

iOS testing from top to bottom

Agile & Automation Days
Kraków, 2017

allegro



Ewa Ludwiczak

Senior Software Tester

ewa.ludwiczak@allegrogroup.com



@ewabielskapoz



Kamil Pyć

Senior iOS Developer

kamil.pyc@allegrogroup.com

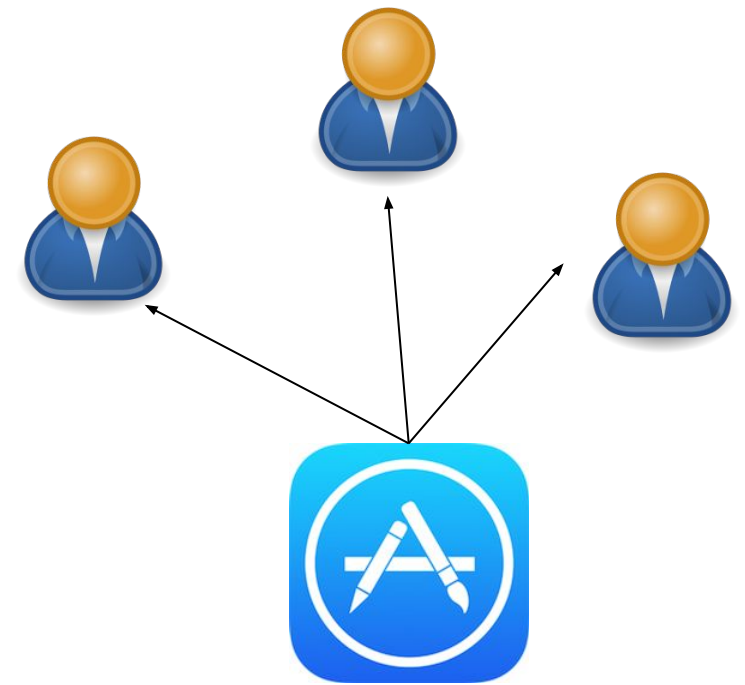
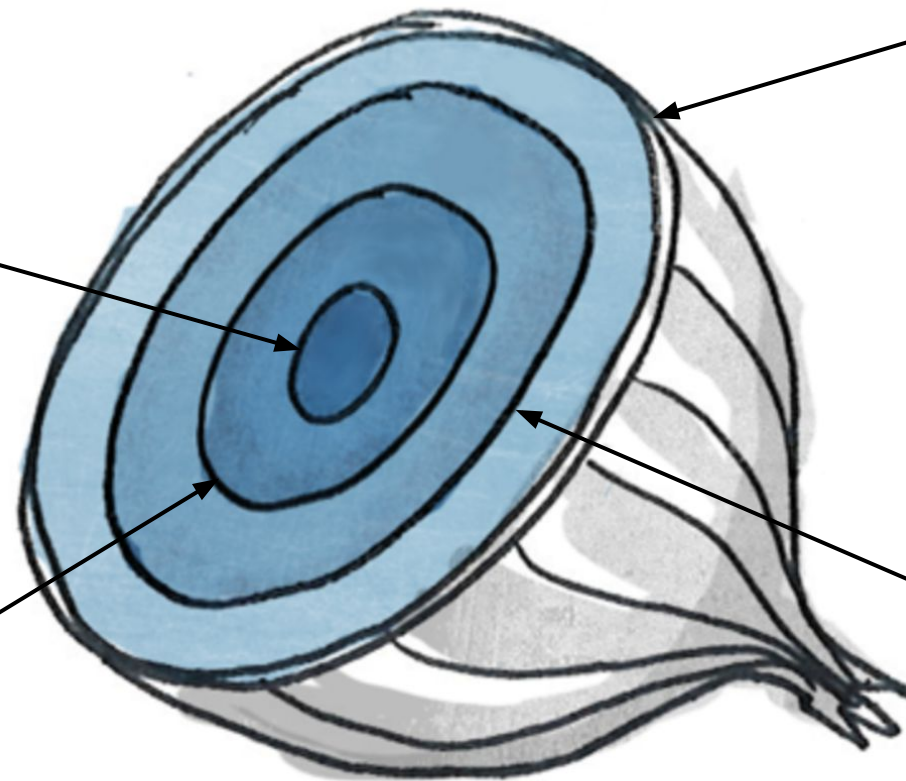


@KamilPyc

“Ogres are like onions”



Tests are like onions

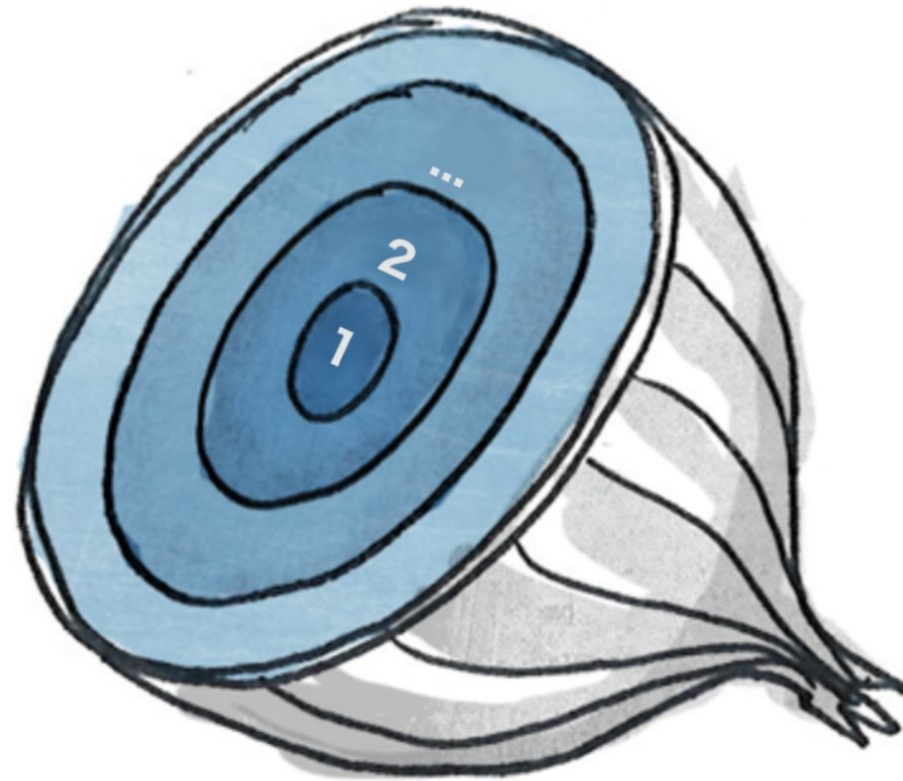


```
1 func sayHelloWorld() -> String {  
2     return "hello, world"  
3 }
```



Tests are like onions

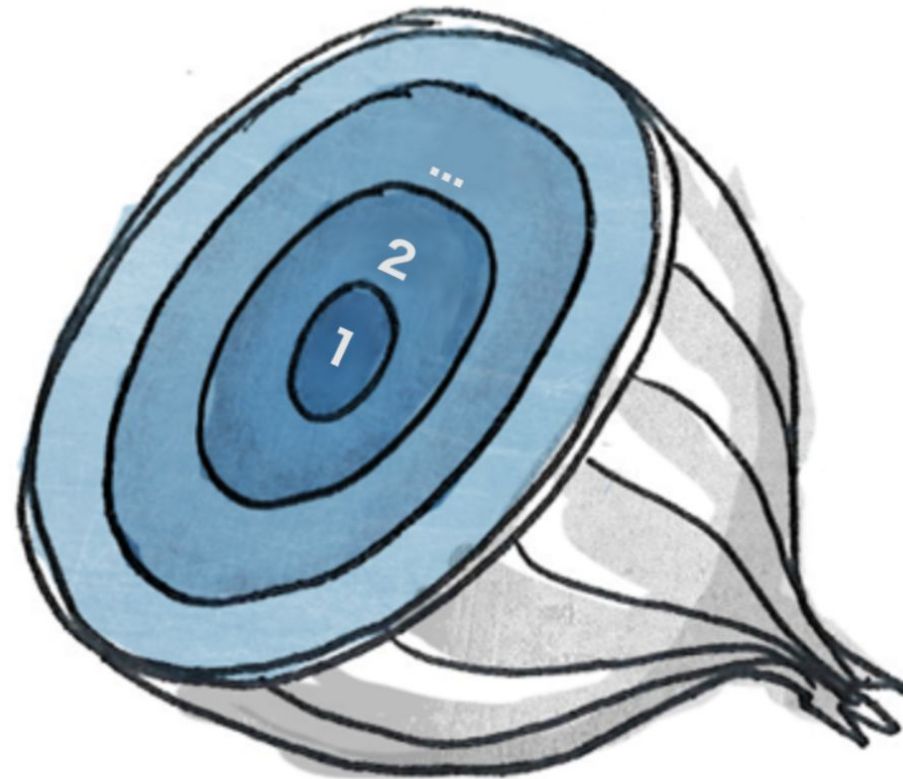
1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 1: Crash on production caused by force unwrapping

```
func printFullName(first: String, middle: String?, last: String) {  
    print(first + " " + middle! + " " + last)  
}
```

```
printFullName(first: "Ewa", middle: nil, last: "Ludwiczak")
```



