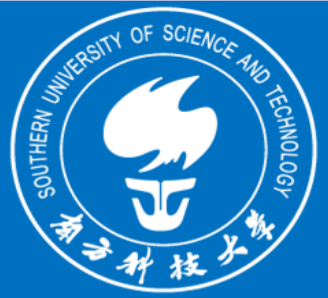


Collaborative bug finding and bug-fixing for Android Apps

Shin Hwei Tan

Southern University of Science and Technology

Accepted in ICSE2020 with Ziqiang Li

How do Developers Test Android Applications?

"I have not found any easy-to-use testing solutions for Android"

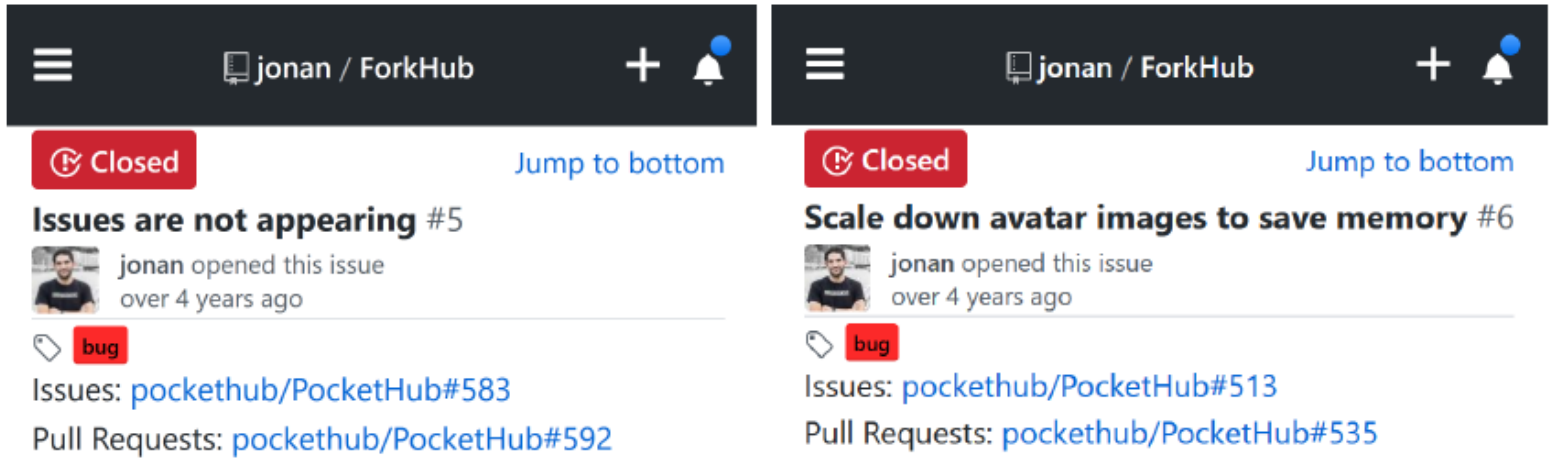
"it's hard to write the useful test case for current project, because the requirement is changed very often, and the schedule is very tiny. So we still prefer hire some tester to do manually testing. In the other hand, the android test framework is not good enough yet, I tried study roboeletric, it's a little bit hard to understand."

"A lot of what I do is related to how the app looks and feels. Therefore a lot of my testing is done manually...

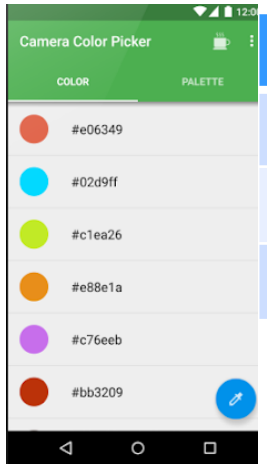
- Developers mostly rely on manual testing and unit testing.
- Developers prefer automatically generated test cases in natural language

from "*How do Developers Test Android Applications?*" [ICSME'17]

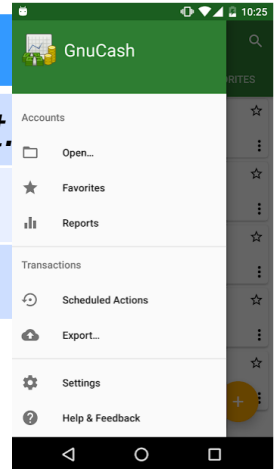
Many bug reports exist in repositories like GitHub



App developers who developed PocketHub and ForkHub found similar bugs across two apps



Bug report's title in CameraColorPicker	Bug report's title in Gnucash
How do I get left top color.	*When an account is edited, its color is lost.*
publish to F-Droid	*Publishing on F-Droid*
Make gradlew executable	*Make gradlew executable*



There exists one-to-one correspondence for the bug report's title in apps of different categories!

Crafting test scenario from bug report

Closed

Jump to bottom

Number of notes in a category not update immediately

#Anon

Student-A opened this issue 9 months ago • edited 9 months ago

Describe the bug

The number of notes in a category does not update immediately.

Context

- Device: Nokia 7 plus
- OS version: Android 8.1.0
- App version: 5.2.2

How to reproduce

Steps to reproduce the behavior:

1. Go to a note
2. Click on category button on the top
3. Click "ADD CATEGORY"
4. Click on category button on the top again
5. The number of notes in a category does not update.

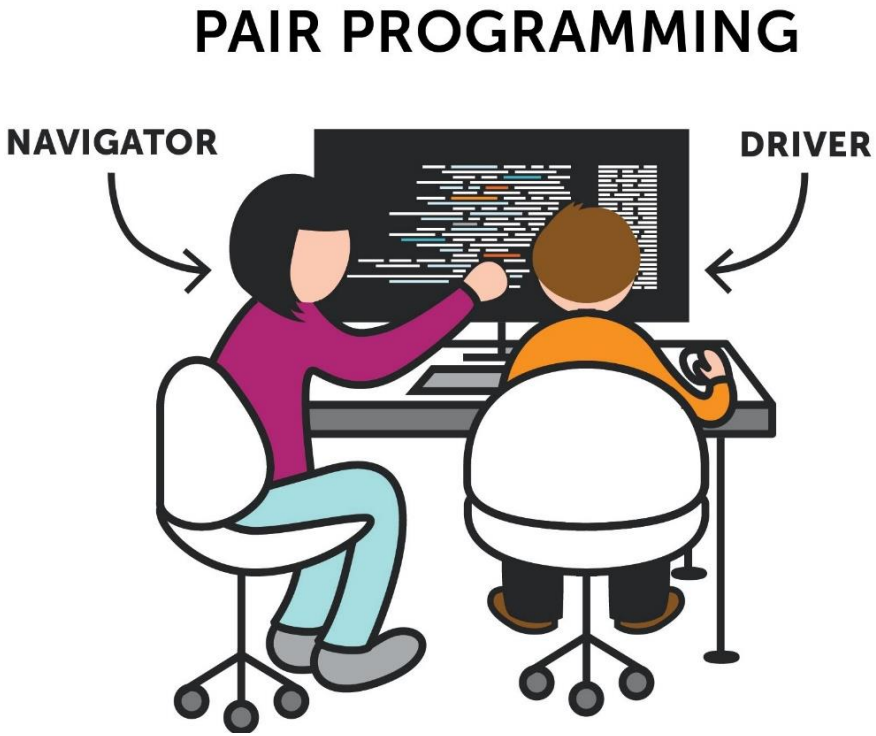
Expected behavior

The number of notes in a category should update immediately.
In this case, It should be plus one.
If I quit the note, then enter the note again. It is updated.

