API Change and Fault Proneness: a Threat to the Success of Android Apps

Mario Linares-Vásquez, Gabriele Bavota, Carlos Bernal-Cárdenas, Massimiliano Di Penta, Rocco Oliveto, Denys Poshvyvanyk
The App Economy
2012: average monthly revenue of over $4,800 (iOS), $3,700 (BlackBerry), $3,300 (Android)

2013: 850,000 apps available (Google Play) with around 48 billion downloads

Tremendous success ...
Revenue models

Tremendous success ...

Customer loyalty

Low cost of handsets
However...
Apps are built using APIs... and....
Apps are built using APIs... and....

THERE ARE SOME ISSUES RELATED TO APIs
Useful, but usable?
Factors Affecting the Usability of APIs

Minhaz F. Zibran  Farjana Z. Eishita  Chanchal K. Roy
Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada S7N 5C9
Email: {minhaz.zibran, farjana.eishita, chanchal.roy}@usask.ca

Abstract—Software development today has been largely driven by the use of API libraries, frameworks, and reusable code. However, the API usability issues often increase the cost and lower code quality. In this paper, we present different bug reports to take into account the API usability issues. Therefore, API designers and developers need a good understanding of API usability and apply it in the design and development phases, so that they can minimize the maintenance difficulties caused by the usability issues associated with such APIs.

There are many different API usability attributes observed in the existing work of researchers and practitioners, though no one has ever been able to create a complete list of 22
Among 1513 bug reports:

Useful, but usable?
Factors Affecting the Usability of APIs

Minhaz F. Zibran       Farjana Z. Eishita       Chanchal K. Roy
Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada S7N 5C9
Email: {minhaz.zibran, farjana.eishita, chanchal.roy}@usask.ca

Abstract—Software development today has been largely driven by the use of API libraries, frameworks, and reusable code. The use of API usability issues often increase the complexity of code and lower code quality. In this paper, we present a new approach for identifying API usability issues and provide guidance to developers on how to take into account the API usability issues. Therefore, API designers and developers need a good understanding of API usability and apply it in the design and development phase, so that they can minimize the maintenance difficulties caused by the usability issues associated with such APIs.

There are many different API usability attributes observed in the literature. This paper focuses on the usability of APIs developed by researchers and practitioners, though no one has yet taken API usability into consideration. Until Zibran [21], the authors of this paper, have done the work.
Useful, but usable?
Factors Affecting the Usability of APIs

Minhaz F. Zibran
Farjana Z. Eishita
Chanchal K. Roy
Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada S7N 5C9
Email: {minhaz.zibran, farjana.eishita, chanchal.roy}@usask.ca

Abstract—Software development today has been largely depen
dent on the use of API libraries, frameworks, and reusable
designs. API usability issues often increase the complexity of APIs
and lower code quality. In this paper, we examine different bug
categories and APIs to take into account the API usability issues. Therefore, API
designers and developers need a good understanding of API
usability and apply it in the design and development phases,
so that they can minimize the maintenance difficulties caused
by the usability issues associated with such APIs.

There are many different API usability attributes observed
together with developers and practitioners, though no one
has yet developed a comprehensive list of 22

Among 1513 bug reports:

API correctness

175
Is the Android API change/fault prone?
17 releases in about 5 years...
17 releases in about 5 years ...

3269 classes
New classes
17 releases in about 5 years...
17 releases in about 5 years ...