Problem 1: Crash on production caused by force unwrapping

Solution: SwiftLint




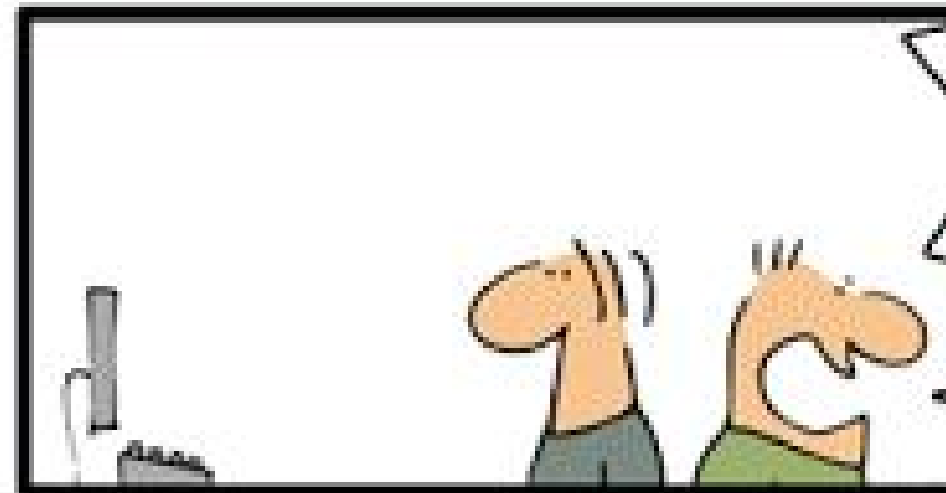
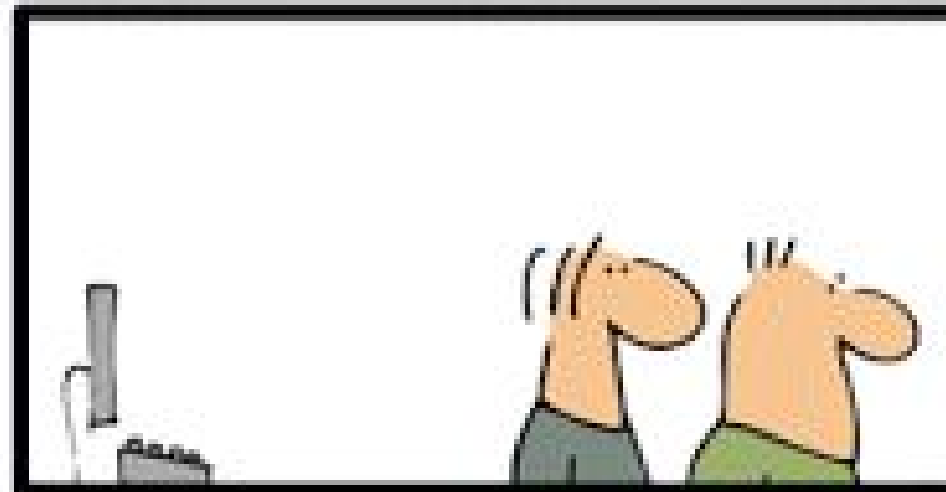
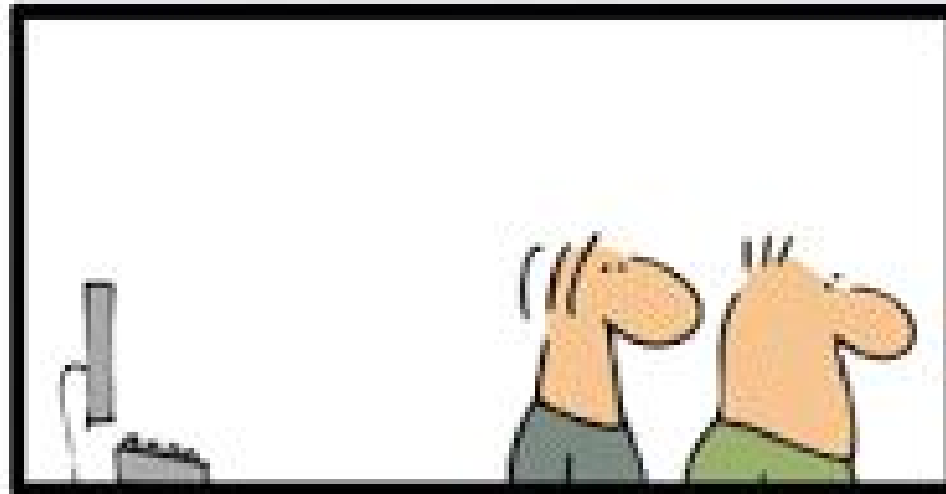
1. enforce Swift style and conventions
2. detect code smells
3. apply custom rules

<https://github.com/realm/SwiftLint>

Problem 1: Crash on production caused by force unwrapping

Solution: SwiftLint

```
func printFullName(first: String, middle: String?, last: String) {  
    print(first + " " + middle! + " " + last)  Force Unwrapping Violation: Fo  
}
```

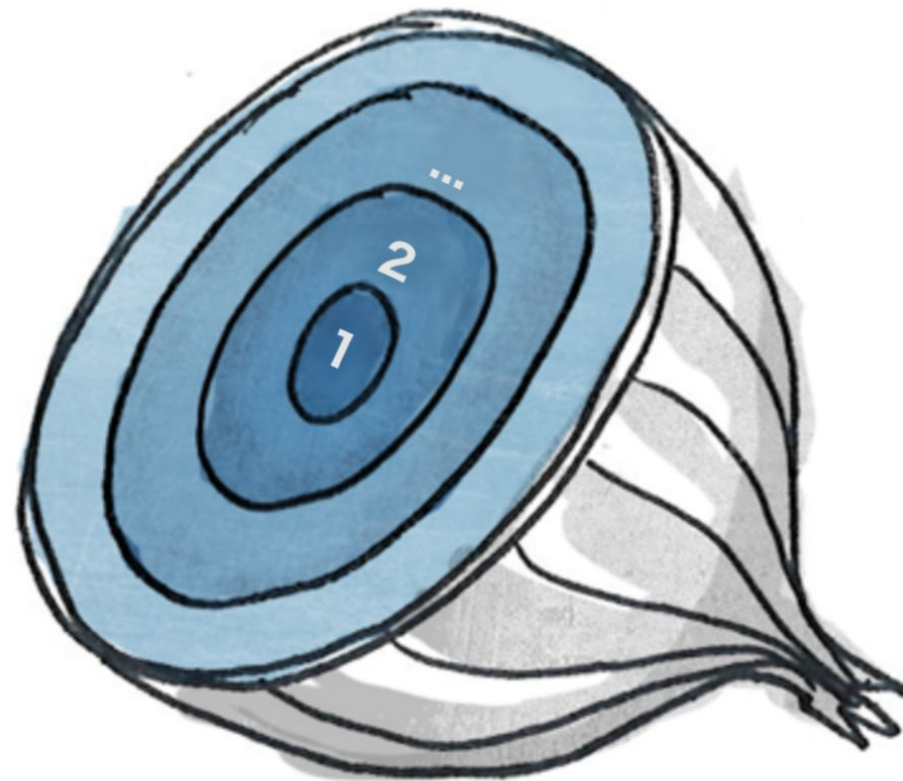
good to make

DON'T LOOK
AT THE
SCREEN!!!!