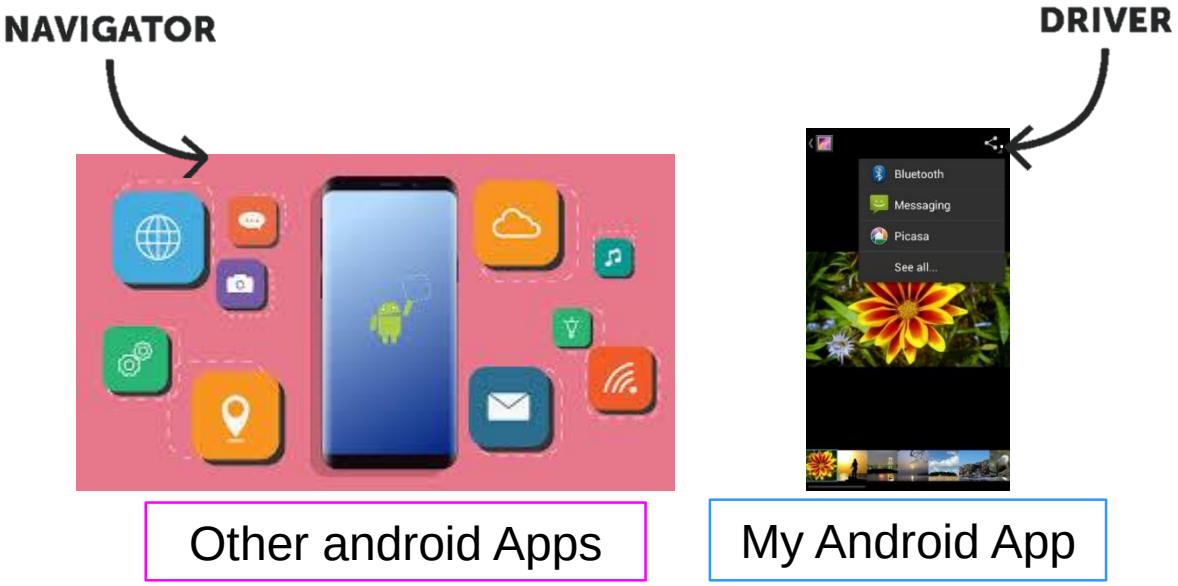
- A test scenario includes:
  - steps to reproduce
  - test data (e.g., an image for image processing app)
  - the expected behavior
    - Could solve the test oracle problem

How does collaborative bug finding works?

- Emulate the role of a competent pair programmer via developers of other similar applications



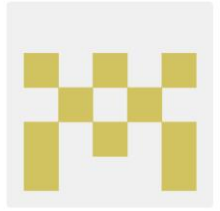
- Designed 3 settings to model different interactions between coders
  - Coders-vs-Coders
  - Coder-vs-Manual-Issues
  - Coder-vs-Auto-Issues (Bugine)



Setup of the study for the first 2 settings

29 students

- Use GitHub Classroom for assignments
  - All students for a class belongs to the same organization



all-students

@sustech-cs409/all-students

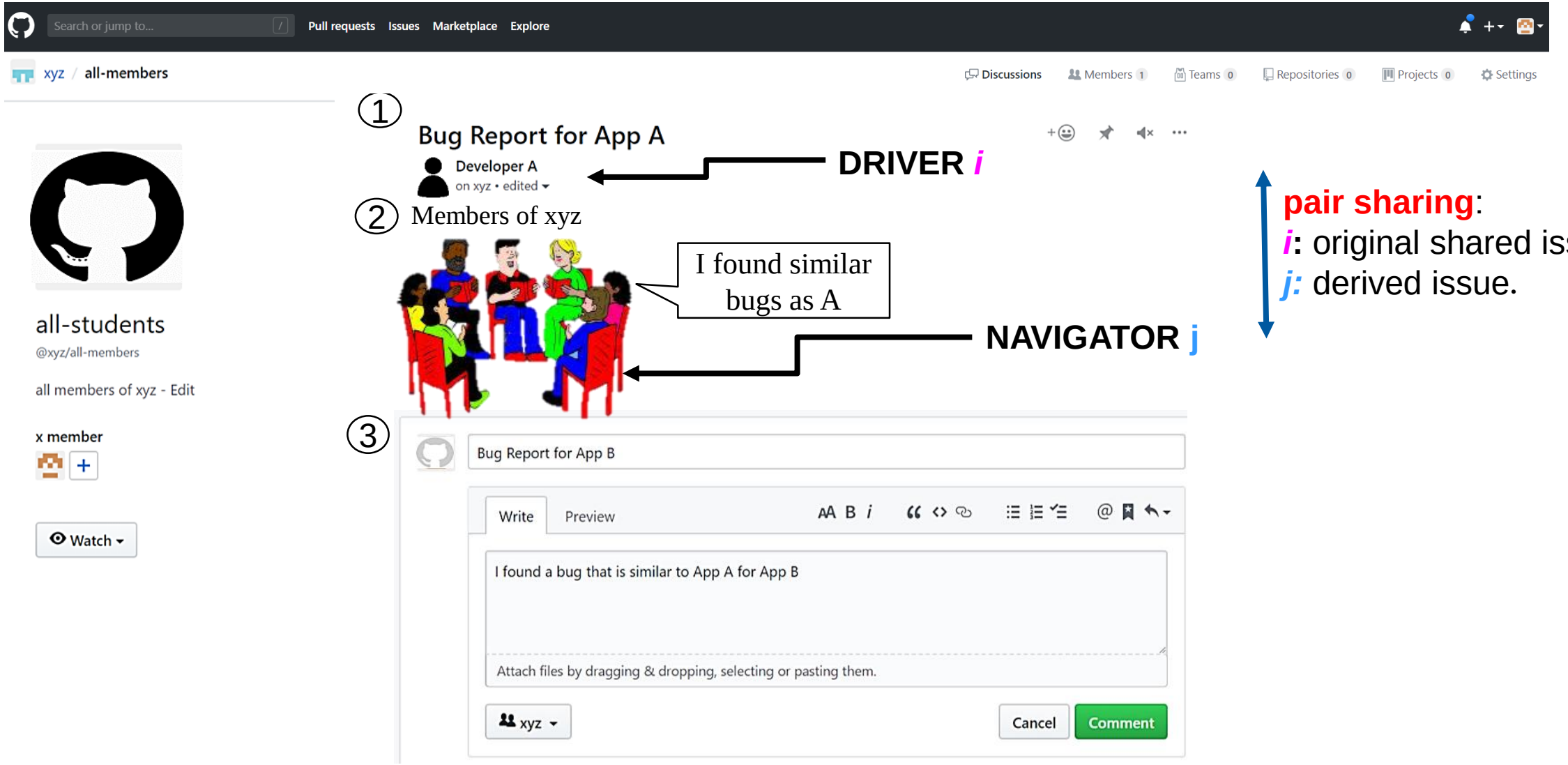
all students of cs409 - Edit

- Each student selects 1 app
- <3 teams could select the same app

➢ Selected 20 apps based on:

- ✓ Ease of use
- ✓ Contains existing tests
- ✓ Popularity on GitHub/Google Play
- ✓ Actively Maintained
- ✓ Likelihood of finding new bugs (# existing bug reports)

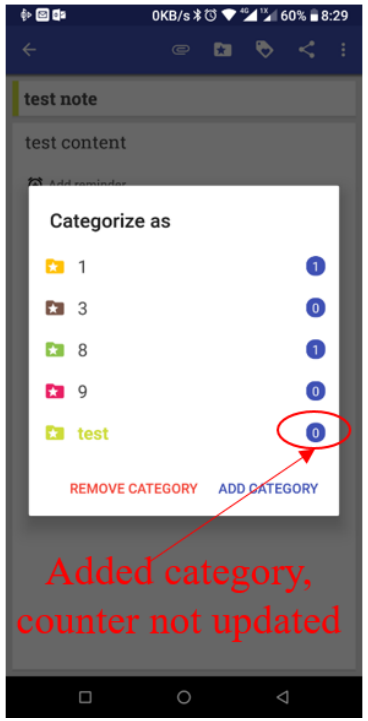
Setting 1: Coders-vs-Coders



pair sharing:
***i*:** original shared issue
***j*:** derived issue.

