











### How to improve CI/CD pipeline for mobile





## Niranjani Manoharan



## **Stakeholders involved**

• QA/SDET



• Engineers/Developers



## • Build/Release



### What were the blockers?

- Break down features into smaller chunks
- Reduce build time
- Test frequently
- Automate as much as possible
- Project management/Scrum process/JIRA workflow





- Learn how containers can simplify the android build process.
- Improvements to test automation.
  - Get more insights into which open source tool works better to provide parallelized test runs on mobile platform.
- Caveats of different CI systems.





**P** Challenge: How the android team reduced build time?

#### Solution:

- Gradle vs Maven (on android)
- Build variants
- Dependencies
- CI system (Jenkins vs Buildkite)



### Gradle

- Task dependencies
- Multi module builds run in parallel
- Incremental changes/build cache

#### dependencies { classpath rootProject.ext.googleServicesClasspath productFlavors { flavorDimensions "environment" releaseVer { dimension "environment" minSdkVersion 23 buildTypes {

```
}
testOptions {
    integrationTests {
        . . .
```

. . .

buildscript {

#### Maven snippet:

- Linear model of phases
- Multi module builds run in parallel

<?xml version="1.0" encoding="UTF-8"?> . . . <modelVersion>4.0.0</modelVersion> . . . <dependencies> <dependency> <groupId>com.google.android</groupId> <artifactId>android</artifactId> <version>2.2.1</version> <scope>provided</scope> </dependency> </dependencies <build> <finalName>\${project.artifactId}/finalName> <sourceDirectory>src</sourceDirectory> <plugins> <!-- platform or api levl (api level 4 = platform 1.6) → . . . </build> </project>



- Builds pass locally but fails on CI
- Multiple instances of jenkins getting blocked
- Difficult to reproduce what's running on CI on my local machine

## **Advantages of Buildkite**

- Easy to add new pipelines
- Builds are faster
- Simple to debug failures



#### **Metrics**

- Jenkins/Maven: ~45 mins
- BuildKite/Gradle: ~20 mins





#### Building the CI/CD pipeline...



Challenge: Why were our android tests taking longer to run?

#### Solution:

- Types of validations: UI + backend
- Number of tests being run
- Categories of test suites
- <u>Where</u> do these tests run?
- Types of devices used for test run
  - Tablet(s)
  - o Phone(s)



## **Solution:**

- <u>AWS</u> device farm vs <u>Firebase</u> labs
- Custom device lab
  - Single device
  - Multiple devices
  - Phone/Tablet

#### **AWS Device Farm**





AWS Device Farm Remote Access



## **Firebase With Flank**



#### Snapshot of AWS device farm test run

<b>\</b>	Device Farm	Project: android test automation $$	Run: jenkins-Android Release					
				Learn more about unlimited testing	assumed-role/engineer/niranjani 🗸	Support ~		
igenkins-Android Release Tests-846 Scheduled at: Feb 20, 2018 11:29:10 AM								
€ 10 FAILED © 85 PAISED								
Unique problems Devices Screenshots Parsing result								
Unique problems								
10 Unique failures found								
testNotificationsToProfile failed: android.support.test.espresso.NoMatchingViewException: No views in hierarchy found matching: (with id: com								
	testInteractionWithPins failed: android.support.test.espresso.NoMatchingViewException: No views in hierarchy found matching: (with id: com.pi							
	• testSearchPopularCategory failed: android.support.test.espresso.NoMatchingViewException: No views in hierarchy found matching: (with id: co							
	• testSearchByPinFilter failed: android.support.test.espresso.NoMatchingViewException: No views in hierarchy found matching: (with id: com.pint							

#### Snapshot of Firebase Test Labs

붣 Firebase	Android Automation Lab 🔫	Go to docs 🌲 🤹
🔒 Project Overview 🌣	Test Lab > Pinterest OTA > Matrix #622156 ← Instrumentation test, Low-resolu	ition MDPI phone, Virtual, API Level 25 💦 👩
<b>DEVELOP</b> Authentication, Database, Storage, H		
STABILITY	🛕 Failed 💼 3/28/18, 3:33 PM 😳 50 sec 🚫 Portrait	English     VIEW SOURCE FILES
🐛 Crashlytics	TEST CASES LOGS VIDEOS PERFORMANCE	
Performance		
🔓 Test Lab	Failed Passed	Skipped
	1 0	0
ANALYTICS		
Dashboard		
Levents		
Audiences		
=, Attribution		
Blaze MODIFY Pay as you go		



# Jenkins/AWS Device Farm/300 tests: ~150 mins

#### BuildKite/ Firebase Labs/300 tests:

<mark>~ 20 mins</mark>



## **Upcoming topics**

- Which test framework did we use?
- How we structured our tests?
- How we handled test runs on different

devices?

• Finally, how did we parallelize test runs?

# **P** Challenge: Which android test framework did we use?

#### **Solution:**

Espresso	Appium
Well-suited for writing white box-style automated tests	Follows a black box approach as it directly uses the app's apk
Tests will break if anything changes in the app	Tests reside outside of the app's codebase.
Native to android applications.	Cross platform testing framework.

#### How does Espresso work?



#### Android Test Architecture



Challenge: Why didn't we mock request/responses?