HOW TO DEBUG HEISENBUGS

Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 2: Evergreen unit tests

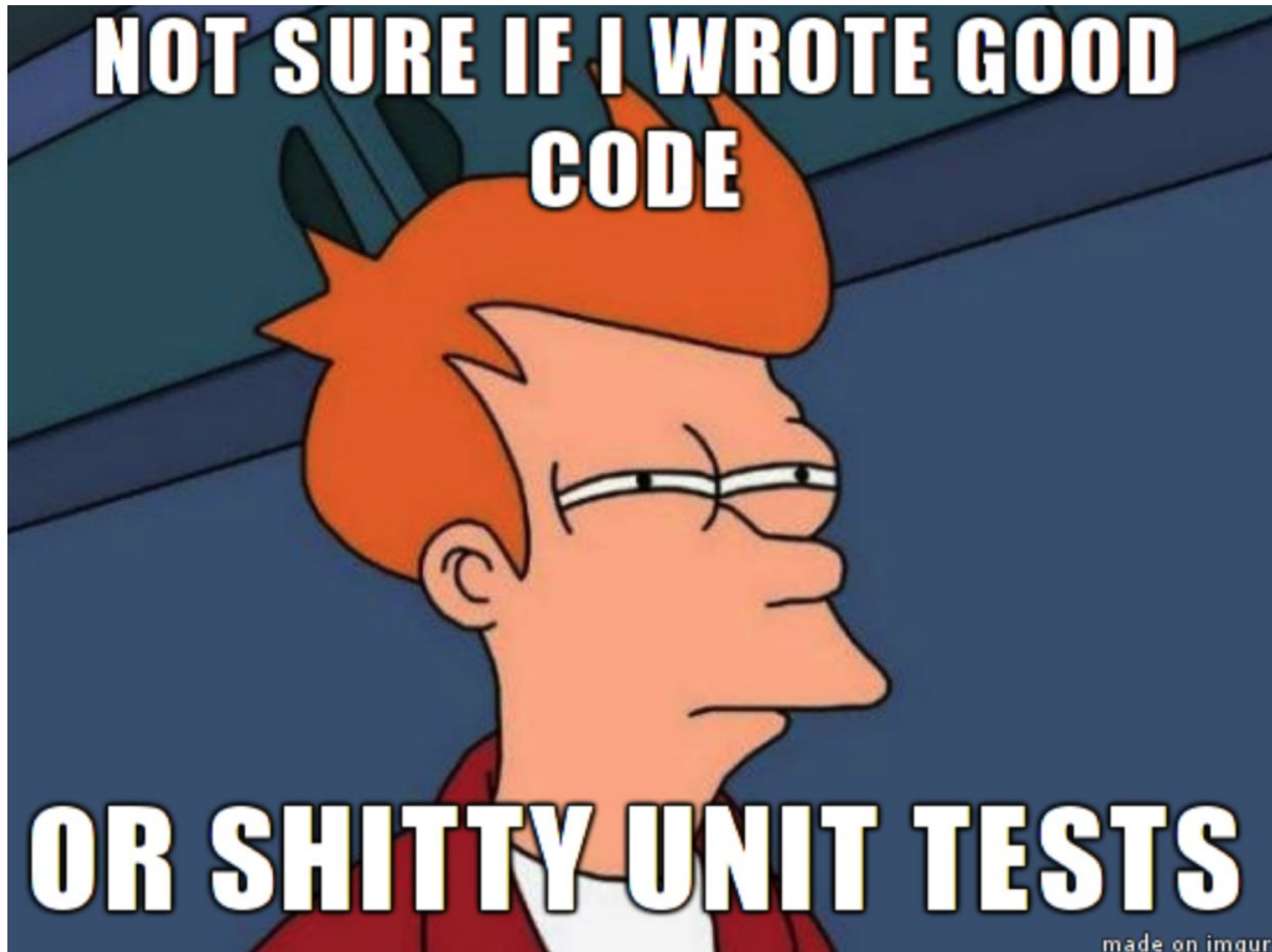
```
func testAddition() {  
    // Arrange  
    let a = 1  
    let b = 2  
  
    // Act  
    let sum = 3  
  
    // Assert  
    XCTAssertTrue(sum == 3)  
}
```