Motivating Example

i: original shared issue



Closed

Jump to bottom

Number of notes in a category not update immediately #Anon

Student-A opened this issue 9 months ago • edited 9 months ago

Describe the bug

The number of notes in a category does not update immediately.

Context

- Device: Nokia 7 plus
- OS version: Android 8.1.0
- App version: 5.2.2

How to reproduce

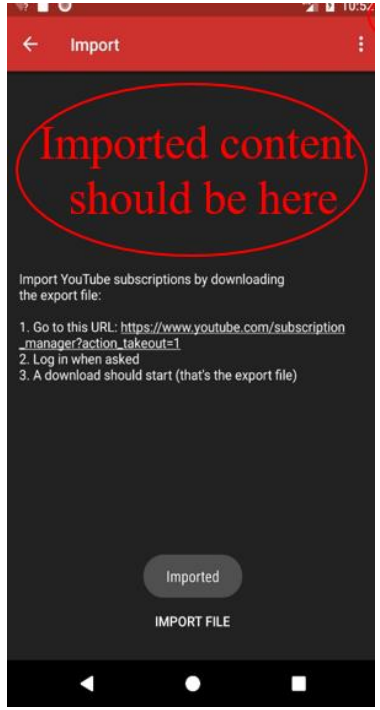
Steps to reproduce the behavior:

1. Go to a note
2. Click on category button on the top
3. Click "ADD CATEGORY"
4. Click on category button on the top again
5. The number of notes in a category does not update.

Expected behavior

The number of notes in a category should update immediately.
In this case, It should be plus one.
If I quit the note, then enter the note again. It is updated.

j: derived issue.



Open

Jump to bottom

Import subscriptions from YouTube not update immediately #Anon

Student-B opened this issue 9 months ago • edited 9 months ago

Describe the bug

After import Subscriptions from YouTube by a subscription_manager file, the Subscriptions page don't update immediately.

Context

- Device: Nexus 5X
- OS version: Android 8.0.0
- App version: newest dev (1/12/2018)

How to reporduce?

Steps to reproduce the behavior:

1. Go to Subscriptions
2. Select import from Youtube
3. Go to the URL and download a file
4. Select file and choice the file download
5. The app show a toast "imported"

Expected behavior

The app should update this page immediately. Only If I go to the home page and reopen the subscriptions menu, it will be updated.

1 1 1 +

Student-E referenced this issue from another issue

#Anon Import from previous export and update strangely

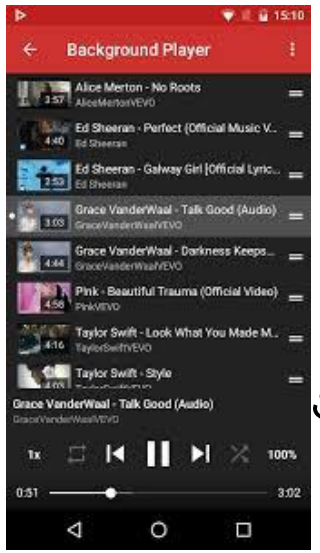
View are not immediately update, will only update the view after restarting the app

