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 Poshyvanyk
The App Economy
Tremendous success ...

2012: average monthly revenue of over $4,800 (iOS), $3,700 (BlackBerry), $3,300 (Android)

2013: 850,000 apps available (GooglePlay) with around 48 billion downloads
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
Breaking changes
Instability
Bugs
Backward compatibility
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 dependent on the use of API libraries, frameworks, and reusable code. However, APIs often increase the complexity of the code, leading to lower code quality. In this study, we examine 1513 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.
Among 1513 bug reports:

562 API usability issues

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}@usu.sk.ca
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}@usu.ca

Abstract—Software development today has been largely de-
ticated to the use of API libraries, frameworks, and reus-
able code. However, the API usability issues often increase
cost and lower code quality. In this paper, we present a
study on API users’ feedback and identify the factors
that affect the usability of APIs. We find that users tend
to take into account the API usability issues. Therefore, API
designers and developers need a good understanding on 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.

We observe that there are many different API usability attributes observed
in practice. However, despite the efforts of researchers and practitioners, though
no one addresses the API usability attributes observed
to date, many users have not been able to
understand and implement the API.

175 API correctness
Is the Android API change/fault prone?
17 releases in 4 years ...
17 releases in 4 years...

3269 classes
New classes
17 releases in 4 years ...
17 releases in 4 years ...

96524 changes in methods
Changes in methods
May API instability and fault-proneness impact the success of Android applications?
7097 Free Apps analyzed
Domain categories from Google play

30
Categories and Apps
Categories and Apps

- Productivity: 161
- Communication: 168
- Business: 169
- Finance: 196
- Sports: 226
- News-and-magazine: 227
- Photography: 236
- Cards: 254
- Media-and-video: 268
- Racing: 272
- Sports-games: 279
- Music-and-audio: 281
- Education: 365
- Arcade: 386
- Casual: 389
- Brain: 396
- Lifestyle: 420
- Tools: 538
- Personalization: 542
- Entertainment: 694
App’s success: Average users rating
App’s success: Average users rating
API fault-proneness: Number of bugs fixed in the Android API
<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>App’s success:</td>
<td>Average users rating</td>
</tr>
<tr>
<td>API fault-proneness:</td>
<td>Number of bugs fixed in the Android API</td>
</tr>
<tr>
<td>API change-proneness:</td>
<td>Number of changes at method level in the Android API</td>
</tr>
</tbody>
</table>
Bug-fixes, and changes in the used APIs

Considered version

Next version or last rating

android.app.ActionBar

android.gesture.Gesture
Bug-fixes in APIS
Changes at method level

- Generic changes
- Method signature
- Exceptions
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 ($r_a$)

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$

Bug-fixes in used APIs

Chart showing the distribution of bug-fixes in used APIs across different average app rating categories.
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

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

- $r_a > 4$: 7
- $3 < r_a \leq 4$: 10
- $2 \leq r_a \leq 3$: 2
- $r_a \leq 2$: 0

Graph showing the average number of bug-fixes in different ranges of $r_a$. The ranges are: $r_a > 4$, $3 < r_a \leq 4$, $2 \leq r_a \leq 3$, and $r_a \leq 2$. The bars represent the average number of bug-fixes for each range.
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
- $2 \leq r_a$: 0
Avg bug-fixes in used APIs

+119%
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?
Average App Rating ($r_a$)

Overall method changes in used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$

The box plots show the distribution of overall method changes in used APIs for different rating ranges.
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
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
- $3 < r_a <= 4$: 36
- $2 < r_a <= 3$
- $2 <= r_a$
Avg changes in used APIs

+33%

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

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

+96%

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</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>53</td>
</tr>
<tr>
<td>$r_a \leq 2$</td>
<td></td>
</tr>
</tbody>
</table>
Avg changes in used APIs

<table>
<thead>
<tr>
<th>Condition</th>
<th>Avg Changes</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>$2 &lt; r_a \leq 3$</td>
<td>53</td>
</tr>
<tr>
<td>$2 \leq r_a$</td>
<td>78</td>
</tr>
</tbody>
</table>
Avg changes in used APIs

+189%

<table>
<thead>
<tr>
<th>Condition</th>
<th>AVG Changes</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>$2 &lt; r_a \leq 3$</td>
<td>53</td>
</tr>
<tr>
<td>$2 \leq r_a$</td>
<td>78</td>
</tr>
</tbody>
</table>
50 most successful vs 50 least successful apps
50 most successful vs 50 least successful apps

+333%
Changes in exceptions thrown by methods

Average App Rating ($r_a$)
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
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.
Cable News Network (CNN) App for Android Tablet
CNN - July 29, 2013
News & Magazines

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
android.speech.tts

More than 200 users complained about problems related to this feature.

15 Classes
460 Method changes
289 to public methods
69 bug fixes
A change each 13 days
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.
API change and fault proneness represent a threat to the success of Android Apps.

Not all APIs are the same...

- Overall method changes in used APIs:
  - $r > 4$
  - $3 < r <= 4$
  - $2 < r <= 3$
  - $r <= 2$

- Average App Rating ($r_a$):

- Bug fixes in used APIs:
  - $<= r_a$

- 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
Long-term Goal
Long-term Goal

Create a recommendation system able to suggest the developers which API to use given specific needs
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
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

$r_a > 4$
$3 < r_a <= 4$
$2 < r_a <= 3$
$2 <= r_a$
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

Avg. Number of Changes in Used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

Avg. Number of Changes in Used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

Avg. Number of Changes in Used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

Avg. Number of Changes in Used APIs

- $r_a > 4$
- $3 < r_a <= 4$
- $2 < r_a <= 3$
- $2 <= r_a$
Same trend for other changes

- All Changes
- All (Public)
- Signature
- Signature (Public)

Avg. Number of Changes in Used APIs

- $r_a > 4$
- $3 < r_a \leq 4$
- $2 < r_a \leq 3$
- $2 \leq r_a$
does not work

black screen

not compatible with

malfunction

not work on my

waste of time

cant even open it

boring

freezes

update

waste of time

uninstalled

version

worth

horrible

uninstall

pay

useless

sucks

please fix it

this is the worst

terrible

annoying

needs update

lacks

would be great if

money

crap

force close

buy

update
does not work
update
force close
crashes
uninstall
useless
pay
money
waste of time

please fix it
sucks
terrible
annoying
this is the worst

would be great if
needs update
lacks

worth
version
horrible
uninstalled
boring
freezes

black screen

not compatible with

not work on my
malfunction

cant even open it

buy

waste of time
Ratings as an indicator of Apps success
Ratings as an indicator of Apps success

Bug-Fixes as indicator of fault-proneness
Ratings as an indicator of Apps success

Bug-Fixes as indicator of fault-proneness

Only Android APIs considered
Ratings as an indicator of Apps success

Bug-Fixes as indicator of fault-proneness

Only Android APIs considered

Only free apps analyzed