

# How to build rock-solid apps & keep 100m+ users happy

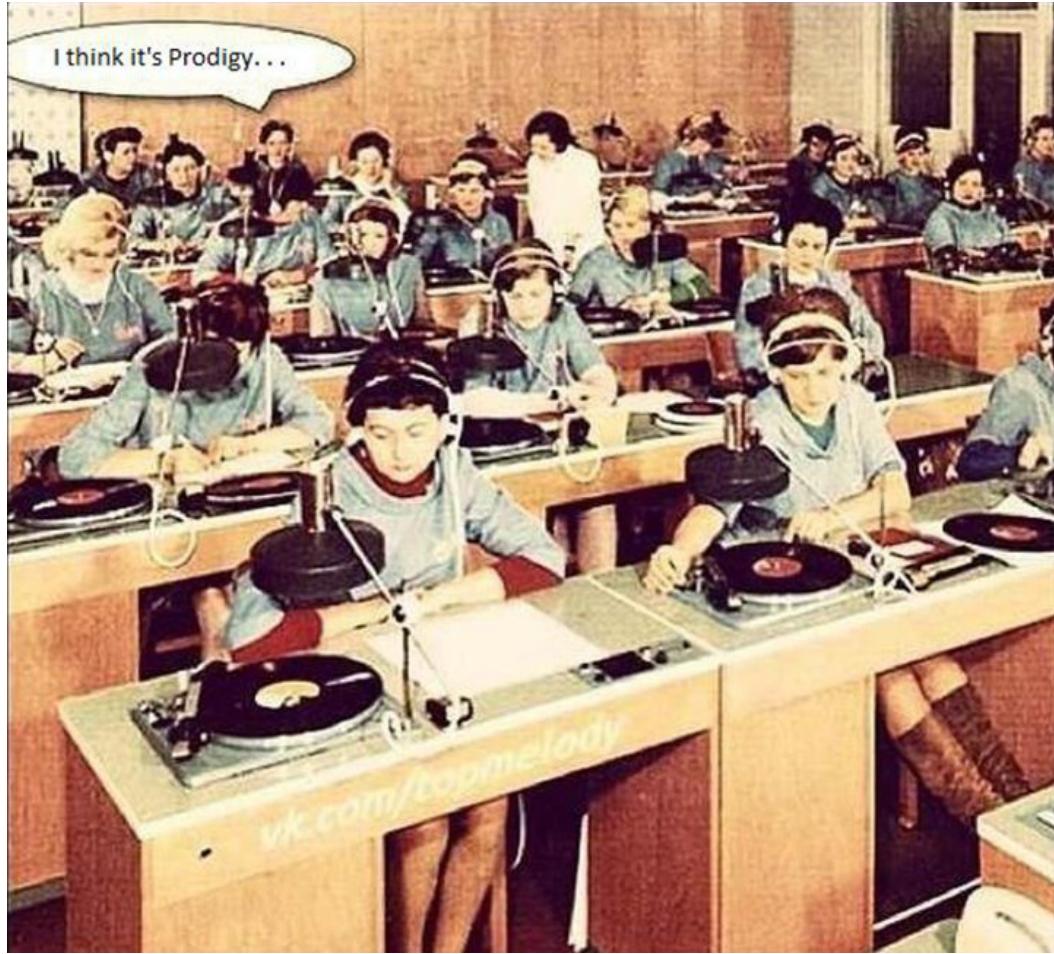
Iordanis Giannakakis [iordanis\\_g](#)

Savvas Dalkitsis [@geeky\\_android](#)

**QCon**

#qconlondon

# How Shazam works



# Some numbers



# Happy users



Even developers?