- Prevalent problems for many apps
  - 5 pair sharings

Setting 2: Coder-vs-Manual-Issues

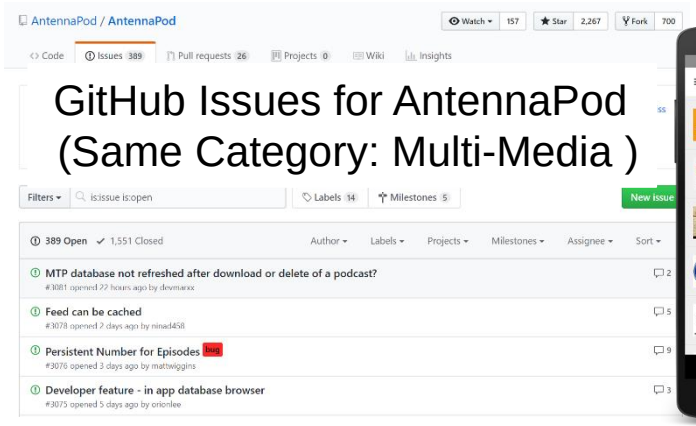
DRIVER

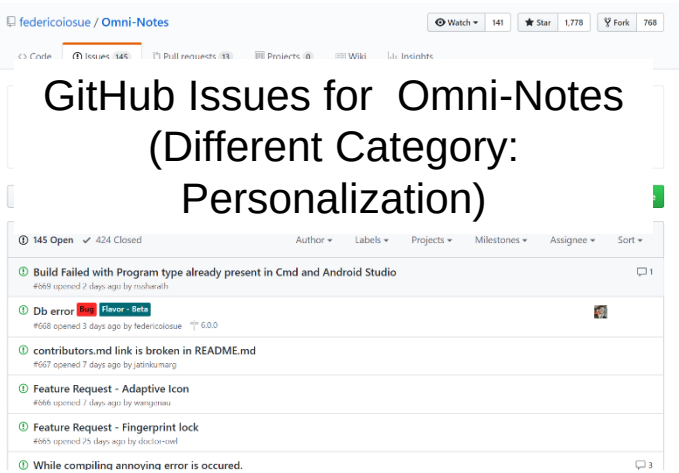
Developer for NewPipe

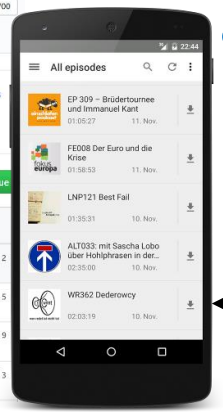


Select 5 issues

Select 5 issues



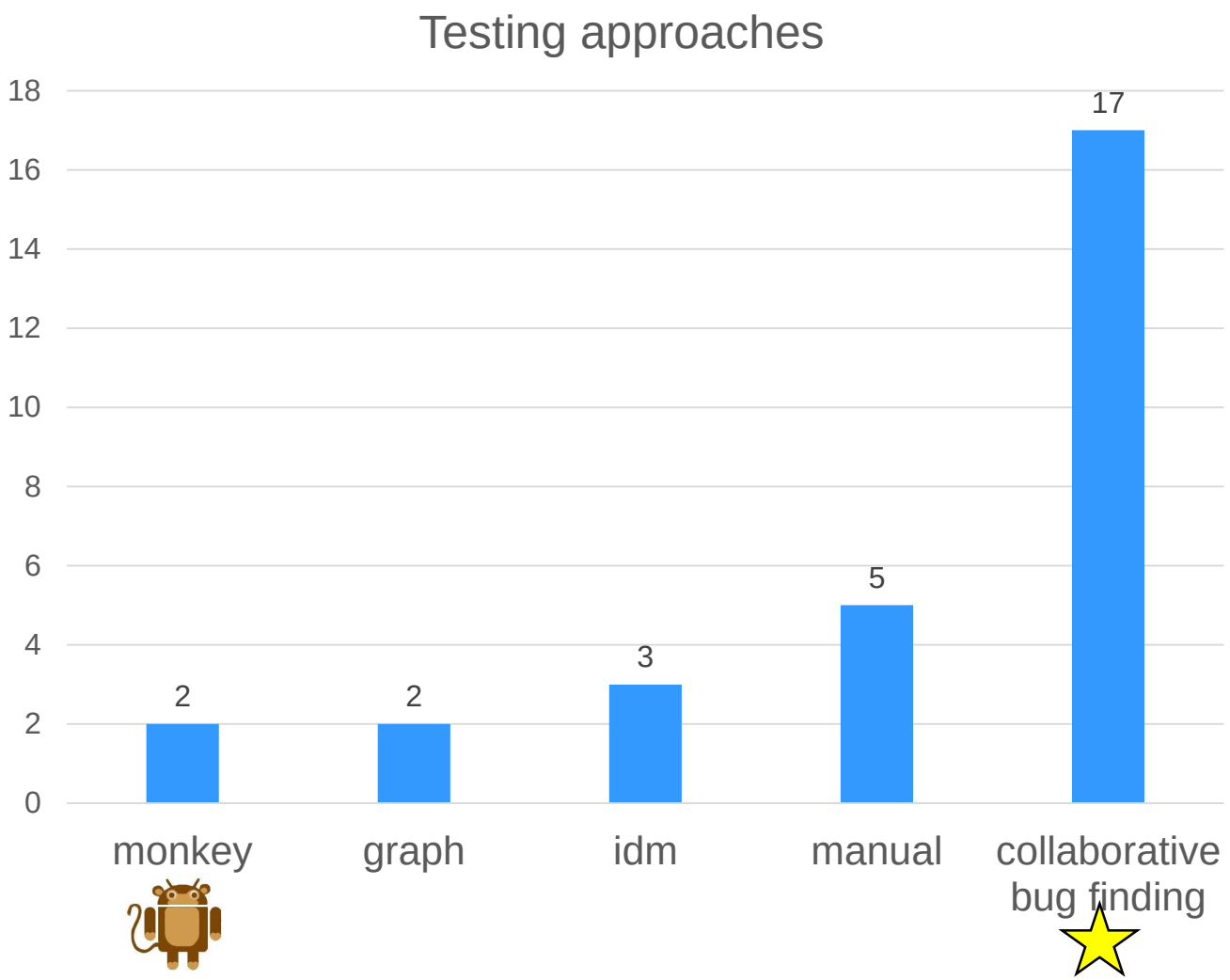




- Coder A needs to manually perform the steps below:
  1. Select the relevant issue *i*
  2. Reproduce the steps in *i*
  3. Check if the same bug described in *i* applies for the app by Coder A

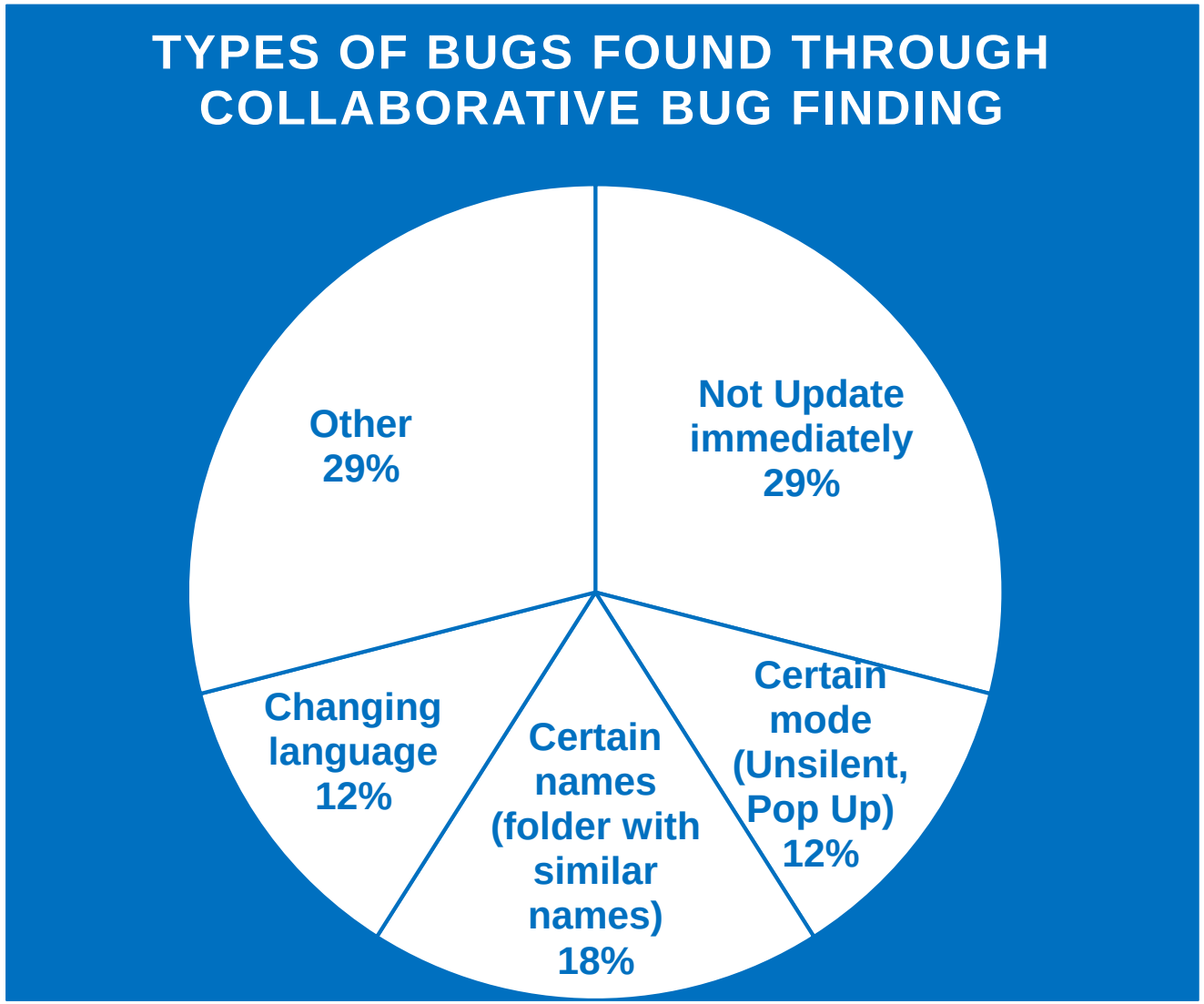
NAVIGATORS

Effectiveness of these two settings



- Collaborative bug finding finds 17 new bugs
- Automatic testing tool like Monkey only find 2 bugs
  - Crashes found are hard to reproduce

Types of Bugs found



- Our approach finds:
  - Prevalent problems (outdated view, certain names, certain mode, change of language)
  - Specific problems (29%)
- Types of bugs are mostly non-crash related
  - Our approach complements existing automated testing approaches (mostly focus on finding crashes).

What do students think about collaborative bug finding?