Problem 2: Evergreen unit tests

Solution: Mutation testing

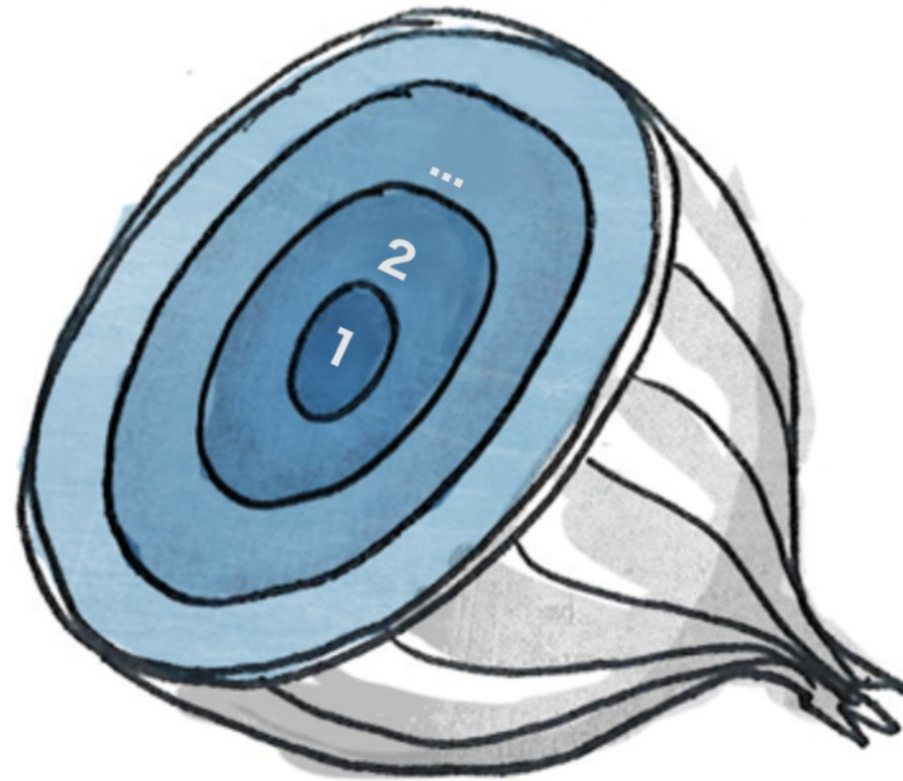


1. modify your source code
2. run tests
3. check if tests failed



Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



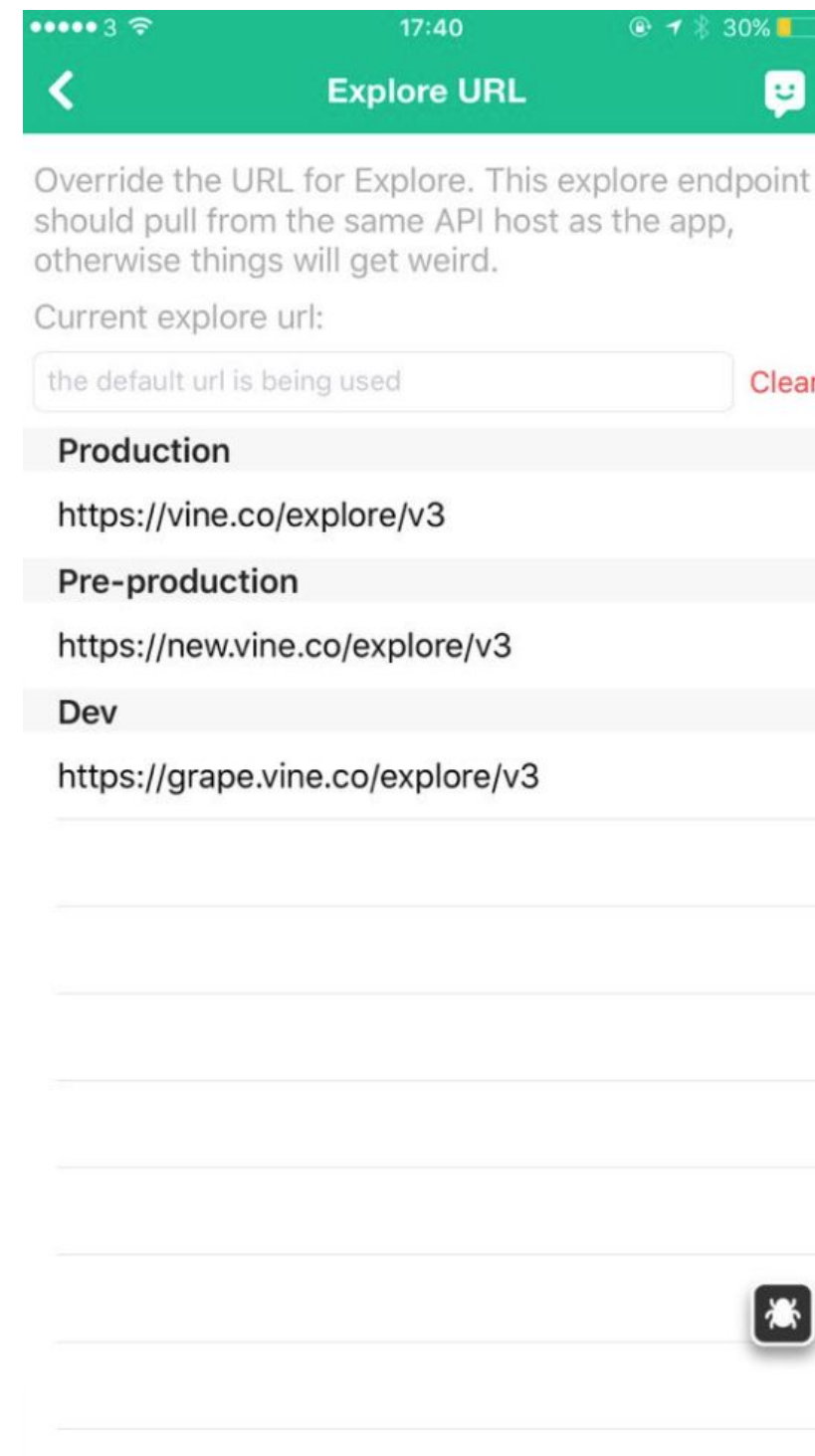
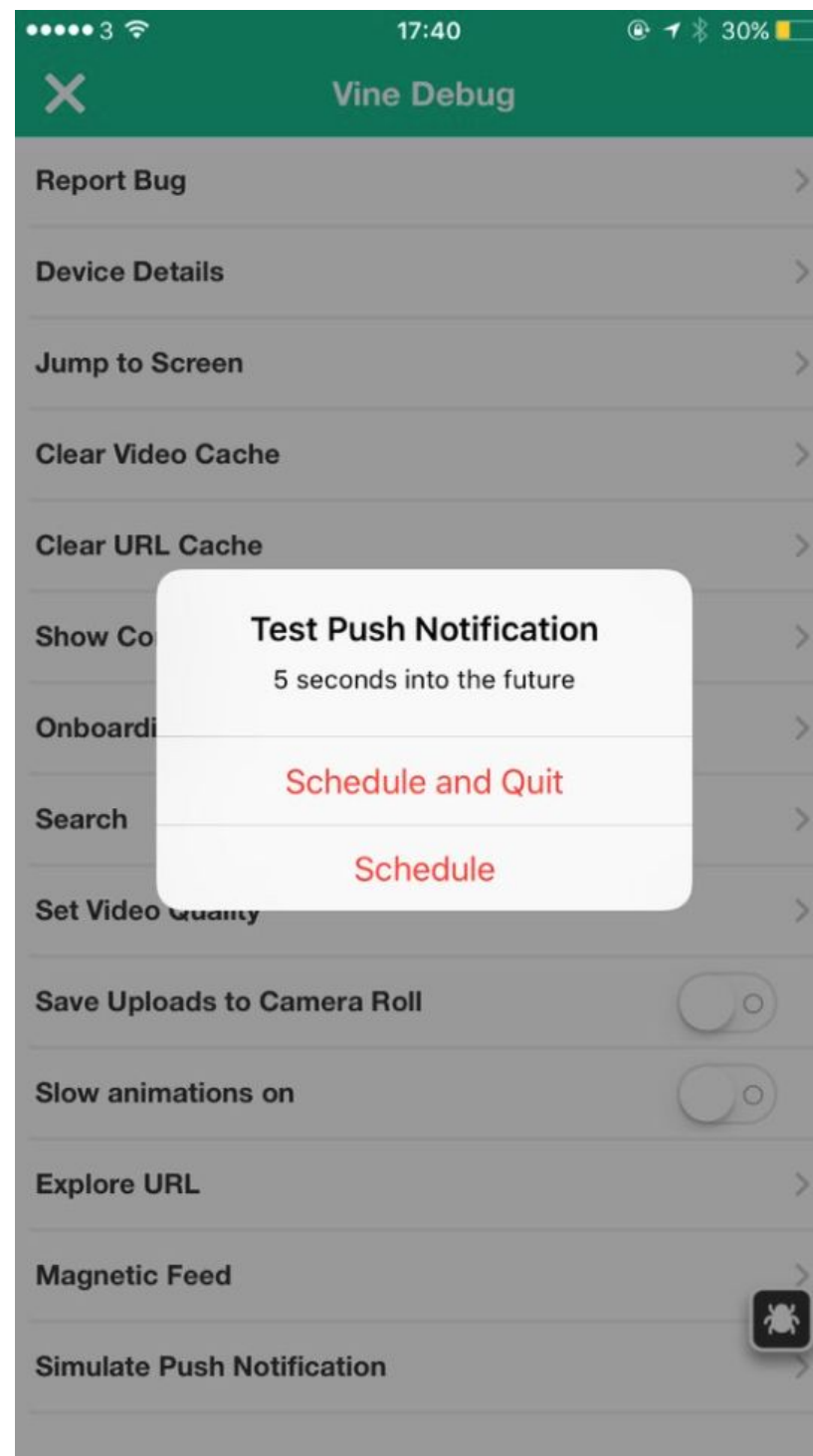
5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 3: Simple mistakes

The words "EPIC" and "FAIL" are written in a large, red, stylized font with a black outline and a green drop shadow. "EPIC" is positioned above the code snippet, and "FAIL" is positioned below it.

```
self.baseURL = "https://test.allegro.pl"
```


Problem 3: Simple mistakes



Problem 3: Simple mistakes

Solution: Code review

The screenshot shows a GitHub pull request interface. At the top, it says "#1 DECLINED" and "bugfix/TIS-67-fix-bug-in-develop... → master". Below this is the title "Bugfix/TIS-67 fix bug in developer toolbox" and "1 Reviewer". The "Overview" tab is selected. On the left, under "Changed files", the file "src/main/java/com/atlassian/devrel/condition/DevModeCondition.java" is listed. The main area shows a diff for this file. A green circle highlights a code review comment from Kevin Campbell, who says: "This isn't gonna work - we need a new c". The comment is dated "31 Jan 2014". The code diff shows a change in the "DevModeCondition.java" file. The original code (line 17) was: `return "true".equals("true").eq`. The new code (line 18) is: `return true;`. A green box highlights the new code. A red box highlights the original code. A green callout bubble points to the comment with the text: "Spot any mistakes before they happen".

#1 **DECLINED** | bugfix/TIS-67-fix-bug-in-develop... → master | [Re-open](#)

Bugfix/TIS-67 fix bug in developer toolbox | 1 Reviewer

Overview | Diff | Commits

Changed files

- src/main/java/com/atlassian/devrel/condition/DevModeCondition.java

src / main / java / com / atlassian / devrel / condition / DevModeCondition.java **MODIFIED**

```
7 7 | }
8 8 | }
9 13 |
14 14 | @Override
15 15 | public boolean showDevMode() throws PluginParseException {
16 16 |     return "true".equals("true").eq
17 - | "true".eq
18 + | true;
19 19 | }
20 20 | }
```

Kevin Campbell
This isn't gonna work - we need a new c
Reply · Delete · 31 Jan 2014

Spot any mistakes before they happen

Problem 3: Simple mistakes

Solution: Snapshot tests in code review

```
FA-bkv"/>
<constraint firstItem="NJn-Jb-beP" firstAttribute="bo
    kH-b4z" secondAttribute="bottom" constant="-6" id
<constraint firstItem="lvH-hc-h3e" firstAttribute="he
    secondItem="A5d-2m-hB0" secondAttribute="height"
<constraint firstItem="NJn-Jb-beP" firstAttribute="le
    kH-b4z" secondAttribute="leading" constant="6" id
<constraint firstAttribute="trailingMargin" secondItem="Ed1-eZ-iLM"
    secondAttribute="trailing" id="pA6-FX-Jq0"/>
<constraint firstItem="Ed1-eZ-iLM" firstAttribute="bottom"
    secondItem="A5d-2m-hB0" secondAttribute="bottom" id="r0d-yo-Jcq"/>
<constraint firstItem="Ed1-eZ-iLM" firstAttribute="leading"
    secondItem="A5d-2m-hB0" secondAttribute="trailing" constant="12"
    id="rlj-L1-Qil"/>
<constraint firstItem="CIO-Wk-mF2" firstAttribute="top" secondItem="lvH-hc-
    h3e" secondAttribute="top" constant="6" id="uBb-lJ-jfH"/>
<constraint firstItem="nQh-kH-b4z" firstAttribute="bottom"
    secondItem="A5d-2m-hB0" secondAttribute="bottom" constant="-6" id="wGY-
    e0-oVh"/>
```