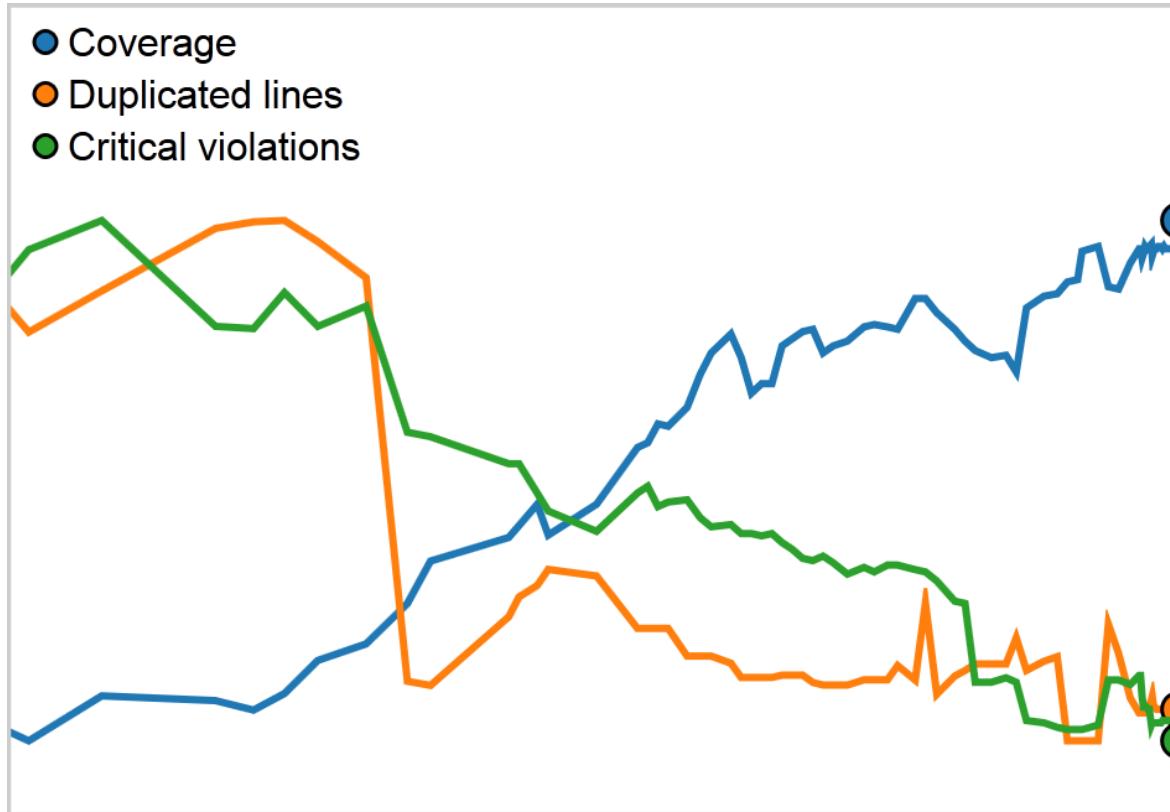
# Android testing



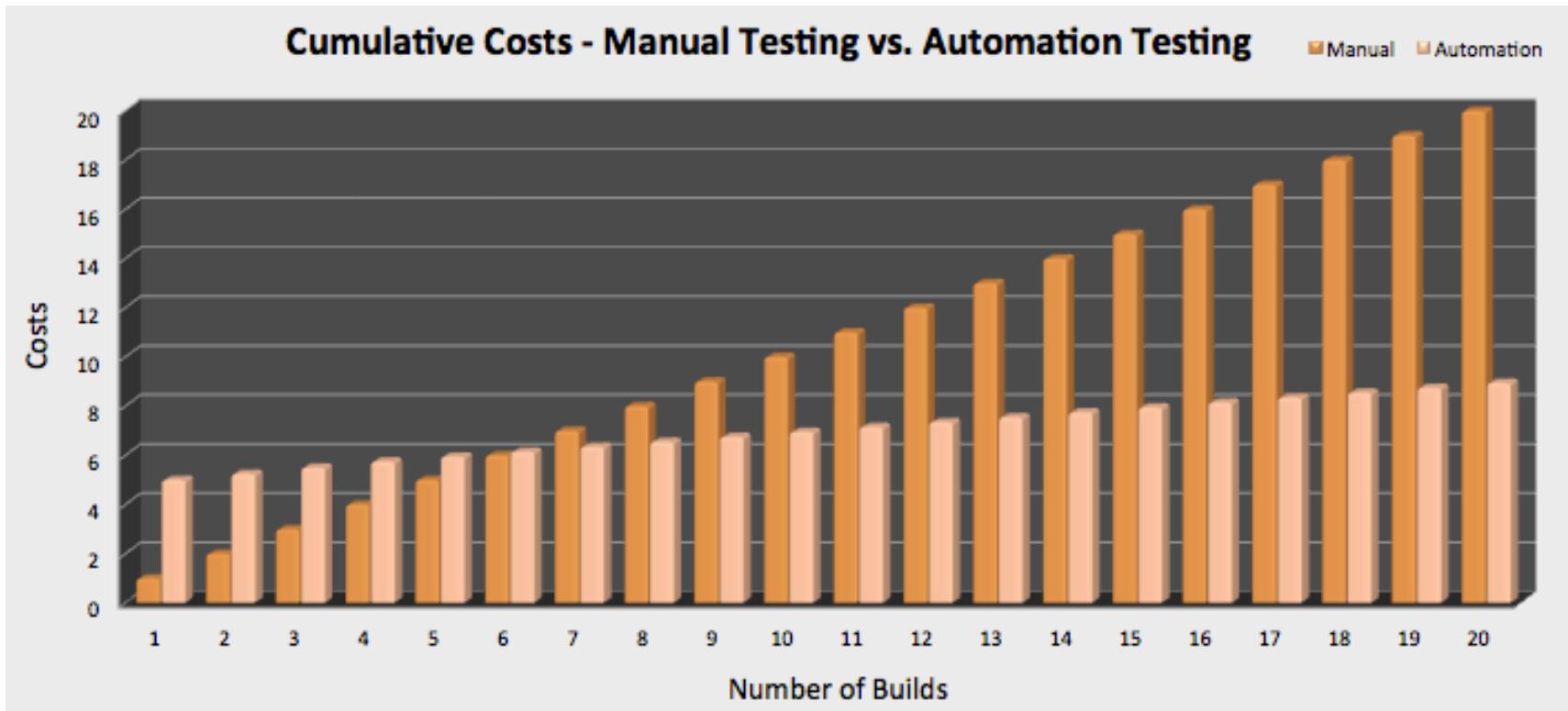
# Faster release cycles



# Better code



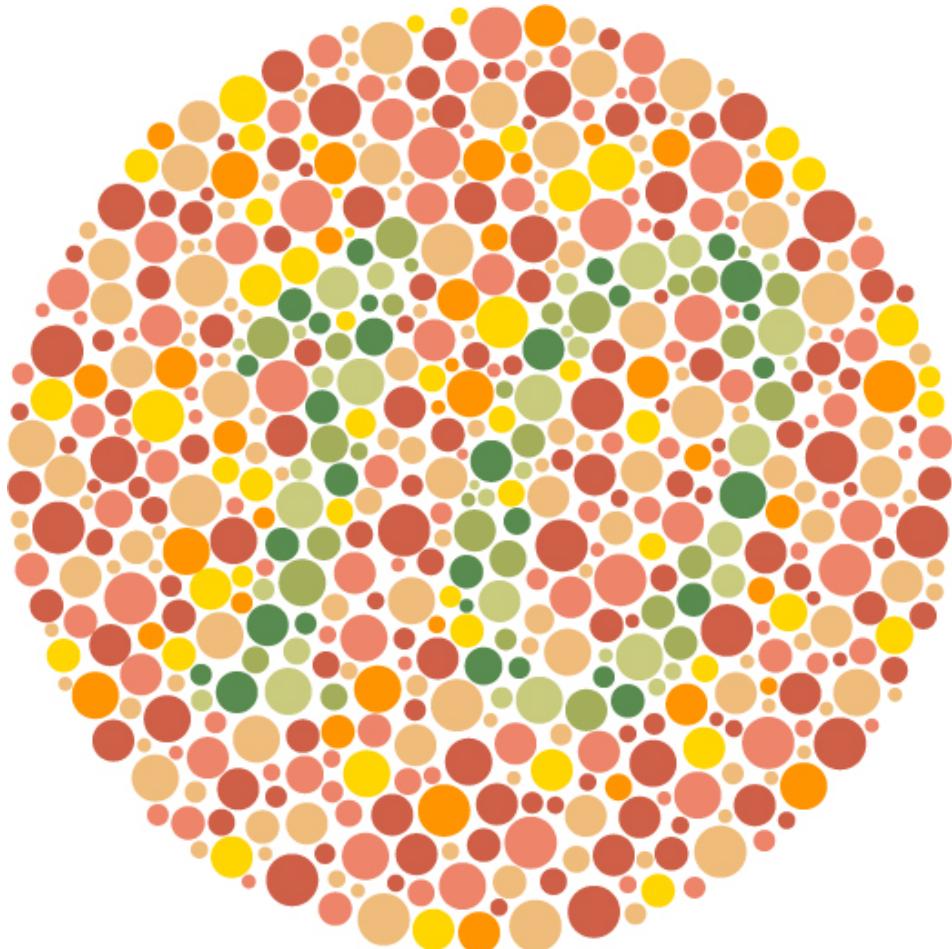
# Cheaper



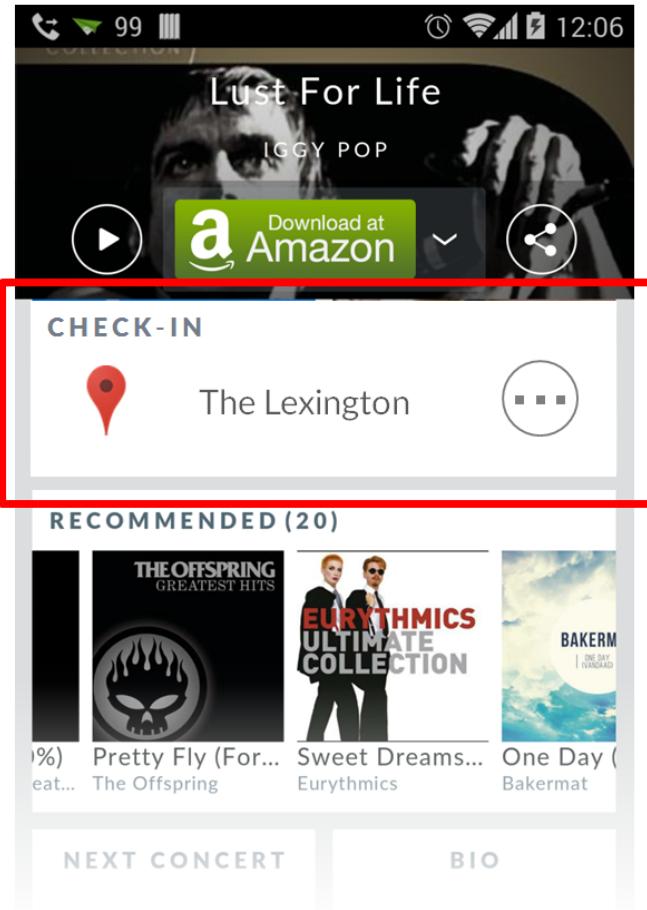
<http://testdroid.com/testdroid/5851>



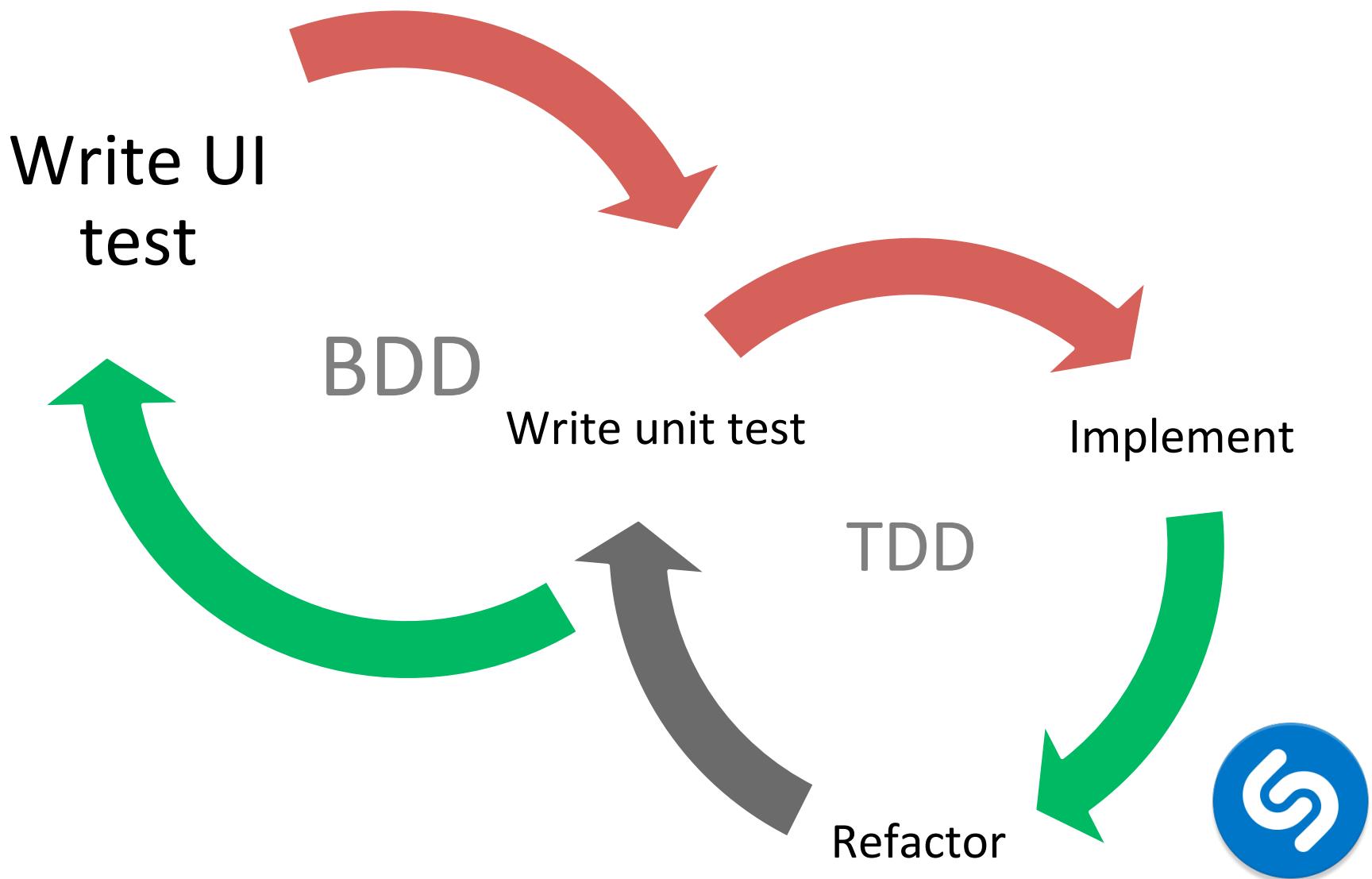
# Easy



# A user walks into a bar



# Doing it the test driven way

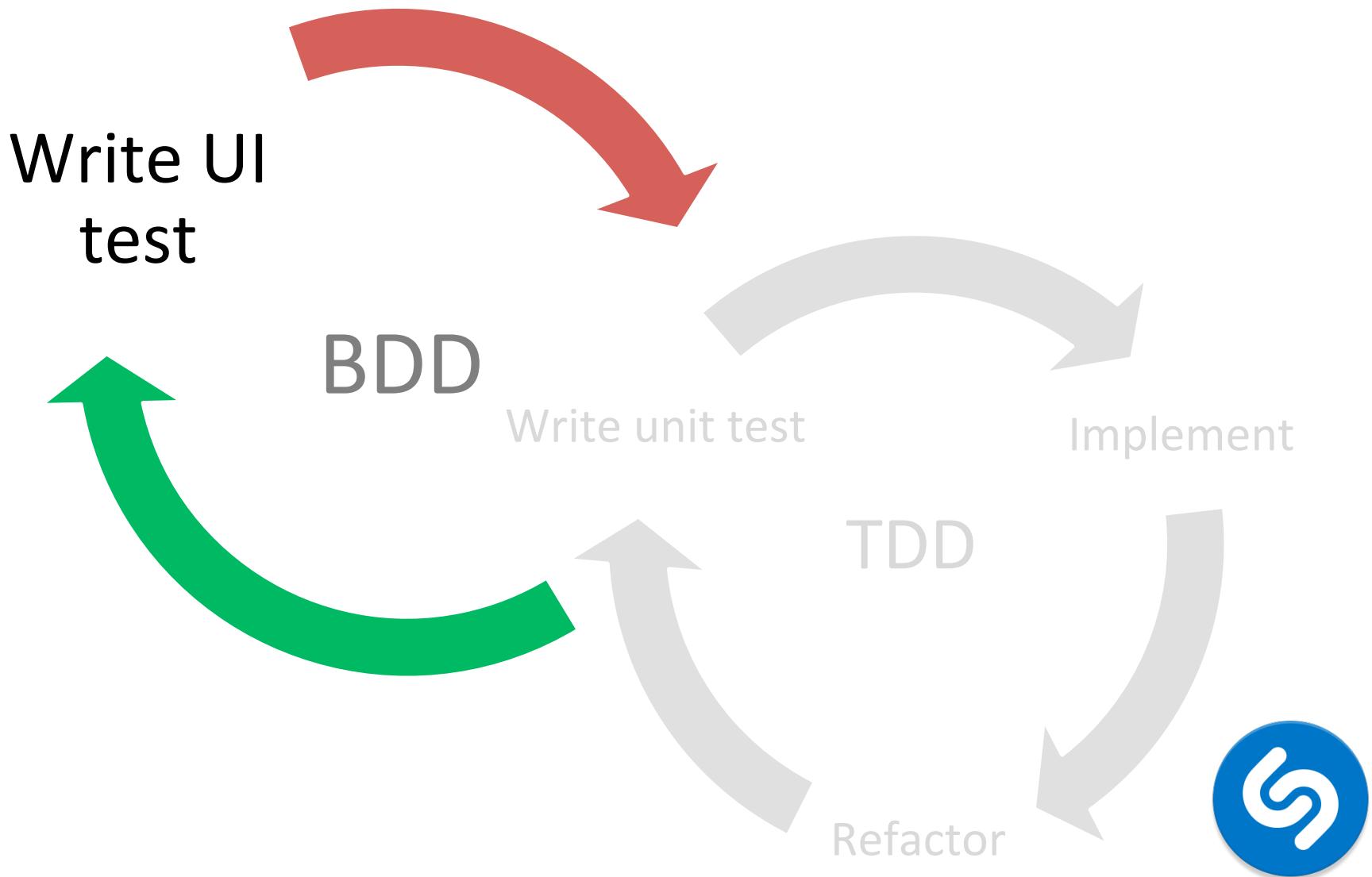


# Test first

- Generally easier
- Eliminates external dependencies
- Easily repeatable



# The Acceptance Tests cycle



# Acceptance criteria

- Given: arrange
- When: act
- Then: assert



# Example

Given a user is near a music venue  
And the server always returns a known result

When the user Shazams

Then the user can check-in their discovery



# gwen

```
given(user).isNear(lexington());
given(server).returns(lustForLife());

when(user).shazams();

then(user).canCheckIn(lustForLife(), lexington());
```



