of apps considered. In particular, Figures 2-(a) and 2-(b) report the

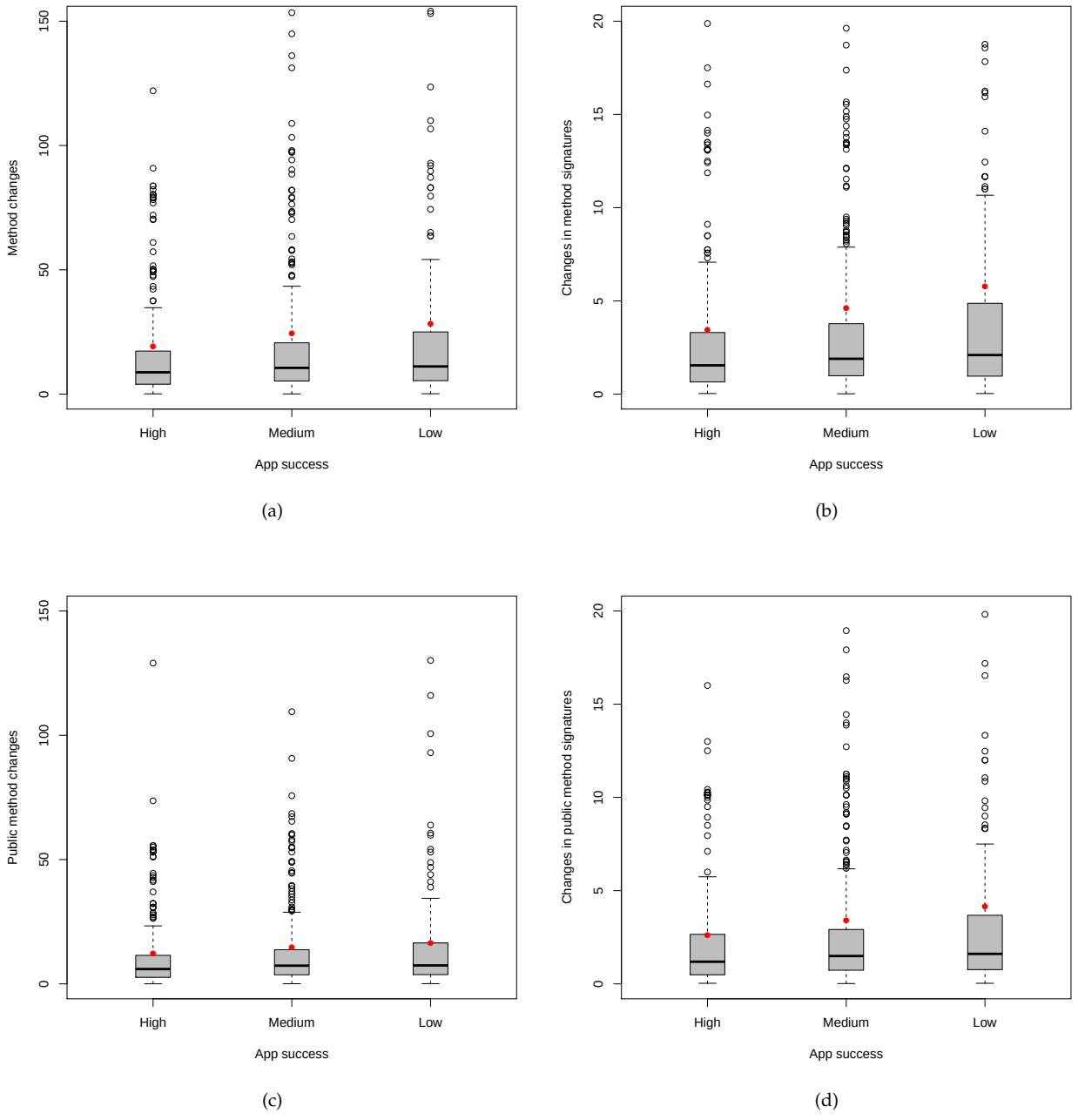

Fig. 4: Boxplots of change-proneness in API classes used by apps having different levels of rating. The red dot indicates the mean. **Most 1,000 popular apps.**

overall number of method changes and the overall number of changes in the method signatures, respectively, while Figures 2-(c) and 2-(d) report the same results, however considering the APIs' public methods only. Again, these boxplots just refer to the 1,000 most popular apps.

Also in this case, Figure 4 confirms the main finding of our paper, suggesting that apps having a higher rating generally use more stable APIs, i.e., APIs having a lower change-proneness. For instance, the APIs used by apps having a *high* rating underwent, on average, 19 method changes, as opposed to the 25 changes in the APIs used by apps having a *medium* rating (+32%) and to the 28 (+47%) of the apps having a *low* rating—see Figure 4-(a). The trend is the same when just focusing on public methods—Figure 4-(c)—as well as when considering changes involving method signatures—Figure 4-(b,d).

The results of the statistical tests are reported in Table 4 (Mann-Whitney test and the Cliff's d) when comparing the change-proneness of APIs used by apps belonging to different groups of average user

TABLE 4: Change-proneness of APIs for apps having different levels of rating: Mann-Whitney test (p-value) and Cliff's delta (d). **Most 1,000 popular apps.**

Test	adj. p-value	d
Overall Method Changes		
high rating *vs* medium rating	0.0257	0.09 (Small)
high rating *vs* low rating	0.0150	0.11 (Small)
medium rating *vs* low rating	0.2409	0.03 (Small)
Changes to Public Methods		
high rating *vs* medium rating	0.0224	0.09 (Small)
high rating *vs* low rating	0.0168	0.11 (Small)
medium rating *vs* low rating	0.2863	0.03 (Small)
Overall Changes in Method Signatures		
high rating *vs* medium rating	0.0113	0.10 (Small)
high rating *vs* low rating	0.0042	0.14 (Small)
medium rating *vs* low rating	0.1481	0.05 (Small)
Changes in Public Method Signatures		
high rating *vs* medium rating	0.0091	0.11 (Small)
high rating *vs* low rating	0.0038	0.14 (Small)
medium rating *vs* low rating	0.1561	0.05 (Small)

ratings. On the one side, Table 4 shows that there is statistically significant difference (p-value < 0.05) when comparing apps having a *high* rating with those having a *medium* and a *low* rating. On the other side, as already observed for the fault-proneness, there is no statistically significant difference when comparing apps having a *medium* and a *low* rating. As explained before, this is due to the fact that *low* successful apps among the most 1,000 popular apps still exhibit high ratings (their average rating is 3.8).