Regression Testing For Android API Changes

Zachary Vasey, Kunal Aggarwal, Dr. Guowei Yang
Department of Computer Science, Texas State University, San Marcos, TX

Abstract

In the first quarter of 2018, the Android smartphone operating system dominated the market share. Android's API updates every few months which often affects the functionality of Android applications. With an absence of tools that help android developers test their apps after an update, android developers are stuck testing the entirety of the Android application. This is certainly expensive and redundant. We propose a regression testing approach to the two prominent testing techniques in Android development today: testing by running test cases and testing by exploring state space.

Problem

- Re-executing the same application may produce unexpected results when the Android API updates
  - API changes every few months
  - API is the foundation of Android software
- Running test cases:
  - Run all the test cases again, which can be expensive
- Exploring state space:
  - Re-exploring all the possible states of an app is time consuming

Approach/Framework

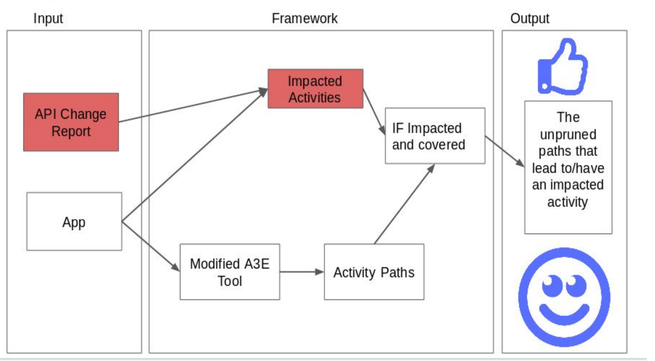

Figure 1: Framework for regression state space exploration.

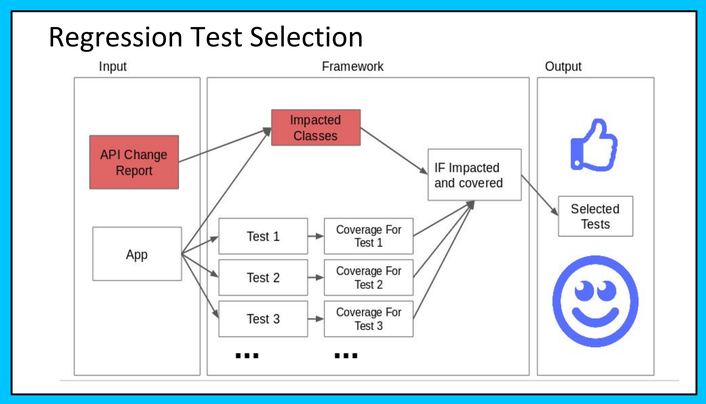

Figure 2: Framework for regression test selection.

App Names	Number of Tests	API Change	Test Class Selection
Amaze File Manager	122	27-28	20/21
Android Wikipedia	478	27-28	55/57
Materialistic	1007	27-28	61/62
K9 Mail	622	22-23	51/56
Firefox Focus	92	27-28	10/10

Figure 3: Subject applications for evaluation of regression test selection. They all have test suites, are open sourced and are openly available for Android users.