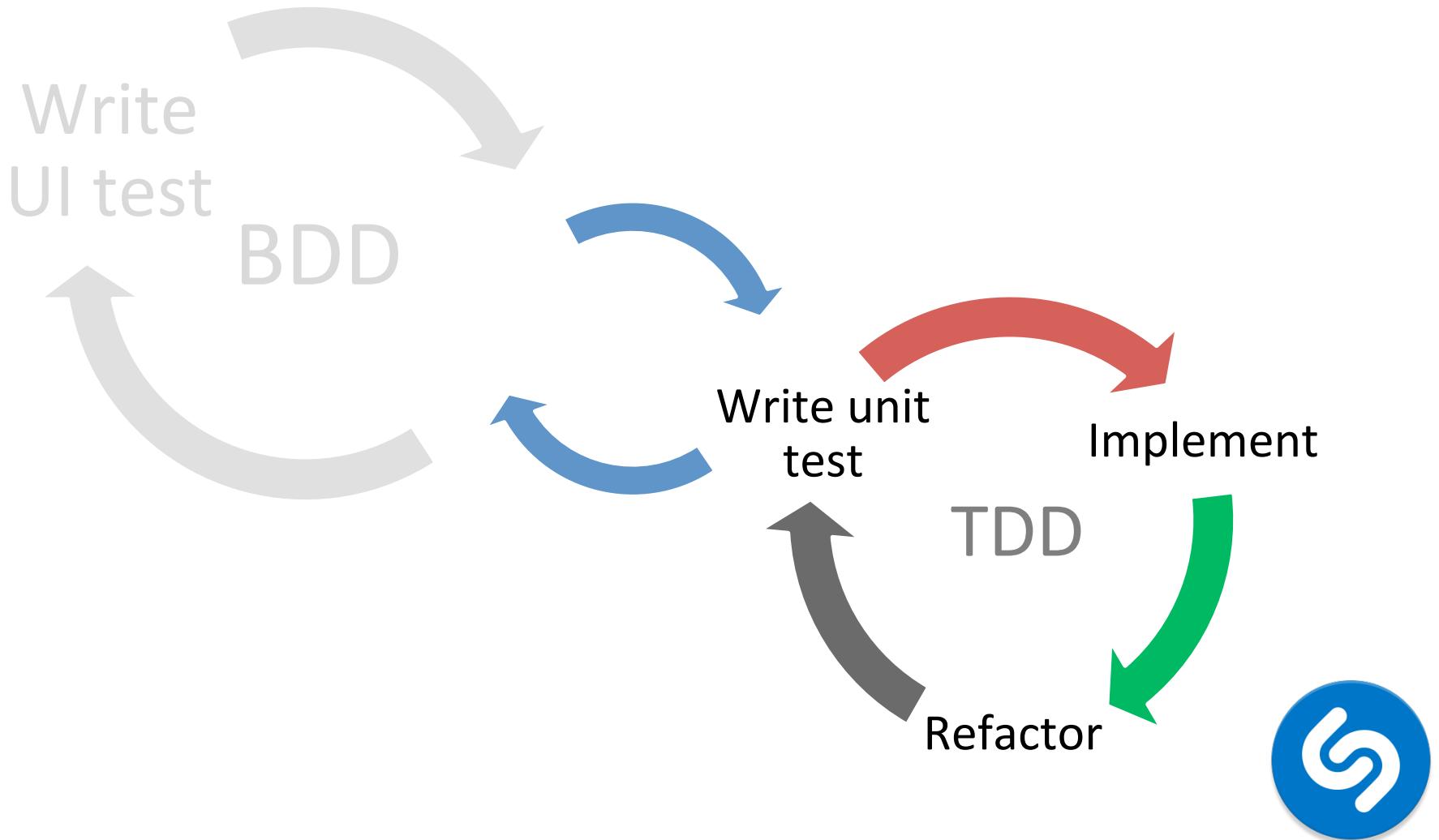
<http://github.com/shazam/gwen>

# Other libraries for Acceptance Tests

- Instrumentation
- JUnit 3
- Espresso & Robotium
- Fork
- HamMock Server



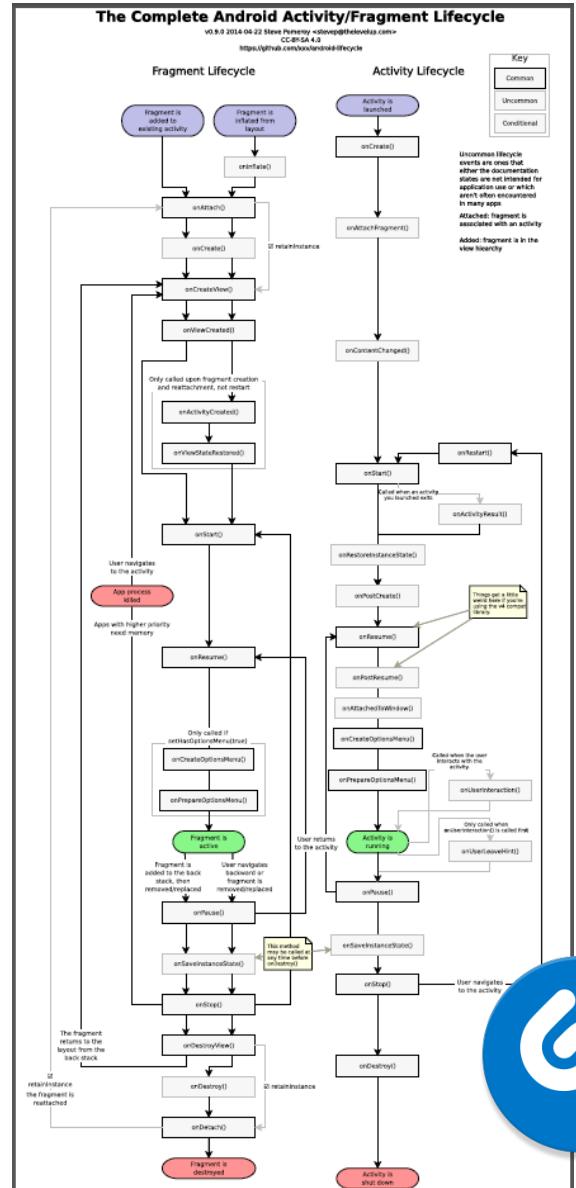
# The Unit tests cycle



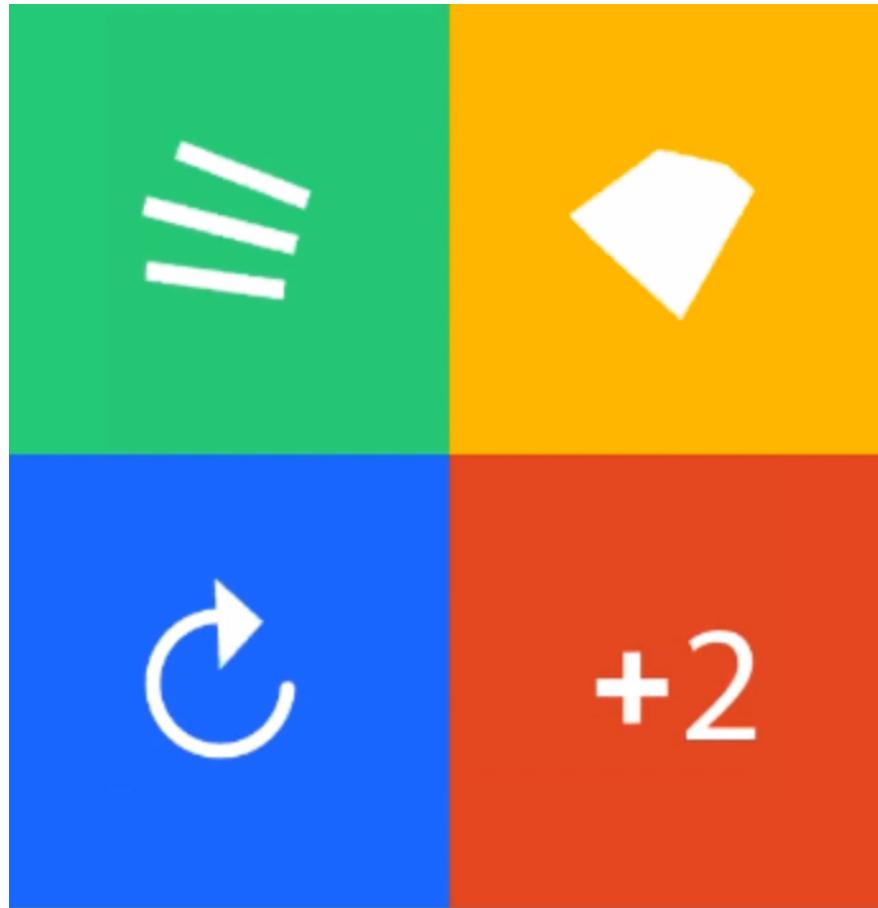
# What we don't test

## Activities & Fragments

<http://github.com/xxv/android-lifecycle>



# What we don't test

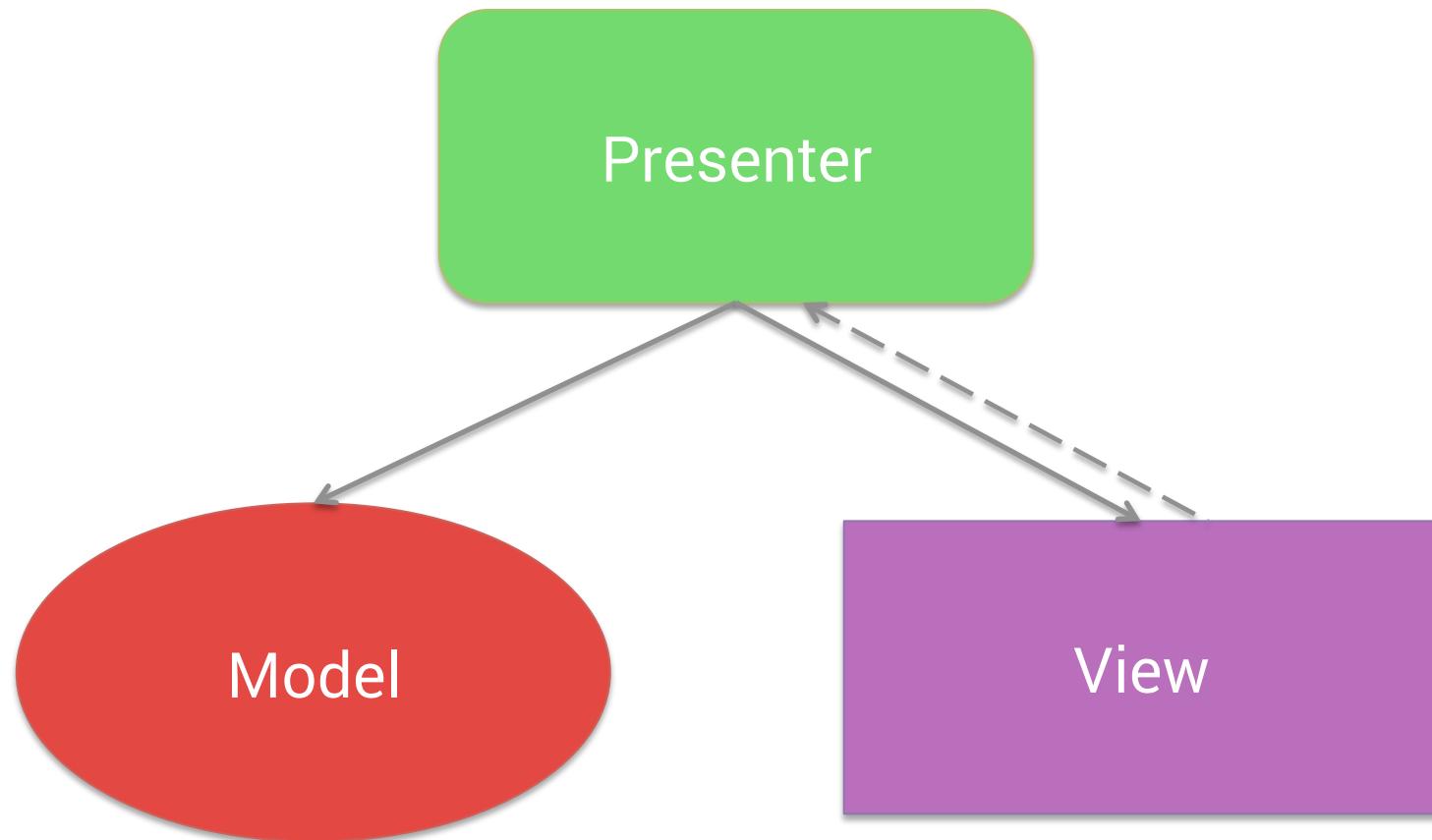