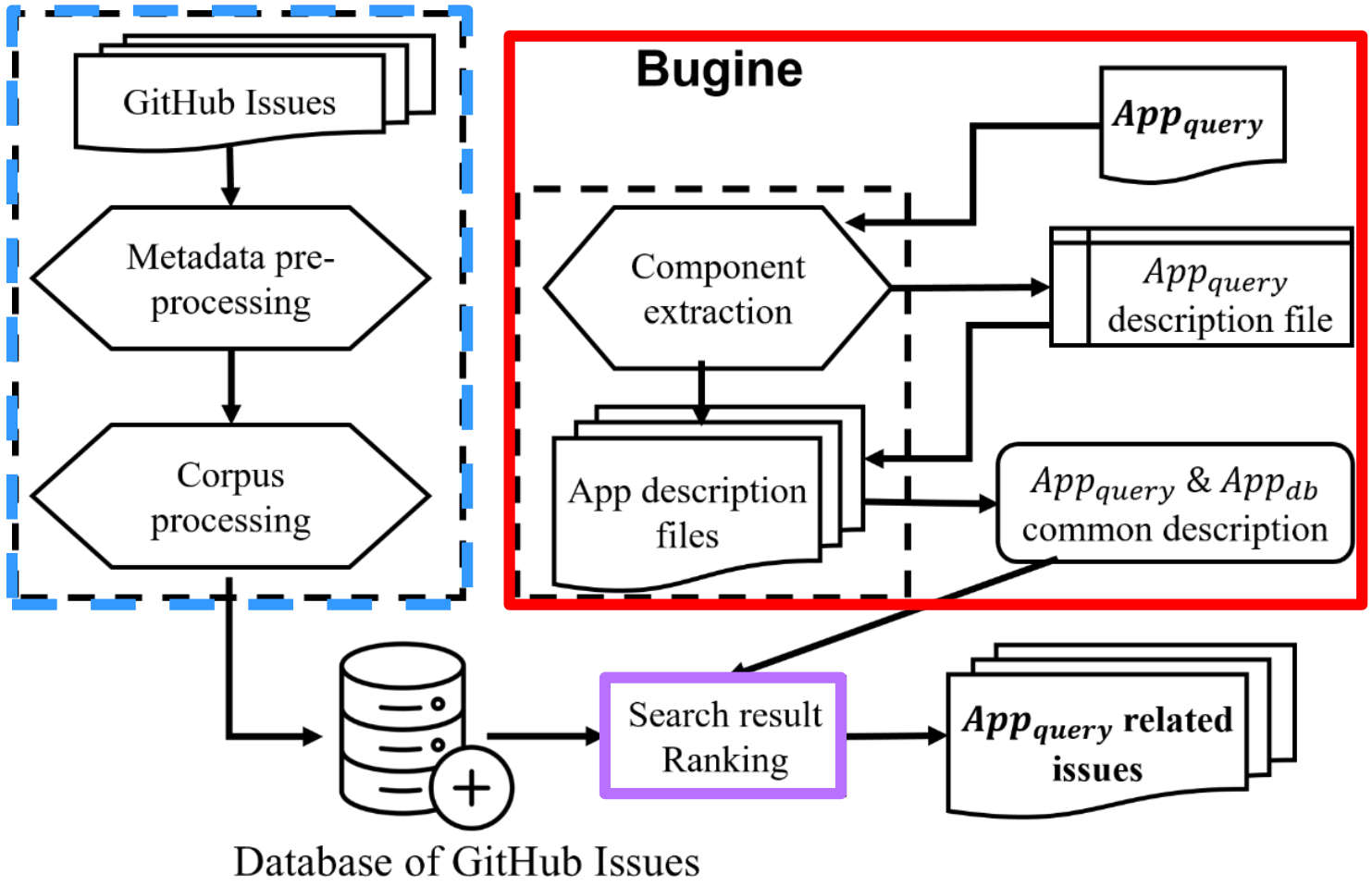
"Because many functions in the app in the same category are similar even totally same … and others' report will also inspire the mind to find bugs which I never considered."

"Collaborative bug finding takes more time to review different apps and search useful issues…but is more likely to find new bug"

Searching for issues could be time-consuming

➢ Need automation for collaborative bug finding!

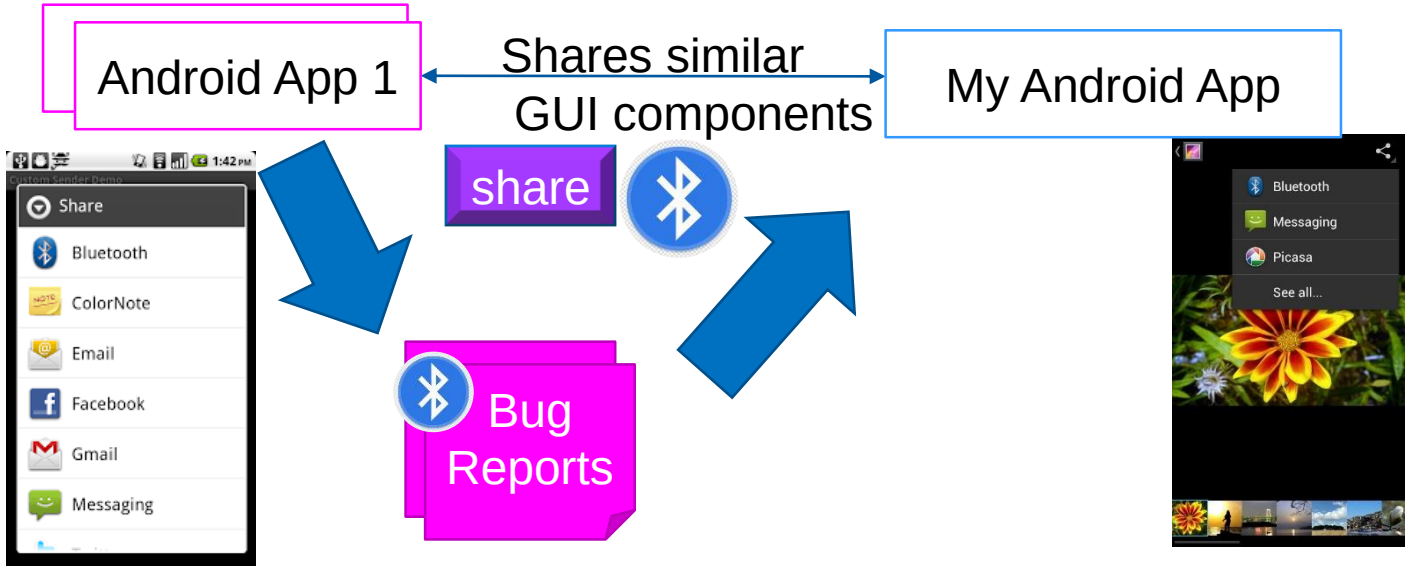
Setting 3: Coder-vs-Auto-Issues (Bugine)



- GitHub issues pre-processsing
- Given an App_{query},

1. Extracts its UI components to get its app description file
2. Use similarities between app description file for App_{query} and app description files for apps in database to search for similar apps
3. Rank issues based on quality