Do wyczerpania przedmiotów

65,00 zł

PASEK SKÓRZANY W
PUDEŁKU GWAR3LATA
BETLEWSKI skóra

106 osób kupiło 120 sztuk

5 705 sztuk zostało

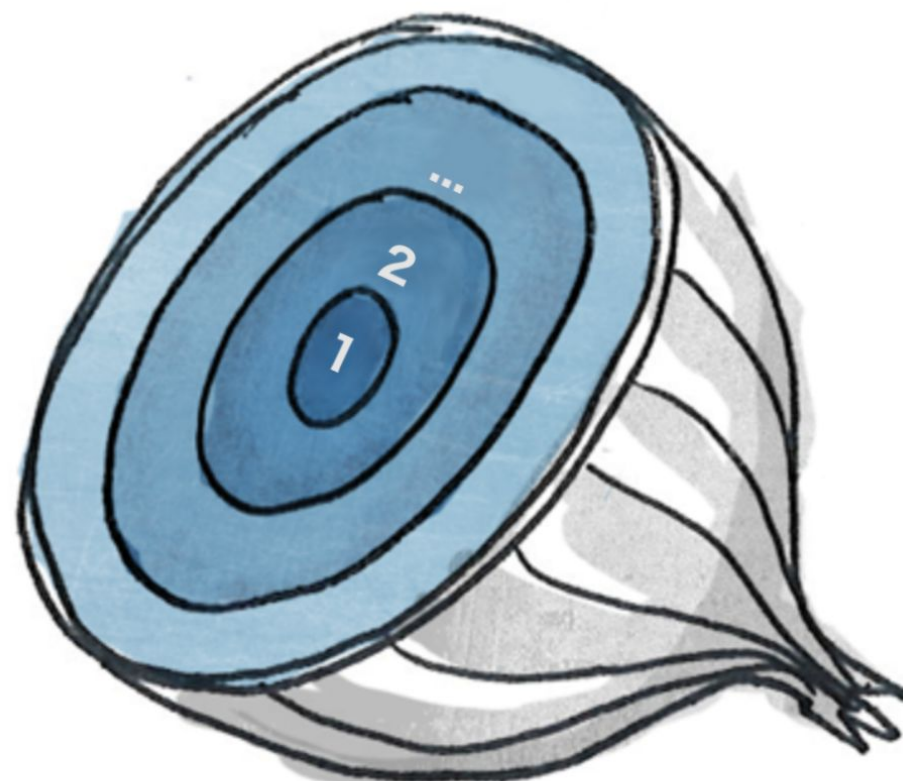
Problem 3: Simple mistakes

Solution: Snapshot tests in code review



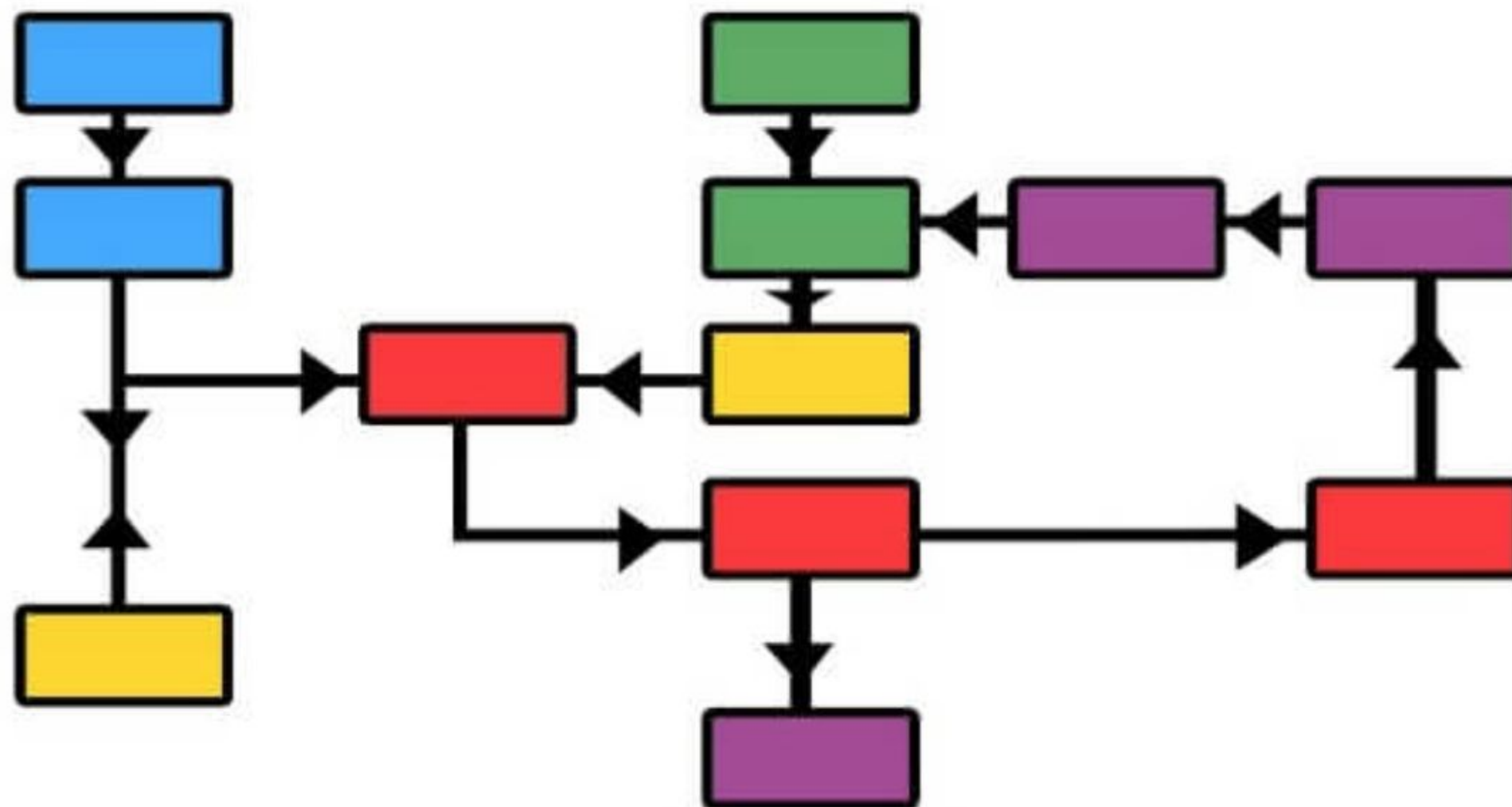
Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 4: Predicting consequences of change



Problem 4: Predicting consequences of change

Solution: Functional tests



1. functional UI tests
2. separate target
3. monitoring change



Remembering Mark Zuckerberg

We hope people who love Mark will find comfort in the things others share to remember and celebrate his life.
Learn more about memorialized accounts and the legacy contact setting on Facebook.

✕



Mark Zuckerberg 

[Follow](#) [Message](#) ⋮

[Timeline](#) [About](#) [Friends](#) [Photos](#) [More ▾](#)

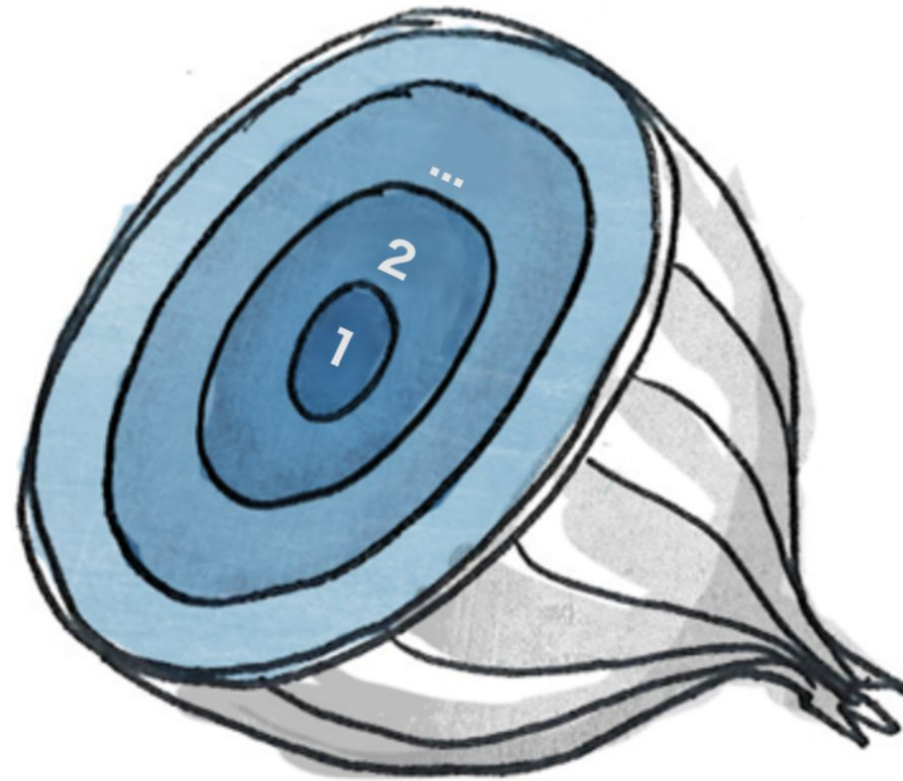
Follow Mark to get his public posts in your News Feed.