# What we do test

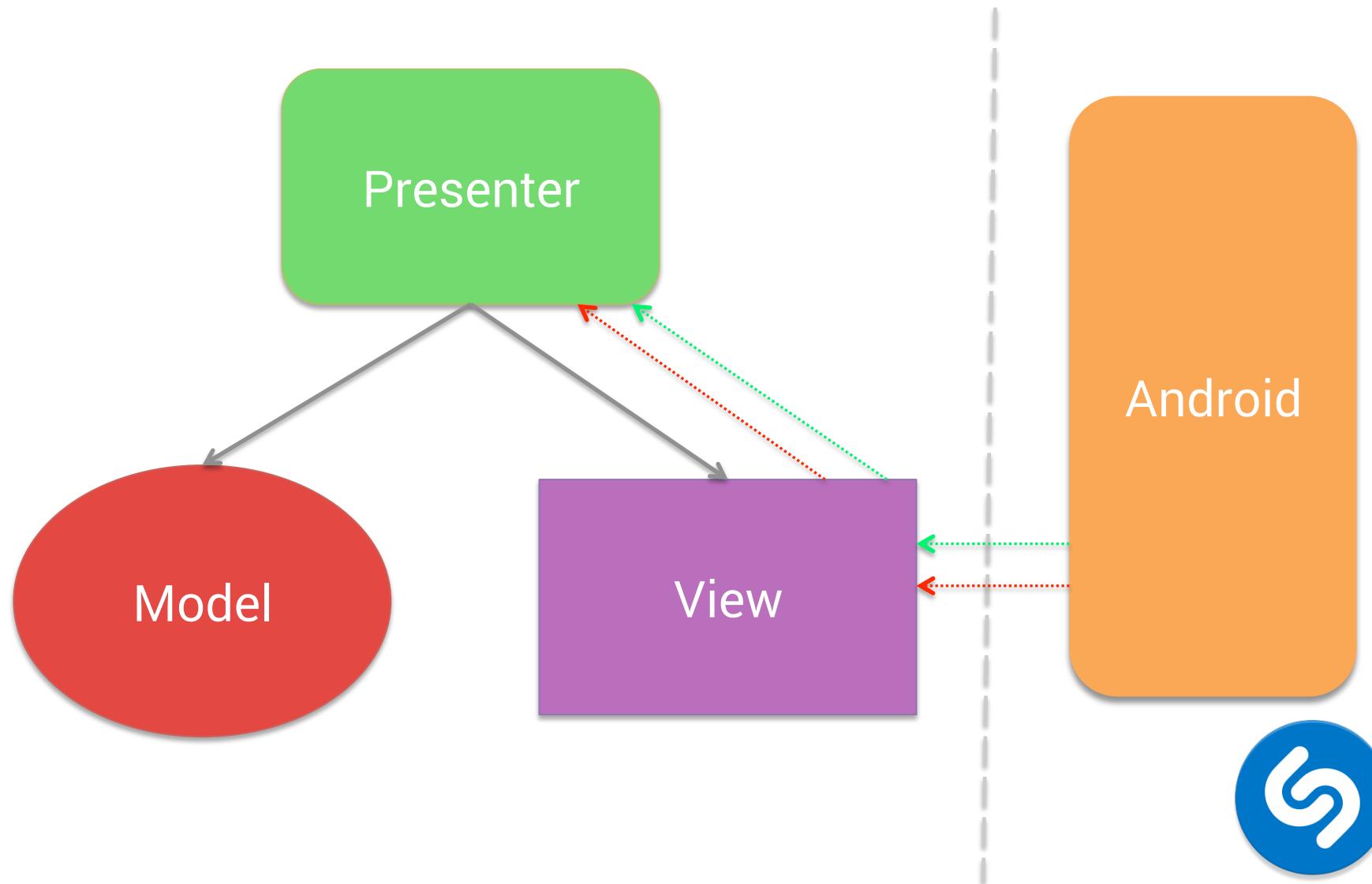
- Presentation logic
- Business logic
- That's it



# The MVP pattern



# The MVP pattern



# The MVP pattern

- Makes presentation logic testable
- No need to test “dummy” view
- Avoid Android dependencies

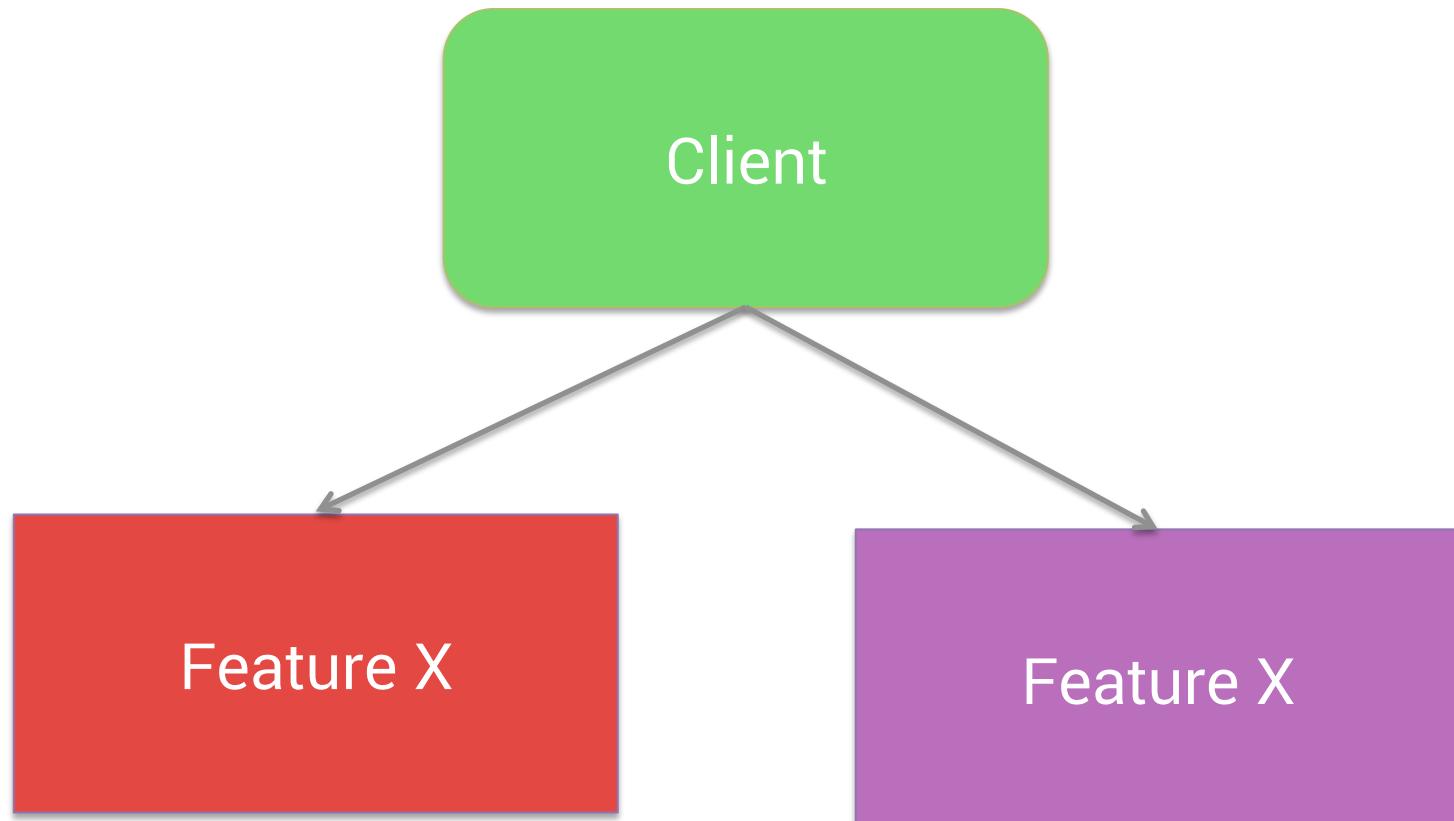


# Dependency Injection, Yourself

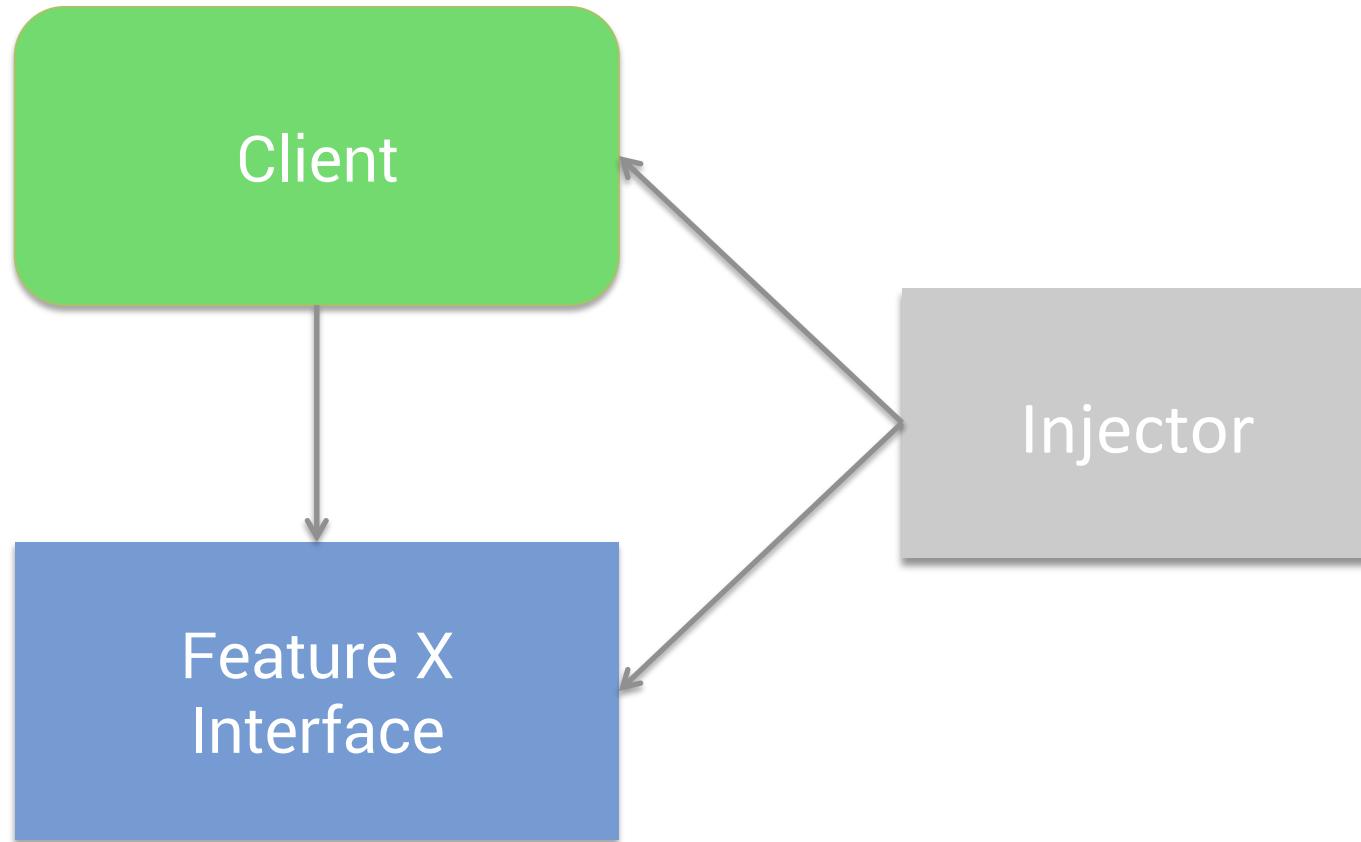
- Breaks hardcoded dependencies
- Behaviour vs Implementation
- Implementations for test & runtime