Finding the similarities between apps

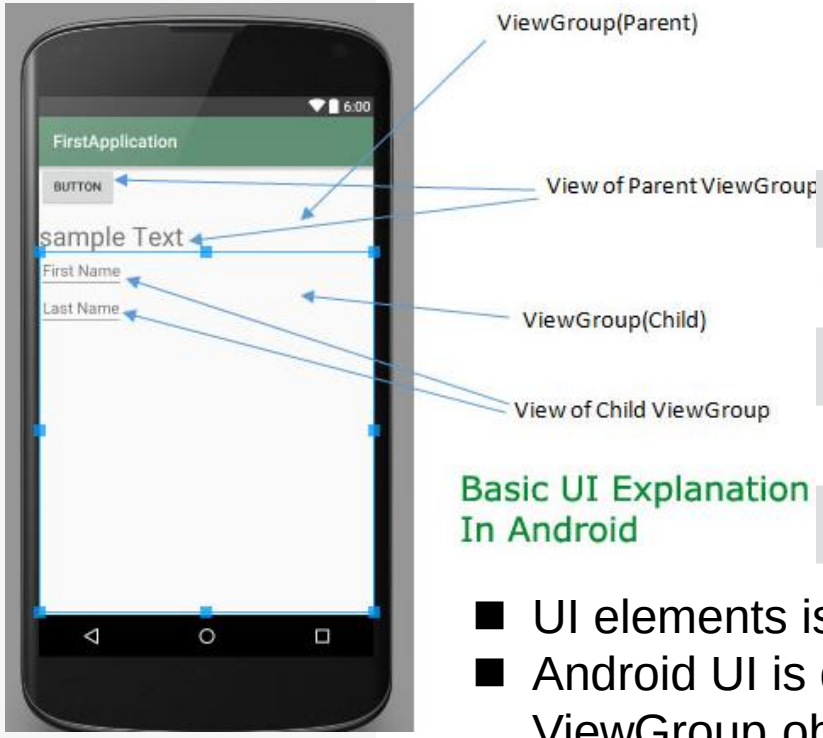


Has some bug reports that mentioned the “shared” GUI components, recommend these bug reports to my app

Are these two apps similar?

Background: What defines Android UI?

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:orientation="vertical" android:layout_width="match_parent"
 android:layout_height="match_parent">


 <Button
 android:id="@+id/buton1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Button"/>

 <TextView
 android:id="@+id/textView1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="sample Text"
 android:layout_marginTop="15dp"
 android:textSize="30dp"/>

 <RelativeLayout
 android:layout_width="match_parent"
 android:layout_height="match_parent">

 <EditText
 android:id="@+id/editTextName"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:hint="First Name"
 />

 <EditText
 android:id="@+id/editTextLastName"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:hint="Last Name"/>