96524 changes in methods
Changes in methods
May API instability and fault-proneness impact the success of Android applications?
Free Apps analyzed

7097
30 Domain categories from Google play
App’s success: Average users rating
App’s success: Average users rating
API fault-proneness: Number of bugs fixed in the APIs
App's success: Average users rating
API fault-proneness: Number of bugs fixed in the APIs
API change-proneness: Number of API changes at method level
### Bug-fixes in APIS

<table>
<thead>
<tr>
<th>Branch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>Reconcile with jb-mr2-zeroday-release - do not merge by The Android Open Source Project - 6 weeks ago</td>
</tr>
<tr>
<td>ics-factoryrom-2-release</td>
<td>Merge in jb-mr2-zeroday-release history after reset to jb-mr2-dev by The Android Automerger - 6 weeks ago</td>
</tr>
<tr>
<td>ics-m0</td>
<td>Fix preview thumb and touch target out of sync by Doris Liu - 6 weeks ago</td>
</tr>
<tr>
<td>ics-mr1</td>
<td>Fix preview thumb and touch target out of sync by Doris Liu - 6 weeks ago</td>
</tr>
<tr>
<td>ics-plus-aosp</td>
<td>Fix preview thumb and touch target out of sync by Doris Liu - 6 weeks ago</td>
</tr>
<tr>
<td>jb-dev</td>
<td>Reconcile with jb-mr2-zeroday-release - do not merge by The Android Open Source Project - 8 weeks ago</td>
</tr>
<tr>
<td>jb-mr0-release</td>
<td>Merge in jb-mr2-zeroday-release history after reset to jb-mr2-dev by The Android Automerger - 8 weeks ago</td>
</tr>
<tr>
<td>jb-mr1-release</td>
<td>Merge remote-tracking branch 'origin/jb-mr2-zeroday-release' into HEAD by The Android Automerger - 8 weeks ago</td>
</tr>
</tbody>
</table>

---

**Populating a Release History Database from Version Control and Bug Tracking Systems**

Michael Fischer, Martin Pinzger, and Harald Gall  
Distributed Systems Group, Vienna University of Technology  
{fischer,pinzger,gall}@infosys.tuwien.ac.at

Anticipating the future evolution of software projects. Unfortunately, current version and bug report systems provide no support for the combination of both data. We provide support for the combination of both data, which results in software evolution include the version.
Changes at method level

- Generic changes
- Method signature
- Exceptions
Bug-fixes, and changes in the used APIs

Considered version

Next version or last rating
Average App rating

2≤ra
2<ra≤3
3<ra≤4
ra>4
All the data are available at...

http://www.cs.wm.edu/semeru/data/fse-android-api/
Does the fault-proneness of APIs affect the success of Android Apps?
Average App Rating ($r_a$) vs Bug-fixes in used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Avg bug-fixes in used APIs

\[ r_a > 4 \quad 3 < r_a \leq 4 \quad 2 \leq r_a \leq 3 \quad 2 \leq r_a \]
Avg bug-fixes in used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Avg bug-fixes in used APIs

- $r_a > 4$: 7
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $r_a \leq 2$
Avg bug-fixes in used APIs

- $r_a > 4$: 7
- $3 < r_a <= 4$: 10
- $2 < r_a <= 3$
- $2 <= r_a$
Avg bug-fixes in used APIs

+42%
Avg bug-fixes in used APIs

- $r_a > 4$: 7
- $3 < r_a \leq 4$: 10
- $2 < r_a \leq 3$: 15
- $2 \leq r_a$:
Avg bug-fixes in used APIs

+119%

<table>
<thead>
<tr>
<th>Condition</th>
<th>Avg Bug-Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_a &gt; 4$</td>
<td>7</td>
</tr>
<tr>
<td>$3 &lt; r_a \leq 4$</td>
<td>10</td>
</tr>
<tr>
<td>$2 &lt; r_a \leq 3$</td>
<td>15</td>
</tr>
<tr>
<td>$2 \leq r_a$</td>
<td></td>
</tr>
</tbody>
</table>
Avg bug-fixes in used APIs

- $r_a > 4$: 7
- $3 < r_a \leq 4$: 10
- $2 < r_a \leq 3$: 15
- $2 \leq r_a$: 23
Avg bug-fixes in used APIs

+249%
50 most successful vs 50 least successful apps
50 most successful vs 50 least successful apps

+500%
APIs used by successful apps are significantly less fault-prone than APIs used by unsuccessful apps.
Does the change-proneness of APIs affect the success of Android Apps?
Overall method changes in used APIs