# Hardcoded dependencies



# Dependency Injection



# Model

```
public interface VenueRetriever {  
    void findClosestVenue(VenueFoundCallback callback);  
}  
  
public class NetworkVenueRetriever implements VenueRetriever {  
    public void findClosestVenue(VenueFoundCallback callback) {  
        // Some slow networking  
    }  
}  
  
public class LocalVenueRetriever implements VenueRetriever {  
    public void findClosestVenue(VenueFoundCallback callback) {  
        // DB look-up / caching layer, perhaps?  
    }  
}
```



# Activity

```
public class ResultActivity extends Activity implements ResultView {  
    private final VenueRetriever venueRetriever;  
    private ResultPresenter resultPresenter;  
  
    public ResultActivity() {  
        venueRetriever = venueRetriever();  
    }  
  
    public void onCreate(Bundle savedInstanceState) {  
        // TODO: Setup layouts & views  
        Result result = resultToDisplay();  
        resultPresenter=new ResultPresenter(this, venueRetriever, result);  
    }  
  
    public void onStart() {  
        resultPresenter.startPresenting();  
    }  
}
```



# Presenter

```
public class ResultPresenter {  
  
    public ResultPresenter(ResultView resultView, VenueRetriever  
        venueRetriever, Result result) {  
        this.resultView = resultView;  
        this.venueRetriever = venueRetriever;  
        this.result = result;  
    }  
  
    public void startPresenting() {  
        resultView.showResult(result);  
        venueRetriever.findClosestVenue(new VenueFoundCallback() {  
            public void venueFound(Venue venue) {  
                resultView.showCheckInPrompt(venue);  
            }  
        });  
    }  
}
```



# View

```
public interface ResultView {  
    void showResult(Result track);  
    void showCheckInPrompt(Venue venue);  
}
```



# Activity

```
public class ResultActivity extends Activity implements ResultView {  
  
    public void showResult(Result result) {  
        //TODO show the result screen & bind result data  
    }  
  
    public void showCheckInPrompt(Venue venue) {  
        //TODO bind the venue with check-in prompt view  
    }  
}
```



# Other Unit Test technologies

- JUnit 4
- Robolectric
  - `java.lang.RuntimeException: Stub!`
- Hamcrest
- JMock