 </RelativeLayout>
</LinearLayout>
```



ANDROID RESOURCE NAMING CHEAT SHEET

by @MOLSJEROEN

<WHAT>_<WHERE>_<DESCRIPTION>_<SIZE>

- <WHAT>: fixed set of options, choose the right one below
- <WHERE>: custom part Android view subclass, "all" if reused in ≠ screens
- <DESCRIPTION>: differentiate multiple elements in one screen
- <SIZE>: always optional, [xdp] or bucket [small]

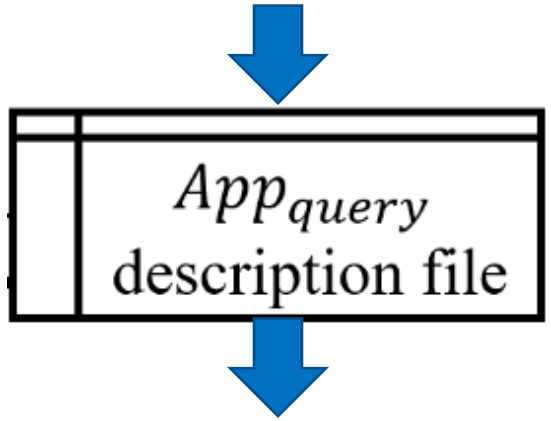
LAYOUTS <WHAT>_<WHERE>.XML	<WHAT> is activity, fragment, view, item or layout e.g. activity_main.xml
STRINGS <WHERE>_<DESCRIPTION>	e.g. main_intro all_done
DRAWABLES <WHERE>_<DESCRIPTION>_<SIZE>	e.g. all_infoicon_small main_background
IDS <WHAT>_<WHERE>_<DESCRIPTION>	<WHAT> is name of Android/Custom view class e.g. linearlayout_main_fragmentcontainer
DIMENSIONS <WHAT>_<WHERE>_<DESCRIPTION>_<SIZE>	<WHAT> is width, height, size, margin, padding, elevation, keyline or textsize e.g. keyline_all_text

- UI elements is usually declared in XML files
- Android UI is defined via the hierarchy of View and ViewGroup objects
- Names of Android resource following certain conventions

From： https://abhiandroid.com/ui/xml

Naming information for extracting description files

Component	Description	Example	Extracted Names
Resource Name	Resource Name	android:id="@+id/my_btn"	my_btn
View Name	Name for the type of UI component	<Button android:id="@+id/my_btn" />	Button
XML File Names	Layout name	main_layout.xml	main_layout

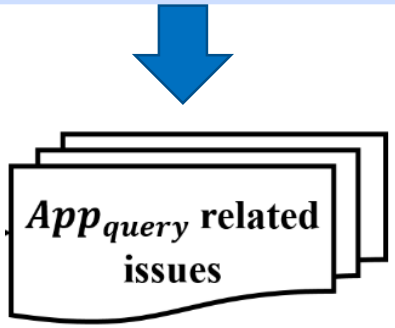


For each XML file, convert each view and each resource in App_{query} to the query of the form:

XML file name $\wedge$ View Name $\wedge$ Resource name

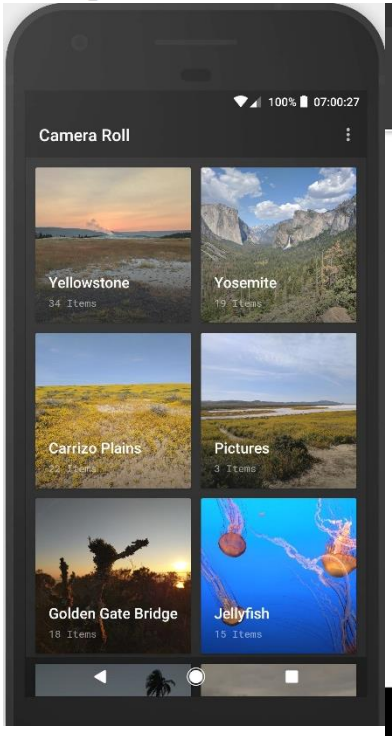
Ranking GitHub Issues

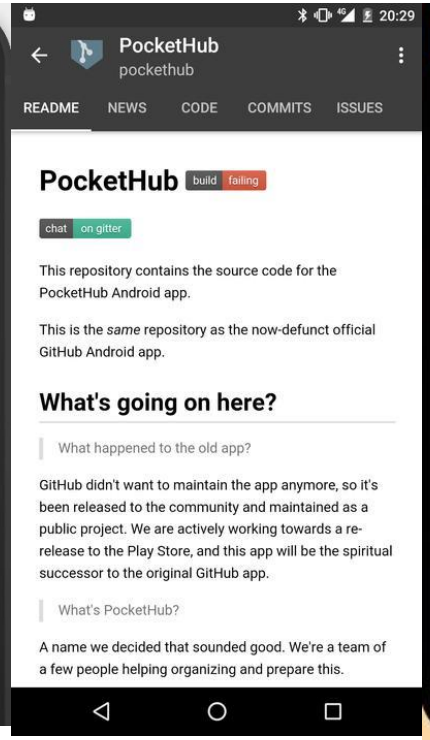
Factors	Description	Rationale
Issue length	Word count of issue body (int)	Longer issue is better
Issue Status	Closed or opened (binary)	Issue is more important if it is : ✓ Closed ✓ Fixed ✓ has more replies (comments)
Ref Commit SHA	Commit SHA referenced by issue (binary)	
Issue Reply Number	The number of replies that an issue received (int)	
Hit_all	Find all search keywords in the corpus (binary)	Search for shared UI components: ✓ Match all keywords (better) ✓ Has some shared components
Hit_overlap	Overlap coefficient between search keywords and corpus (float)	
Hit_Hot_Words	Word count of descriptive hot words like reproduce, defect (int)	Issues that meet the criteria for good bug reports is better

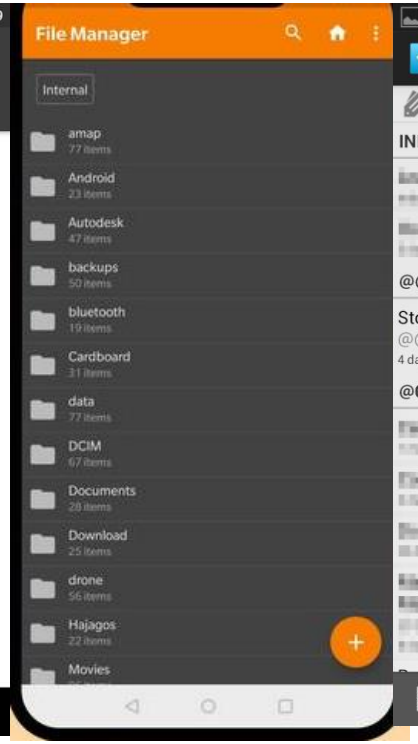


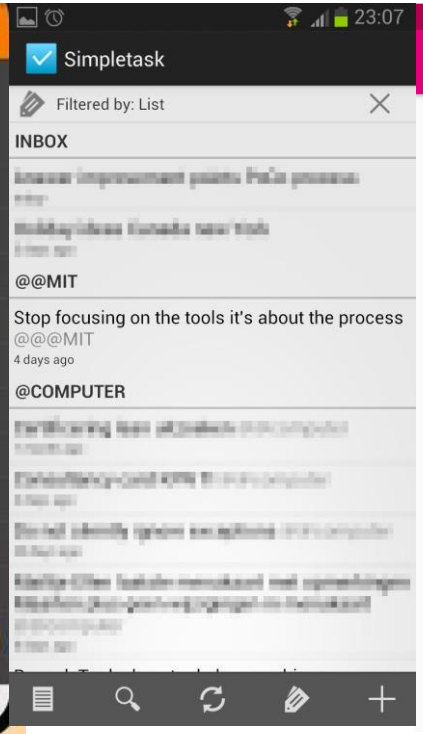
Evaluation of Bugine

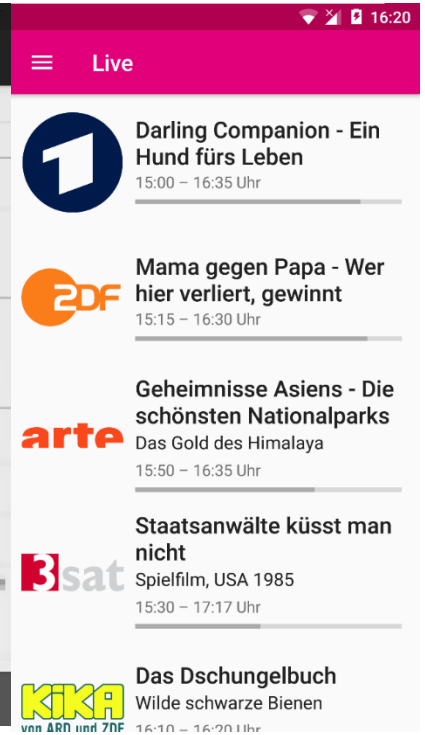
App Name	Category	KLOC	#Downloads	Rating	Version No.	#GitHub Stars	#GitHub Issue (closed)
Camera-Roll	Gallery	26.00	100,000+	4.2	1.0.6	420	227(133)
PocketHub	GitHub client	31.35	10,000+	3.3	0.5.1	9429	644(526)
Simple File Manager	Explorer	5.84	50,000+	4.5	6.3.4	378	189(130)
Zapp	Broadcast	8.41	N.A.	N.A.	3.2.0	60	151(137)
Simpletask	Reminder	24.80	10,000+	4.7	10.3.0	349	821(583)











Evaluate the ranking performance of Bugine

Use two metrics commonly used in prior recommendation systems:

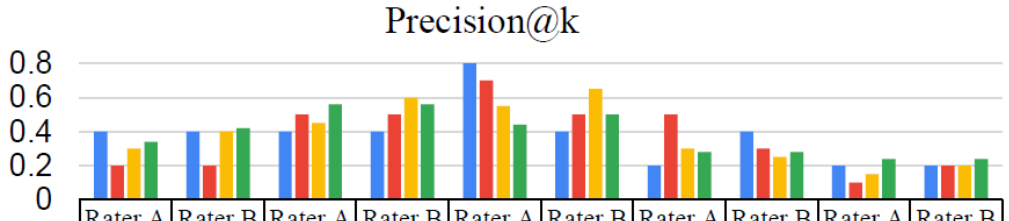
$$(1)\ \text{Prec@k} = \frac{\#\ relevant\ documents\ in\ top\ k}{k}$$

- Retrieval precision over the top k documents in the ranked list

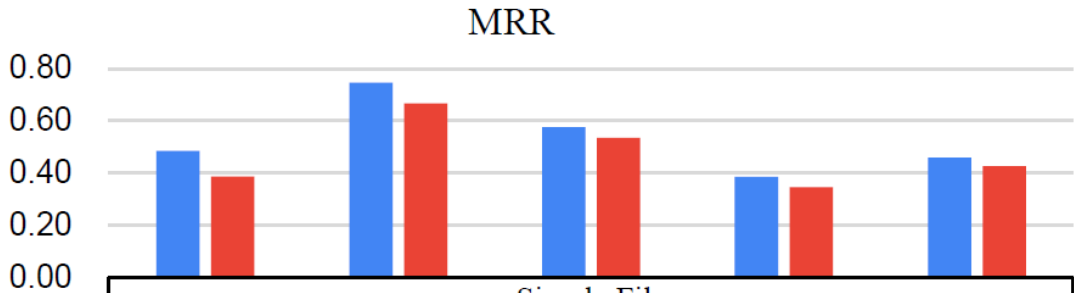
$$(2)\ \text{Mean Reciprocal Rank (MRR)} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{first_q}$$

- For each query q, the MRR measures the position $first_q$ of the first relevant document in the ranked list

Ranking performance of Bugine



	Camera-Roll		PocketHub		Simple File Manager		zapp		Simpletask	
	Rater A	Rater B	Rater A	Rater B	Rater A	Rater B	Rater A	Rater B	Rater A	Rater B
Prec@5	0.4	0.4	0.4	0.4	0.8	0.4	0.2	0.4	0.2	0.2
Prec@10	0.2	0.2	0.5	0.5	0.7	0.5	0.5	0.3	0.1	0.2
Prec@20	0.3	0.4	0.45	0.6	0.55	0.65	0.3	0.25	0.15	0.2
Prec@50	0.34	0.42	0.56	0.56	0.44	0.5	0.28	0.28	0.24	0.24



	Camera-Roll	PocketHub	Simple File Manager	zapp	Simpletask
Rater A	0.48	0.75	0.58	0.39	0.46
Rater B	0.39	0.67	0.53	0.34	0.43

- Prec@10: 0.1–0.7
  - Among the top 10 issues recommended by Bugine, there is ≥ 1 relevant issue
- MRR values: 0.34–0.75
  - Ranking for the first relevant document is btw 3rd (0.34) and 1st (0.75)

➢ Bugine could recommend relevant issues for all evaluated apps

Bugs found by Bugine

App Name	#Bugs Found (new, old)
Camera-Roll	(11, 0)
PocketHub	(12, 2)
Simple File Manager	(6, 2)
Zapp	(2, 7)
Simpletask	(3, 2)

- Found 34 new bugs and 13 old bugs
  - In first two settings, 29 students find 17 new bugs in 20 apps
  - ➢ Bugine could discover more bugs despite being evaluated only on five apps.

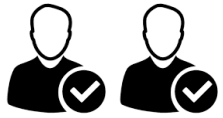
Feedbacks from the ICSE 2020 Reviewers

"This is one of those simple but great ideas that make a lot of sense…"

"I thought the idea of collaborative testing was intriguing and thought provoking… I really liked the effort by the authors to think creatively about this and present an out of the box idea for test generation"

"The idea of collaborative bug finding is refreshing and interesting"

Interesting Research Questions:

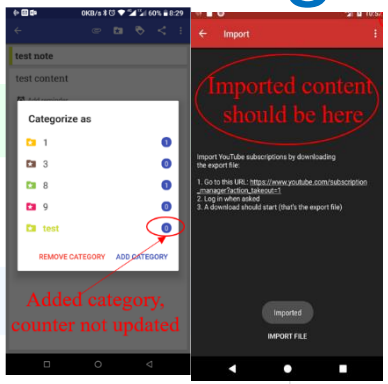


Can the authors provide any insights on how to automate collaborative bug finding?

- Can we extract fix patterns from these common issues?

Could we make sure of the fixes of similar bugs for Automated Program Repair?