84,638,237 Followers

[Follow](#)

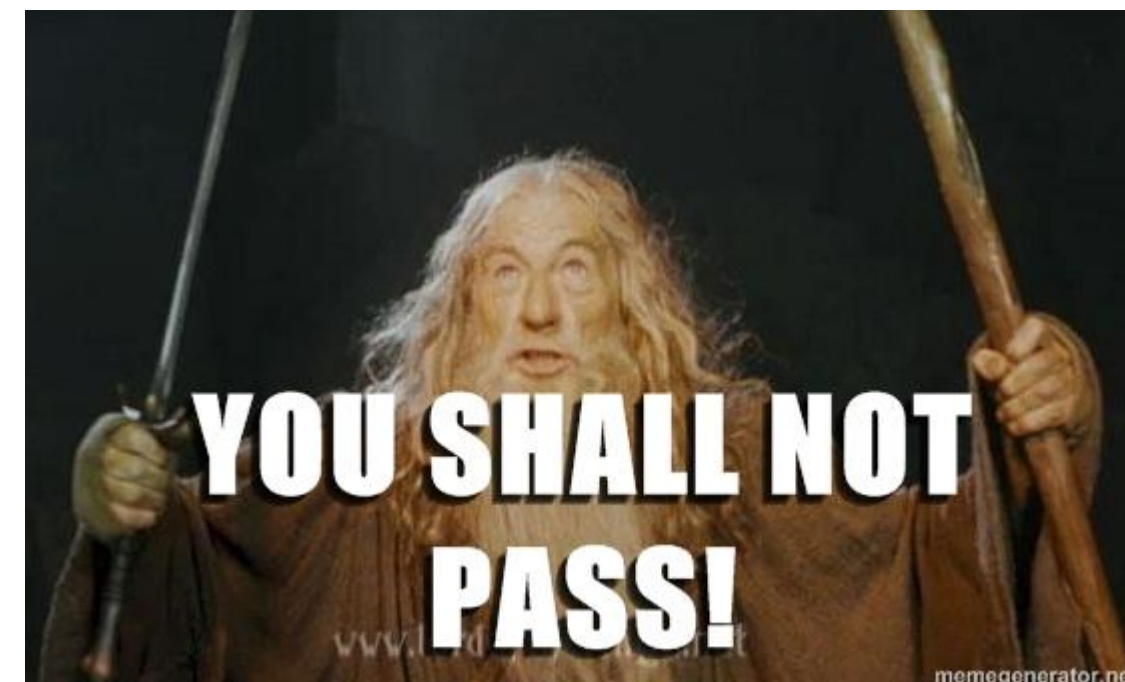
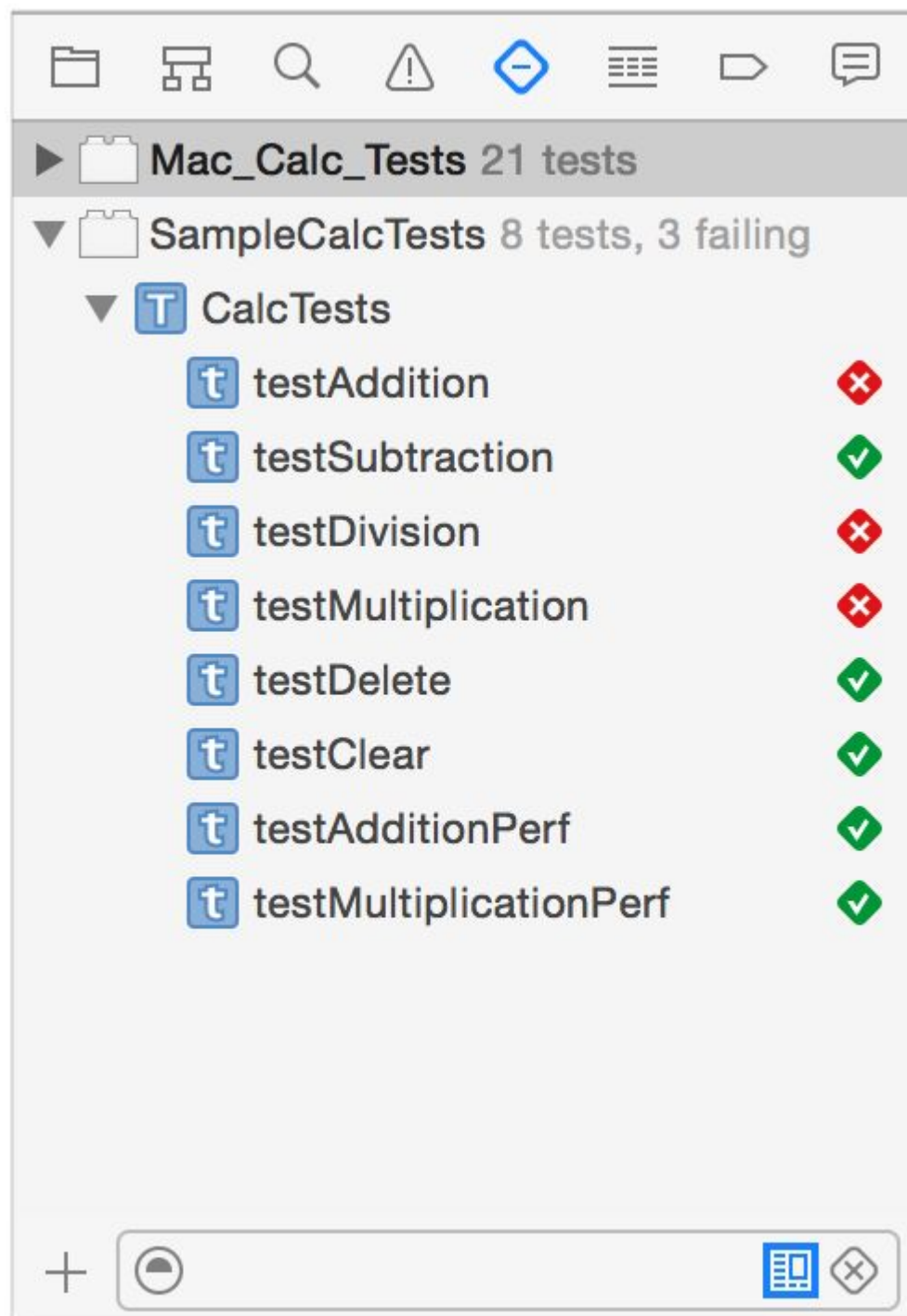
Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 5: Late feedback



Problem 5: Late feedback

Solution: Continuous integration

```
[22:15:41]: ▶ ✓ testShowNewPage (10.260 seconds)
[22:15:53]: ▶ ✓ testUsernameIsWasin (11.824 seconds)
[22:15:53]: ▶ Executed 3 tests, with 0 failures (0 unexpected)
[22:15:53]: ▶
[22:15:53]: ▶ Test Succeeded
+-----+
|      Test Results      |
+-----+
| Number of tests   | 3 |
| Number of failures | 0 |
+-----+

[22:15:54]: Successfully generated report at '/Volumes/Slave/Data/
[22:15:55]: Successfully generated report at '/Volumes/Slave/Data/

+-----+-----+-----+-----+
|                               fastlane summary                               |
+-----+-----+-----+-----+
| Step | Action                               | Time (in s) |
+-----+-----+-----+-----+
| 1     | Verifying required fastlane version | 0            |
| 2     | default_platform                     | 0            |
| 3     | scan                                 | 51           |
+-----+-----+-----+-----+

[22:15:55]: fastlane.tools finished successfully 🎉
```



fastlane

<https://fastlane.tools/>

Problem 5: Late feedback

How Projects Really Work (version 1.5)