Average App Rating ($r_a$)

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Avg changes in used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Avg changes in used APIs

- $r_a > 4$: 27
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Avg changes in used APIs

- $r_a > 4$: 27
- $3 < r_a \leq 4$: 36
- $2 \leq r_a \leq 3$
- $2 \leq r_a$
Avg changes in used APIs

+33%

<table>
<thead>
<tr>
<th>Condition</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_a &gt; 4$</td>
<td>27</td>
</tr>
<tr>
<td>$3 &lt; r_a \leq 4$</td>
<td>36</td>
</tr>
<tr>
<td>$2 \leq r_a \leq 3$</td>
<td></td>
</tr>
<tr>
<td>$r_a \leq 2$</td>
<td></td>
</tr>
</tbody>
</table>
Avg changes in used APIs

- $r_a > 4$: 27
- $3 < r_a \leq 4$: 36
- $2 < r_a \leq 3$: 53
- $2 \leq r_a$: (bar not shown)
Avg changes in used APIs

- $r_a > 4$: 27
- $3 < r_a \leq 4$: 36
- $2 < r_a \leq 3$: 53
- $2 \leq r_a$: +96%
Avg changes in used APIs

- $r_a > 4$: 27
- $3 < r_a \leq 4$: 36
- $2 < r_a \leq 3$: 53
- $2 \leq r_a$: 78
Avg changes in used APIs

+189%
50 most successful vs 50 least successful apps
50 most successful vs 50 least successful apps

+333%
Average App Rating ($r_a$)

Changes in exceptions thrown by methods
APIs used by successful apps are less prone to changes than APIs used by unsuccessful apps.

No difference when the changes are on the exceptions thrown by API methods.
Black magic?
482 out of 812 votes were 1 star, mostly due to bug presence
482 out of 812 votes were 1 star, mostly due to bug presence

Rating: ★
A Google User - July 3, 2012
Widget?
The widget looks awesome when it doesn’t foul up. I just don’t understand the invisible widget thing. please fix
Rating: ★
A Google User - July 3, 2012
Widget?
The widget looks awesome when it doesn’t foul up. I just don’t understand the invisible widget thing. please fix

Rating: ★ ★
A Google User - July 6, 2012
Needs some MAJOR bug fixes
I was excited to see that the app has finally been updated. But some of its widgets became invisible.

482 out of 812 votes were 1 star, mostly due to bug presence
482 out of 812 votes were 1 star, mostly due to bug presence

Rating: ★
A Google User - July 3, 2012
Widget?
The widget looks awesome when it doesn’t foul up. I just don’t understand the invisible widget thing. please fix

Rating: ★★
A Google User - July 6, 2012
Needs some MAJOR bug fixes
I was excited to see that the app has finally been updated. But some of its widgets became invisible.

FIX BUG #6773607: Layered views animating from offscreen sometimes remain invisible
android.speech.tts
android.speech.tts

15 Classes
android.speech.tts

15 Classes
460 Method changes
android.speech.tts

15 Classes

460 Method changes

289 to public methods
android.speech.tts

15 Classes
460 Method changes
289 to public methods
69 bug fixes
android.speech.tts

15 Classes
460 Method changes
289 to public methods
69 bug fixes
A change each 13 days
More than 200 users complained about problems related to this feature.
Conclusion
7097 Free Apps analyzed

android.speech.tts

15 Classes
460 Method changes
289 to public methods
69 bug fixes
A change each 13 days

More than 200 users complained about problems related to this feature
API change and fault proneness represent a threat to the success of Android Apps.
Create a recommendation system able to alert the developer when some of her choices could negatively impact the success of her App.
Thank you!

Questions and/or comments

Mario Linares-Vásquez
PhD Student
The College of William and Mary
mlinarev@cs.wm.edu

Gabriele Bavota
Research Fellow
University of Sannio
gbavota@unisannio.it