Materialistic App. API 27 → API 28

```
test_class_selecctor.py running in PreferencesActivityTest .java
SELECTED:       29 out of 38
['HelpLazyLoadView.java\n', 'HelpLazyLoadView$1.java\n', 'ItemSyncWifiReceiver.java\n', 'AlgoliaClient.java\n', 'FavoriteManager$Companion.java\n', 'FontSizePrefer
ava\n', 'FontPreference.java\n', 'SimpleSpinnerPreference.java\n', 'PreferenceHelp.java\n', 'SpinnerPreference.java\n', 'ThemePreference.java\n', 'ThemePreference$
ec.java\n', 'ThemePreference$DayNightSpec.java\n', 'ThemePreference$ThemeSpec.java\n', 'AdBlocker.java\n', 'Preferences.java\n', 'Preferences$BoolToStringPref.java
Preferences$StoryViewMode.java\n', 'Preferences$Theme.java\n', 'PreferencesActivity.java\n', 'PreferencesActivity$SettingsFragment.java\n', 'R$styleable.java\n', '
ntDelegate.java\n', 'ThenedActivity.java\n', 'AppUtils.java\n', 'Application.java\n', 'WidgetProvider.java\n', 'TintableTextView.java\n', 'TextView.java\n']
PreferencesActivityTest test analyzed...


test_class_selecctor.py running in PreferencesMigrationTest .java
SELECTED:       15 out of 15
['ItemSyncWifiReceiver.java\n', 'AlgoliaClient.java\n', 'FavoriteManager$Companion.java\n', 'ThemePreference.java\n', 'ThemePreference$DarkSpec.java\n', 'ThemePref
SDayNightSpec.java\n', 'ThemePreference$ThemeSpec.java\n', 'AdBlocker.java\n', 'Preferences.java\n', 'Preferences$BoolToStringPref.java\n', 'Preferences$Theme.java
R$styleable.java\n', 'AppUtils.java\n', 'Application.java\n', 'WidgetProvider.java\n']
PreferencesMigrationTest test analyzed...


test_class_selecctor.py running in PreferencesThemeTest .java
SELECTED:       15 out of 15
['ItemSyncWifiReceiver.java\n', 'AlgoliaClient.java\n', 'FavoriteManager$Companion.java\n', 'ThemePreference.java\n', 'ThemePreference$DarkSpec.java\n', 'ThemePref
SDayNightSpec.java\n', 'ThemePreference$ThemeSpec.java\n', 'AdBlocker.java\n', 'Preferences.java\n', 'Preferences$BoolToStringPref.java\n', 'Preferences$Theme.java
R$styleable.java\n', 'AppUtils.java\n', 'Application.java\n', 'WidgetProvider.java\n']
```

```
AboutActivityTest.java
AccountAuthenticatorTest.java
AdBlockerTest.java
AlgoliaClientTest.java
AlgoliaPopularClientTest.java
AppUtilsTest.java
AuthenticatorServiceTest.java
BaseListActivityLandTest.java
BaseListActivityTest.java
CacheTest.java
ComposeActivityTest.java
CustomTabsDelegateTest.java
DrawerActivityLoginTest.java
DrawerActivityTest.java
FavoriteActivityEmptyTest.java
FavoriteActivityTest.java
```

```
FavoriteManagerTest.java
FeedbackActivityTest.java
FeedbackClientTest.java
FontPreferenceTest.java
FontSizePreferenceTest.java
HackerNewsClientTest.java
HackerNewsItemTest.java
ItemActivityTest.java
ItemFragmentMultiPageTest.java
ItemFragmentSinglePageTest.java
ItemSyncAdapterTest.java
ItemSyncJobServiceTest.java
KeyDelegateTest.java
LauncherActivityTest.java
ListFragmentTest.java
ListFragmentViewHolderEdgeTest.java
```

Figure 4: Regression test selection results for the Materialistic App when API is changed from Version 27 to Version 28.

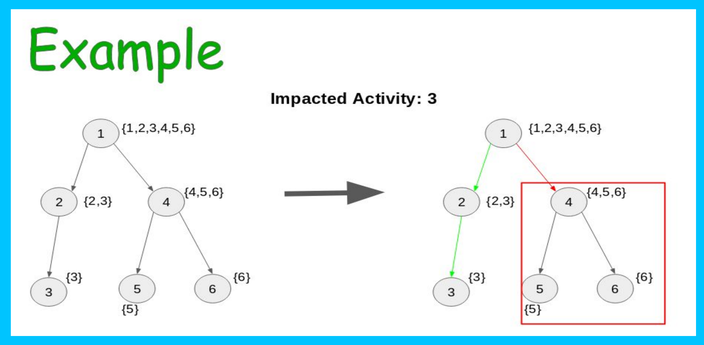

Figure 5: An example to illustrate regression state space exploration for A3E.

Results

- Framework shows the number of tests that need to be re-executed given a change in the Android API.
- Framework shows the impacted and covered classes for each test.
- Most tests were impacted by the API change and are thus selected for re-execution.

Conclusions and Future Work

- Regression testing can potentially reduce the cost of testing for API changes.
- Class level regression test selection is not precise enough to generate significant results.
- Conduct finer-grain regression test selection (e.g. methods level analysis, statement level analysis).
- Implement and evaluate Regression State Space Exploration.

Acknowledgments

We thank the National Science Foundation (NSF) for funding this research under the Research Experiences for Undergraduates Program (CNS-1358939, CNS-1659807) and the infrastructure support provided by the NSF-CRI 1305302 award.

References

1. Guowei Yang, Jeffrey Jones, Austin Moninger, and Meiru Che. 2018. How do Android operating system updates impact apps? *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems - MOBILESoft 18* (2018).
2. Quan Do, Guowei Yang, Meiru Che, Darren Hui, and Jefferson Ridgeway. 2016. Redroid: A Regression Test Selection Approach for Android Applications. *Proceedings of the 28th International Conference on Software Engineering and Knowledge Engineering* (January 2016).
3. Tanzirul Azim and Iulian Neamtiu. 2013. Targeted and depth-first exploration for systematic testing of android apps. *ACM SIGPLAN Notices* 48, 10 (December 2013), 641–660.
4. Alessandro Orso and Gregg Rothermel. 2014. Software testing: a research travelogue (2000–2014). *Proceedings of the on Future of Software Engineering - FOSE 2014* (2014).