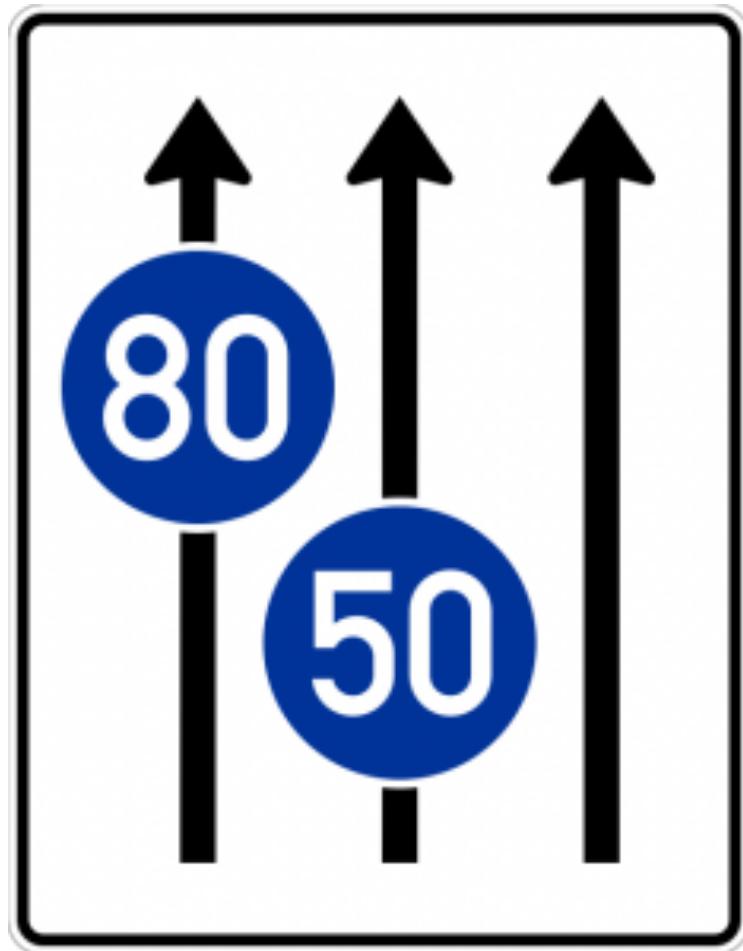
# Test execution



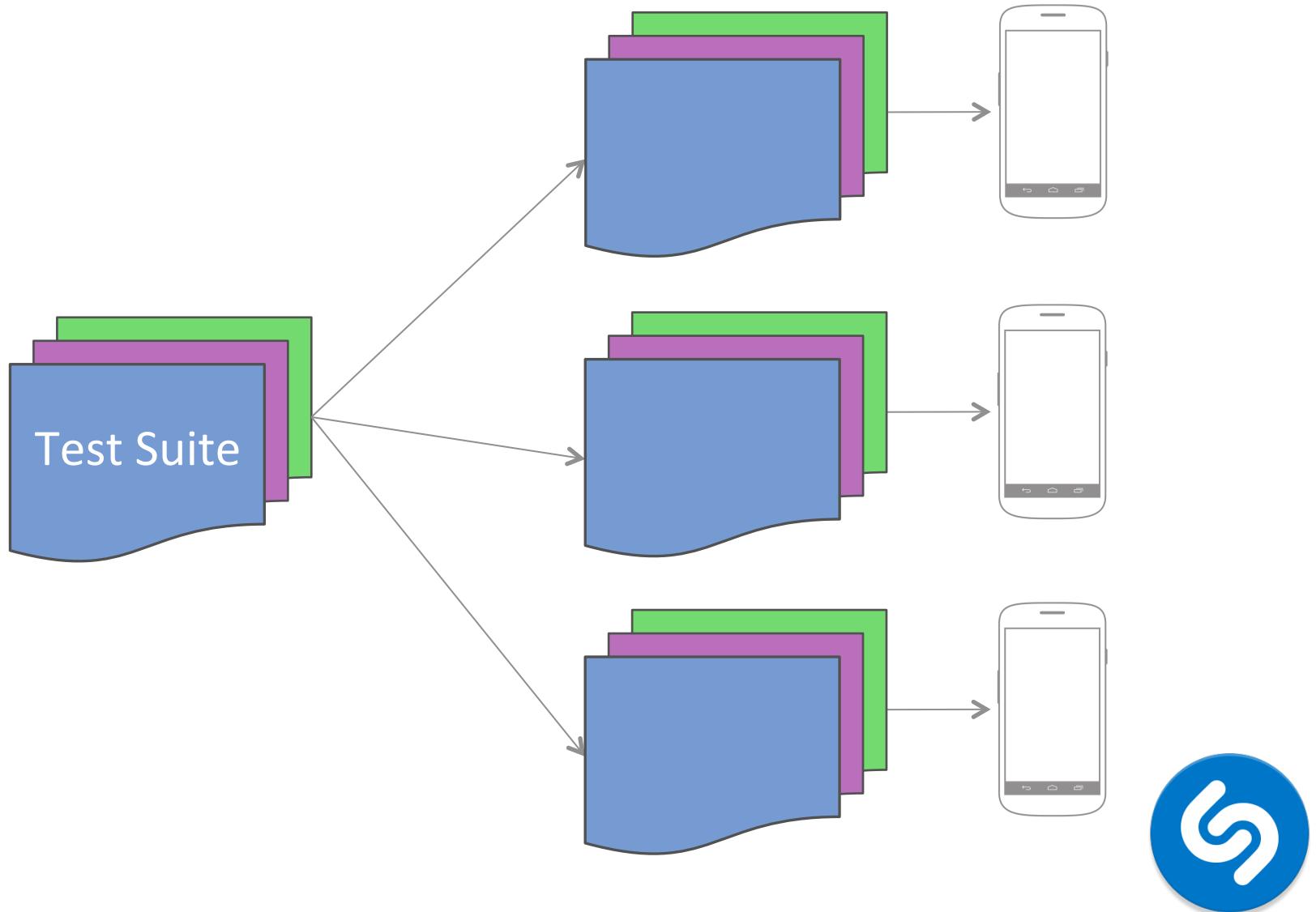
# Continuous Integration



# Speed (Hint: slow)



# Usual execution

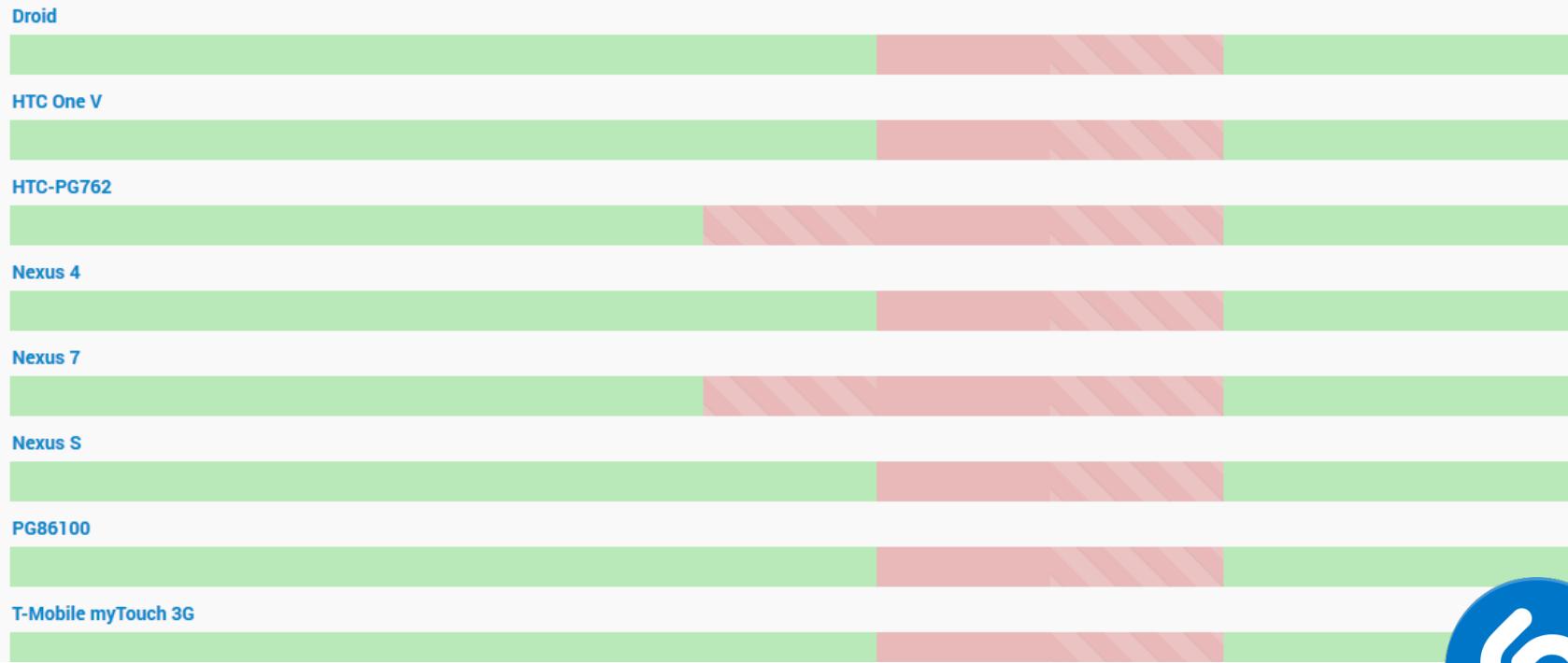


# Spoon



## Spoon Sample App

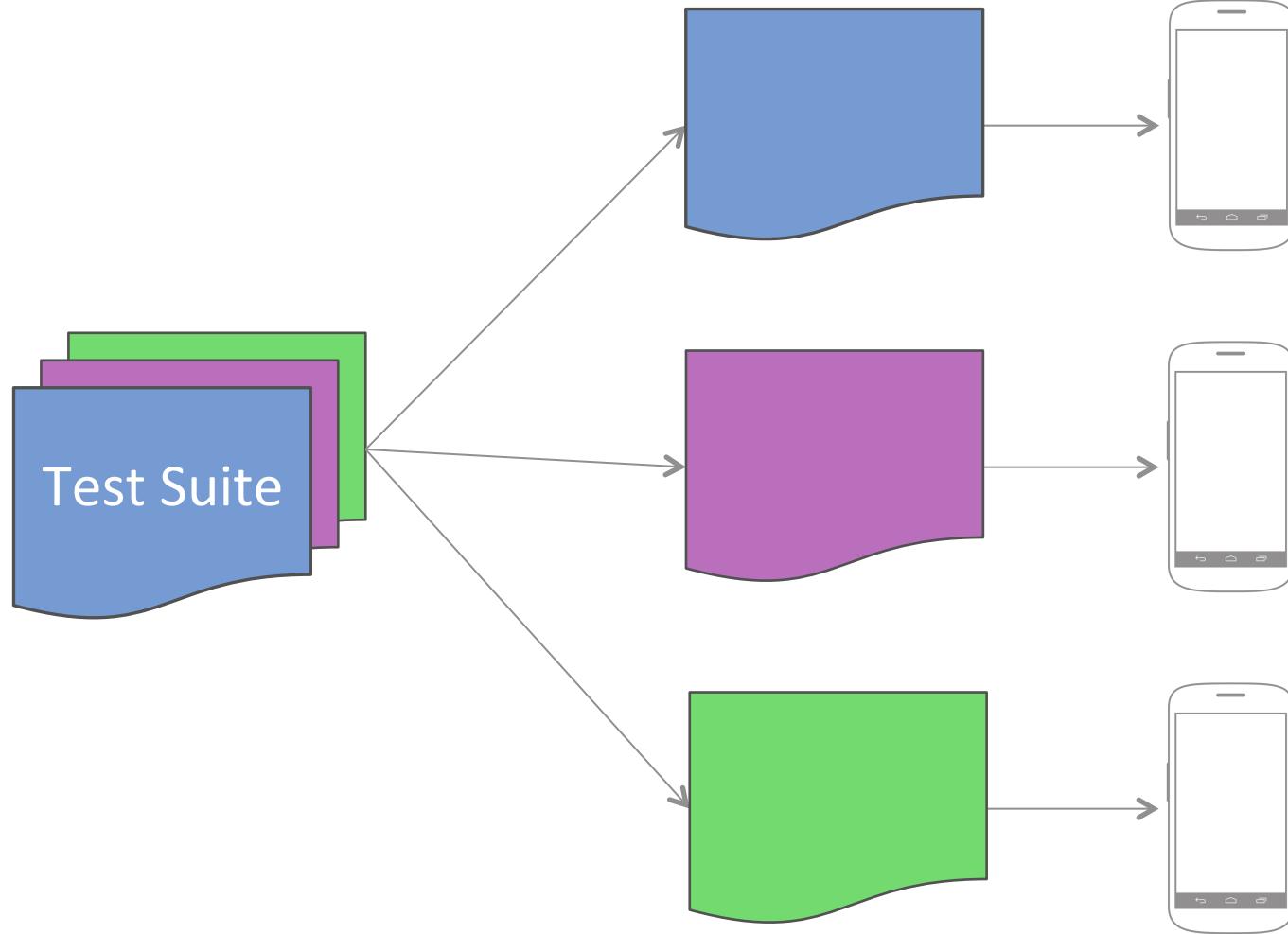
72 tests run across 8 devices with 54 passing and 18 failing in 2 minutes, 55 seconds at 2013-02-12 10:27 AM



<http://github.com/square/spoon>



# Fork



<http://github.com/shazam/fork>



# Example



```
:shazam-android:processGoogleFreeDebugTestJavaRes UP-TO-DATE  
:shazam-android:packageGoogleFreeDebugTest  
:shazam-android:assembleGoogleFreeDebugTest  
:shazam-android:forkGoogleFreeDebugTest  
Use -Dfork.tablet=(true/false) to configure pools depending on their manufacturer's 'tablet' flag (ro.build.characteristics)  
Use -Dfork.pool.POOL_NAME=(Serial','?)* to add devices with a given serial to a pool with given name,e.g. hdpi=01234567,abcdefg  
Use -Dfork.eachdevice=(true/false) to create a pool per device (a.k.a. Spoon mode). This is the default behaviour.  
Use -Dfork.excluded.serial=(Serial','?)* to exclude specific devices from running any tests  
Use -Dfork.report.title=Title to specify a title for the generated report  
Use -Dfork.report.subtitle=Subtile to specify a subtitle for the generated report  
Attempt #1 failed: Could not clear logcat on device: 192.168.56.101:5555  
Success after: Could not clear logcat on device: 192.168.56.101:5555  
Attempt #1 failed: Error while installing com.shazam.android on 192.168.56.104:5555  
Success after: Error while installing com.shazam.android on 192.168.56.104:5555  
> Building 98% > :shazam-android:forkGoogleFreeDebugTest
```

```
No Filters  
Gradle Console  
compiler that did not target the modern .class file format. The  
solution is to recompile the class from source, using an up-to-  
and without specifying any "-target" type options. The conseque  
this warning is that reflective operations on this class will i  
indicate that it is *not* an inner class.  
:shazam-android:processGoogleFreeDebugTestJavaRes UP-TO-DATE  
:shazam-android:packageGoogleFreeDebugTest  
:shazam-android:assembleGoogleFreeDebugTest  
:shazam-android:forkGoogleFreeDebugTest  
Use -Dfork.tablet=(true/false) to configure pools depending on their manufacturer's 'tablet' flag (ro.build.characteristics)  
Use -Dfork.pool.POOL_NAME=(Serial','?)* to add devices with a given serial to a pool with given name,e.g. hdpi=01234567,abcdefg  
Use -Dfork.eachdevice=(true/false) to create a pool per device (a.k.a. Spoon mode). This is the default behaviour.  
Use -Dfork.excluded.serial=(Serial','?)* to exclude specific devices from running any tests  
Use -Dfork.report.title=Title to specify a title for the generated report  
Use -Dfork.report.subtitle=Subtile to specify a subtitle for the generated report  
BUILD SUCCESSFUL  
Total time: 1 mins 58.448 secs  
Event Log Gradle Console
```