How the customer
explained it



How the project leader
understood it

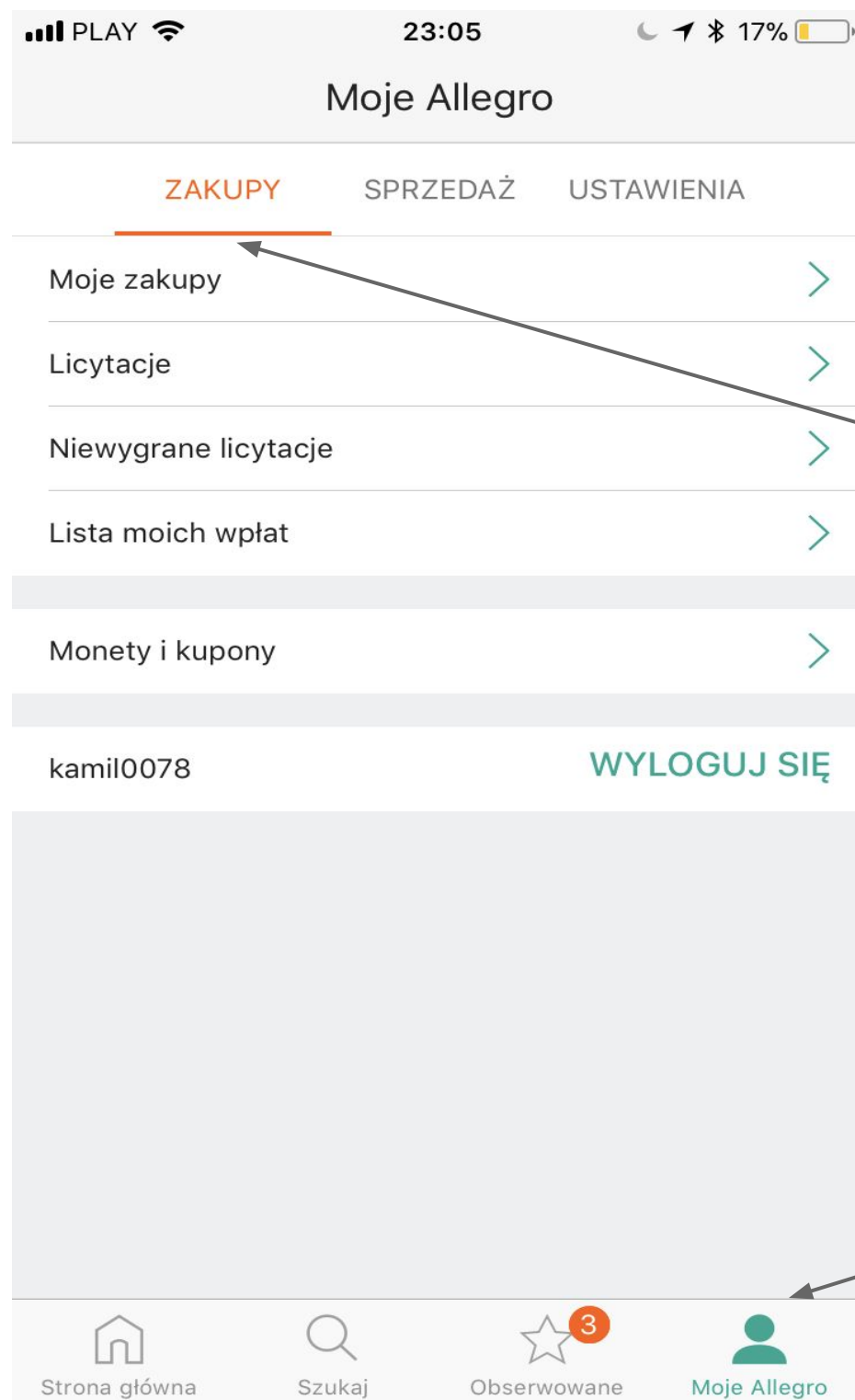


How the analyst
designed it



How the programmer
wrote it

Problem 5: Late feedback



TabControl

TabBar

Problem 5: Late feedback

Solution: Continuous delivery

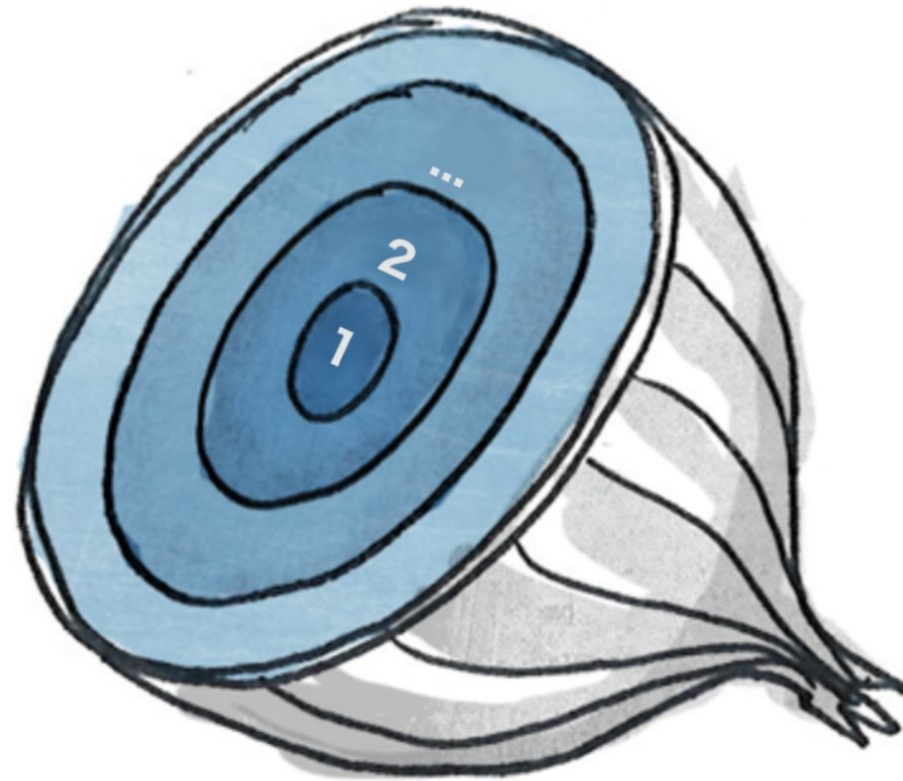


TestFlight

Beta Testing On The Fly

Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 6: Testing in laboratory condition



Allegro ★

Autor: Mateusz XXX

Wogole sie nie uruchamia po altualizacji

Problem 6: Testing in laboratory condition

Solution: Phased release

Phased Release for Automatic Updates (Day 1)

This update is being gradually released over a 7-day period to users with automatic updates turned on. It's current update to this version manually. You can release this update to all users at any time or pause the phased release.

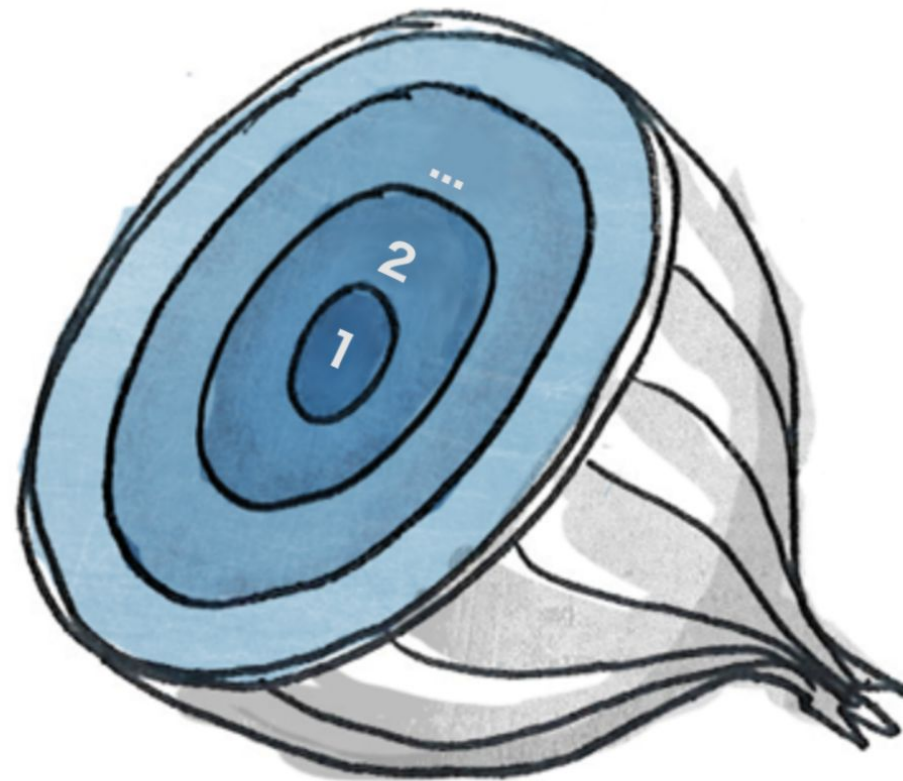
- ☒ Use phased release
- ☐ Pause the phased release



i This update will be available through automatic updates to the maximum percentage shown for each day.