```
+ import rx.Observable;
+ import rx.functions.Func1;

  import static com.nineoldandroids.view.ViewPropertyAnimator.animate;
  import static java.lang.Integer.parseInt;

@@ -1216,10 +1218,15 @@ private void toggleChecklist2(final boolean keepChecked, final boolean showCheck
         * Categorize note choosing from a list of previously created categories
         */
        private void categorizeNote() {
-               // Retrieves all available categories
-               final ArrayList<Category> categories = DbHelper.getInstance().getCategories();

                String currentCategory = noteTmp.getCategory() != null ? String.valueOf(noteTmp.getCategory().getId()) :
+         final List<Category> categories = Observable.from(DbHelper.getInstance().getCategories()).map(category -> {
+             if (String.valueOf(category.getId()).equals(currentCategory)) {
+                 category.setCount(category.getCount() + 1);
+             }
+             return category;
+         }).toList().toBlocking().single();
+
```



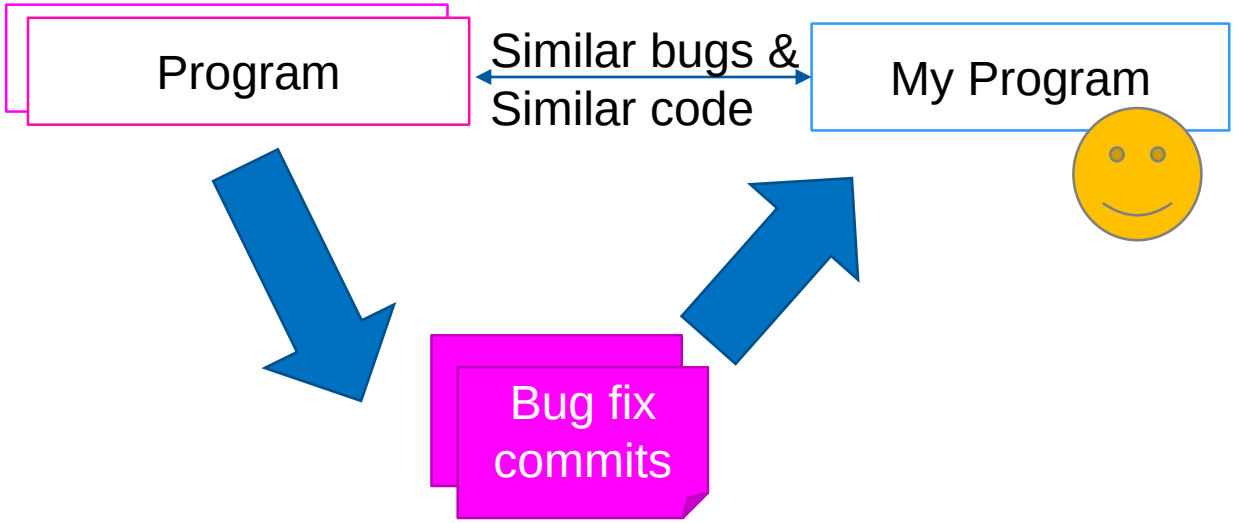
mauriciocolli commented on 25 Apr 2019 • edited

- Sort the items chronologically and by type (live, then upload date in descending order)
- Let users group their subscriptions
- Load the feed items in parallel in a background service (like importing works)
- Show the watching count in live stream items
- Use `Groupie` library for easier development of lists

- Could provide high-level fix suggestion
  - Use Observable vs. Background Service for handling asynchronous events for fixing the outdated view problem

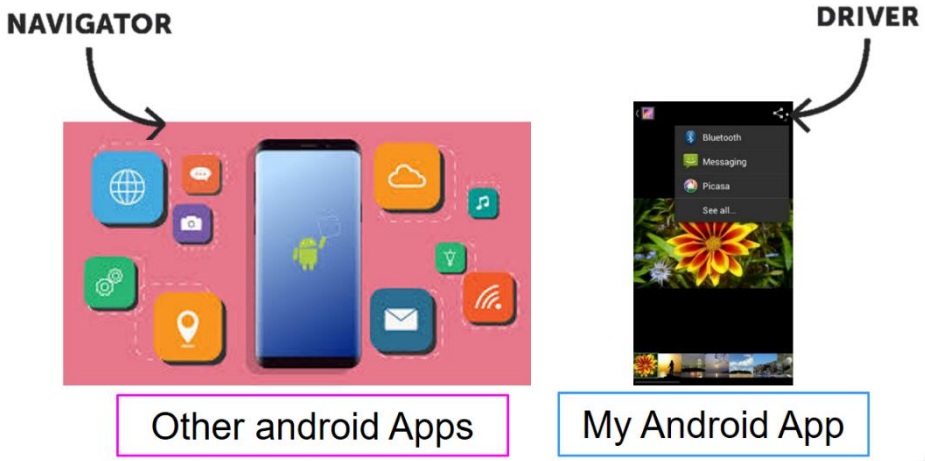
Future Work: Collaborative Testing and Repair

- How about recommendation systems for bug fix commits?



Has bug fix commits, recommend these commits to another Program

Summary



- 3 settings:
  - Coders-vs-Coders
  - Coder-vs-Manual-Issues
  - Coder-vs-Auto-Issues (Bugine)

- Reusing bug reports from different apps
  - ✓ Exploit the redundancies of bug reports in open-source repositories
  - ✓ Use similarities between apps
- New way of testing Android Apps
  - ✓ Instead of test input generation, we re-formulate the test generation problem as bug report recommendation problem
    - ✓ Bugine recommends relevant bug reports automatically
  - ✓ Developer could read reports written in Natural Language
    - ✓ Do not need to learn a new testing framework nor API
- Found 51 new bugs, 5 confirmed, 7 fixed