- Replace manual testing
- Mocking ~ unit tests
- End to end scenarios mimic user behavior
- Helpful to find integration bugs

#### Test Architecture with TAP(Test Account Pool)



#### **Metrics**

#### Reduced test flakiness by 50%.



#### Espresso code snippet:Page Object Model implementation

#### **@Test**

```
public void testContextualMenuSaveButton() {
```

// setup:

openView(Navigate.View.HOME, Collections.emptyList());

// when:

```
HomefeedSurfaceActions.openContextMenuForPinWithIndex0();
HomefeedSurfaceActions.clickSaveInContextMenu();
flushEvents();
```

```
// then:
Assert.assertTrue(...);
}
```



#### **Test Utils**



#### Without design pattern

```
@Test
public void testVideoAutoplay() {
    // setup:
    ...
    // when:
    watchVideoInCloseupFrom0(VideoConstants.VIDEO_TIME +
```

VideoConstants.LOG\_TIME);

#### // then:

. . .

#### Audit Data class

public static Config createVideoConfig(String userId,...) {

```
List<Filter> filters = Arrays.asList(
    new Filter<>(Constants.PATH_EVENT_TYPE,
        Constants.COMPARISON_EQUALS,
        Constants.VIDEO_PLAYBACK_COMPLETION)
```

);

. . .

List<QuantityCheck> quantityChecks = ...;

return new Config(userId,filters, quantityChecks, ... Constants.CHECKER NAME, checkerSettings);

#### **Issues faced**

- Audit Data Class ~ 3000 lines of code
- Not complying to good programming practices
- UI Test + backend validation ~ 2-3 hours => 4

tests per day

#### With Builder pattern

```
public class VideoAuditEventBuilder {
    public interface ViewType {
        String FEED = "FEED";
    ...
    }
    public static class VideoEventsName {
        public static final String VIDEO 1
```

public static final String VIDEO\_PLAYBACK\_STATE\_PLAYING = "PLAYING";
public static final String VIDEO\_PLAYBACK\_STATE\_PAUSED = "PAUSED";
...

```
public VideoAuditEventBuilder
addPlaybackStatePlayingInterval(...,
       int min, int max,
       List<Filter> additionalFilters,
      . . . )
   return this.addPlayingWithIntervalAndFilters(startTime,
endTime,
           VideoEventsProperty. PATH,
           min, max,
           additionalFilters,
          ...);
```

#### In the test method

// verify:

this.assertEvent(new VideoAuditEventBuilder(...,

this.getUserId())

.<mark>addPlaybackStatePlayingInterval</mark>(...)

.addPlayingWithIntervalAndFilters(startTime, endTime, VideoEventsProperty.PATH,

```
...)
.build());
```

#### **Metrics**

## Test + backend validation = 2-3 hours => 4 tests per



## With design patterns:

#### Test + backend validation < 1 hour => 8+ tests per



- Organizing libraries
  - ViewActionUtils
  - ViewMatcherUtils



- What design patterns did we use?
  - Builder pattern the same construction process can create different representations
  - Page Object Model helps with ease of maintenance

 Test flakiness will occur if test framework & design patterns are not used effectively alongwith ReTry rules.



- Resolution is different
- Types of click listeners
  - Click vs long press

#### **Solution:**

Try increasing the long press delay values of your test devices/emulators under Setting -> Accessibility -> Touch & hold delay



#### Workaround

If the click takes longer than the 'long press' duration (which is possible) the provided \* rollback action is invoked on the view and a click is attempted again. \*

public static ViewAction click(ViewAction rollbackAction) {
 checkNotNull(rollbackAction);
 return new GeneralClickAction(Tap.SINGLE, GeneralLocation.CENTER, Press.FINGER,
 rollbackAction);

# **P** Challenge: Why did we choose Flank for parallelizing test runs?

Flank	Spoon
Flank is a open source tool to <mark>run test suites in</mark> <mark>parallel</mark> .	Spoon <mark>runs tests</mark> on multiple devices <mark>simultaneously</mark>
To runs tests with Flank you will need the app and test apk's. Stores test results in a folder named: results.	Using the application APK and instrumentation APK, a static HTML summary is generated with detailed information about each device and test.

#### **Metrics**

## Jenkins/AWS Device Farm/ Spoon for 300 tests: ~ 150 mins

### BuildKite/ Firebase Labs/ Flank for 300 tests:



#### Building the CI/CD pipeline...



## **P** How containers add value to the pipeline? Solution:

- Pre-packaged containers have all the dependencies loaded and can be deployed through the docker.yml file.
- Customize pipeline with .yml scripts.
- Use a CI system that can delineate the different steps involved in the build/test/deploy process.

#### How docker ties them all together!



#### Code snippet of docker file - how to package dependencies of a project

```
# Update apt-get
RUN rm -rf /var/lib/apt/lists/*
RUN apt-get update
RUN apt-get dist-upgrade -y
```

```
# Installing packages
RUN apt-get install -y \
    ca-certificates-java \
    git \
    software-properties-common \
    unzip \
    wget \
    zip \
    --no-install-recommends
```

. . .

#### **CI/CD** pipeline execution



 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

## pipeline.yml

cat .buildkite/steps/root.yml \
 .buildkite/steps/docker.yml \
 .buildkite/steps/build\_ota.yml \
 ...
 .buildkite/steps/prod\_tests.yml \
 .buildkite/steps/wait\_continue\_on\_failure.yml \
 .buildkite/steps/upload\_test\_results.yml \
 .buildkite/steps/wait\_continue\_on\_failure.yml \
 .buildkite/steps/post\_build.yml | buildkite-agent pipeline upload



#### **Summary**



- How dockers simplified our build process.
- Caveats of Jenkins over Buildkite.
- Automation best practices.
- Got more insights into which open source tool works better to provide parallelized test runs on the android platform.
- How our overall metrics for CI/CD pipeline improved



#### **#FALLONTONIGHT**

# THANK YOU!