# Fork

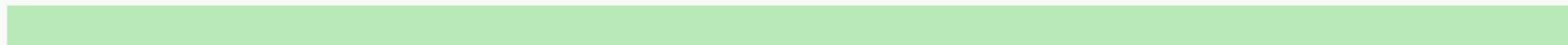


## Shazam on Android

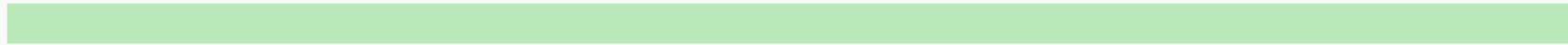
trunk



Tablet Sw720 Up



Phone Sw0 719



<http://github.com/shazam/fork>



# Fork



- Pooled execution
- Infinitely scalable
- Current setup 1 test / 2 seconds

<http://github.com/shazam/fork>



# Flakiness monitor

com.shazam.android.acceptancetests.tagdetails.my.MusicMyTagDetailsTest	testCanDeleteTag	1												
m.shazam.android.acceptancetests.newsfeed.buy.NewsFeedHavingStoresTest	testCanLaunchAmazonStoreFromAnnouncementCard	1												
m.shazam.android.acceptancetests.newsfeed.buy.NewsFeedHavingStoresTest	testCanLaunchAmazonStoreFromTagCard	1												
com.shazam.android.acceptancetests.tagging.LyricsOnLegacyMusicDetailsTest	testCanSeeTheLyricPlay	1												
m.shazam.android.acceptancetests.musicdetails.MusicDetailsWithoutRdioTest	testCannotAddToAnRdioPlaylist	1												
id.acceptancetests.musicdetails.modules.MusicDetailsModuleDiscographyTest	testClickOnAlbumShowsAlbumTracksPage	1												
com.shazam.android.acceptancetests.tagdetails.my.MusicMyTagDetailsTest	testDeleteTagSendsBeaconToServerWithTagVersion	2												
	com.shazam.android.acceptancetests.settings.RdioSettingsTest	1												
com.shazam.android.acceptancetests.tagdetails.my.MusicMyTagDetailsTest	testDisplaysAmazonBuyMp3AddOn	1												
com.shazam.android.acceptancetests.musicdetails.interactiveinfo.buy.	MusicDetailsHavingJustGoogleStoreTest	1												
com.shazam.android.acceptancetests.musicdetails.MusicDetailsTest	testDisplaysGenericBuyButton	1												
com.shazam.android.acceptancetests.musicdetails.MusicDetailsTest	testDisplaysMusicTrack	1												
com.shazam.android.acceptancetests.tagdetails.my.MusicMyTagDetailsTest	testDisplaysPreviewAddOn	1												
acceptancetests.musicdetails.modules.MusicDetailsModuleRecommendationsTest	recommendationsModuleWithCorrectRecommendationsCount	1												
com.shazam.android.acceptancetests.tagdetails.my.MusicMyTagDetailsTest	testDisplaysShareButton	1												
com.shazam.android.acceptancetests.newsfeed.NewsFeedTagCardTest	testDisplaysTimestampWithEventForNewsFeedTagCard	1												
com.shazam.android.acceptancetests.musicdetails.interactiveinfo.buy.	MusicDetailsHavingJustAmazonStoreTest	2												
com.shazam.android.acceptancetests.musicdetails.interactiveinfo.buy.	testDoesNotDisplayAnyStoreIfAmazonFailsToLoad	2												
android.acceptancetests.musicdetails.details.GenericDetailsInteractiveInfoTest	testForPreview	1												
android.acceptancetests.musicdetails.details.GenericDetailsInteractiveInfoTest	testForStores	1												
i.shazam.android.acceptancetests.newsfeed.NewsFeedAnnouncementCardTest	testLaunchesFullscreenWebPageWhenUserClicksOnCard	1												
com.shazam.android.acceptancetests.DeepLinkTest	testLaunchesTagDetailsForGooglePlusDeepLink	3												
com.shazam.android.acceptancetests.shweb.ShWebTest	testPassesContextToShWebInTag	1												



# ADB Remote

home menu/unlock back power Type here Autoscale:

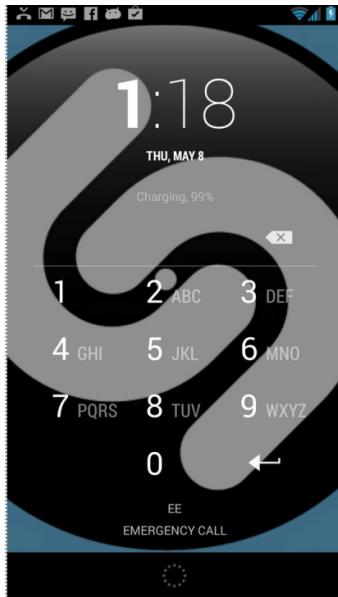
To set asset (with quotes): adb shell "echo sdcard.asset=PHN-1234 > /sdcard/asset"

adb shell   
Auto refresh in: \*\*\*\*\* (17 more), scale 1  
adb reboot

If the device's clock shows the wrong time and it's unresponsive, the screen is probably powered off.

refresh 0deg  
refresh 270deg refresh 90deg  
refresh 180deg

TapSwipe  
Experimental: Multitouch  
Experimental: Monkeyrunner  
Experimental: Old devices  
Experimental: Keyboard  
Settings



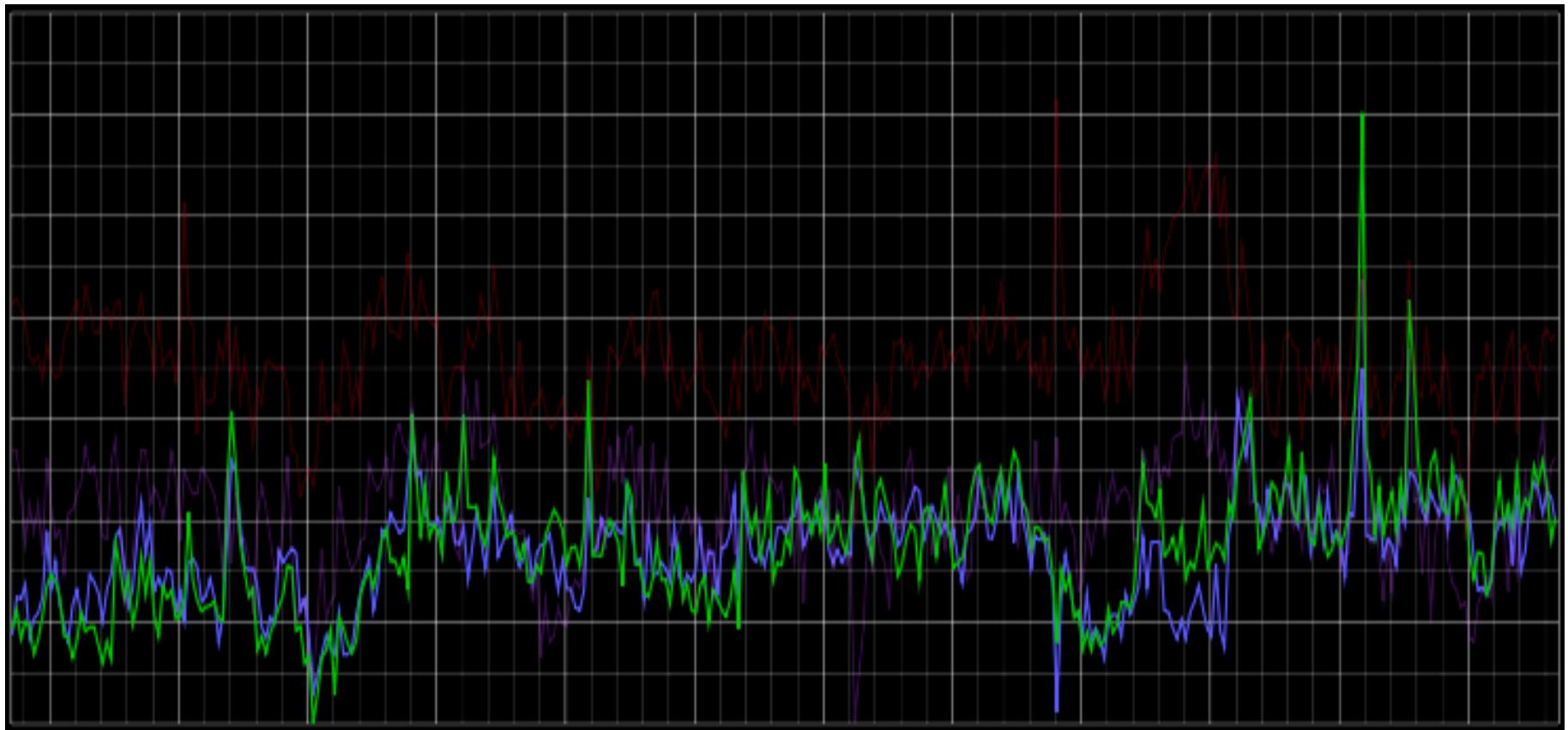
<http://github.com/sleekweasel/CgiAdbRemote>



# If all else fails



# If all else fails



# If all else fails



# Summary

- Testing is easier than you may think
- Practice, practice, practice
- Toolset is limited but getting better
- Ship it!



# Questions?