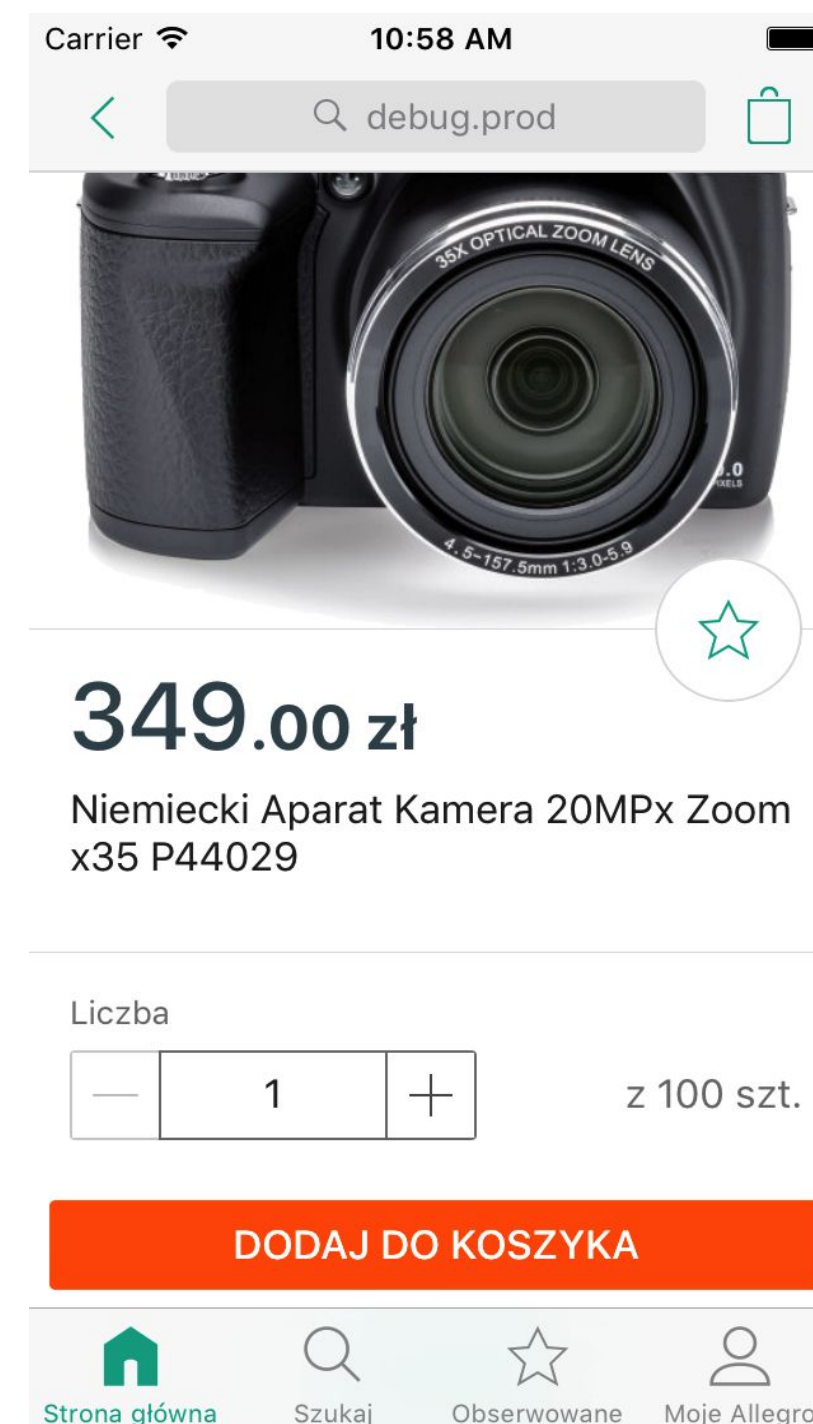
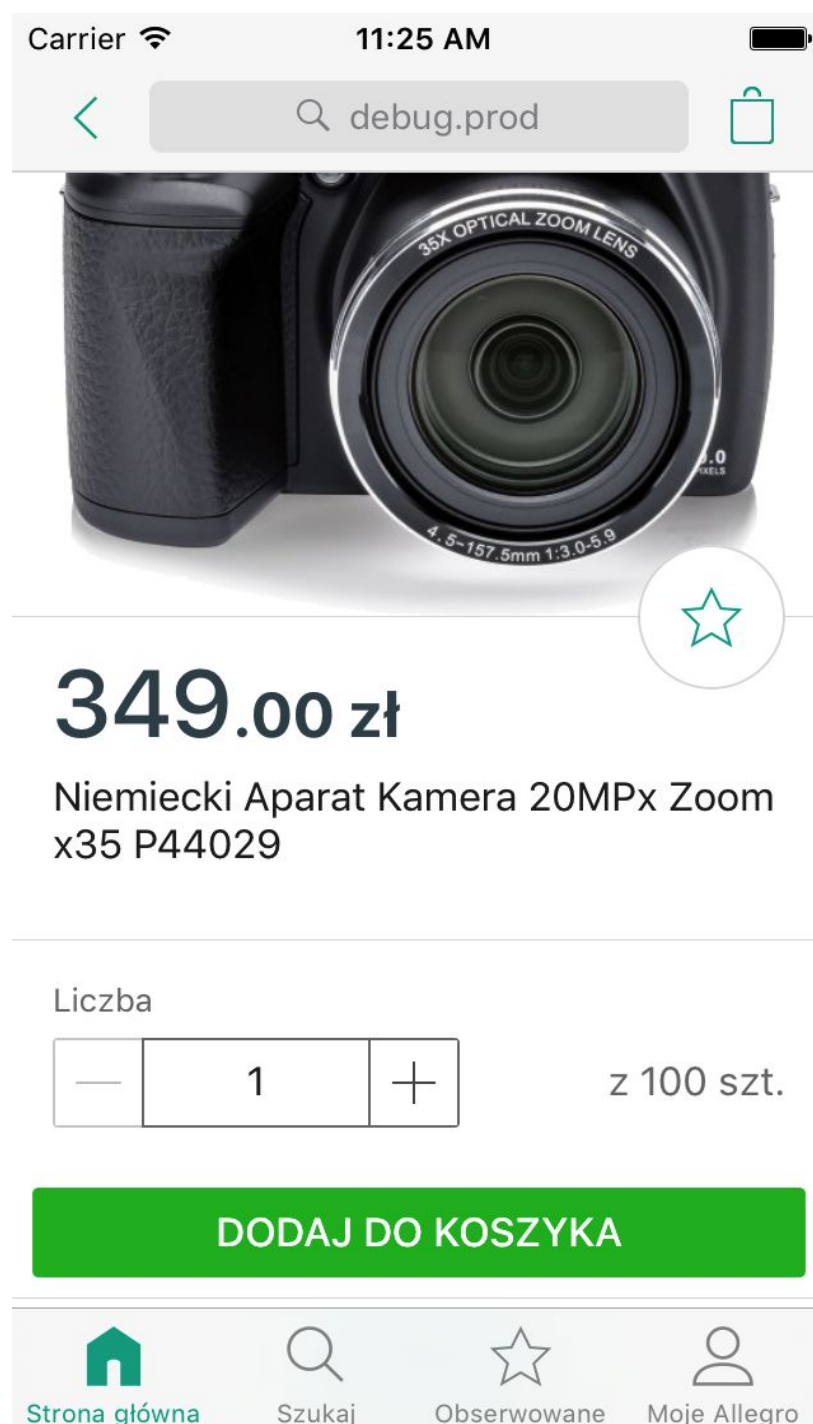
Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



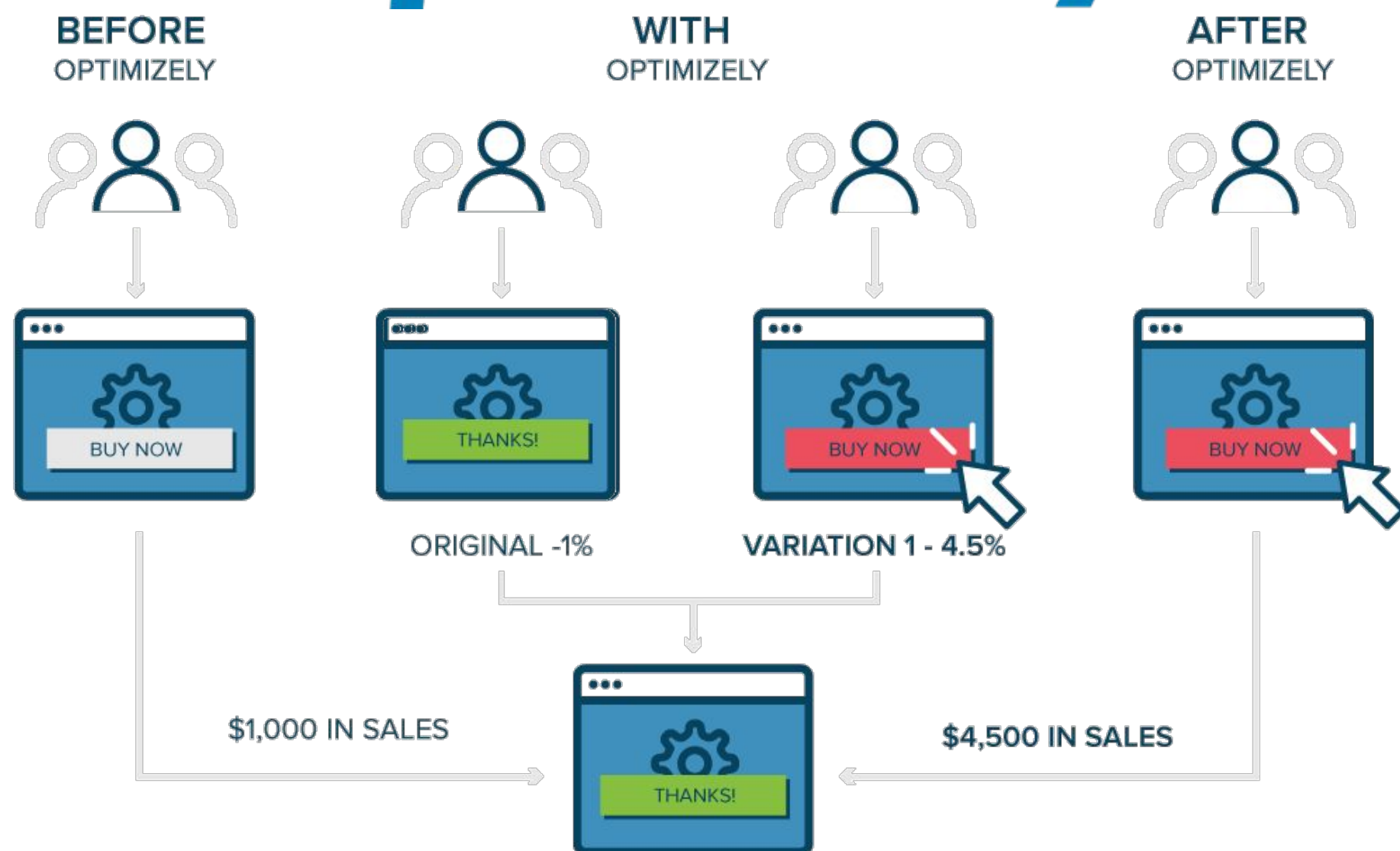
5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 7: Testing effectiveness



Problem 7: Testing effectiveness

Solution: AB Tests



Problem 7: Testing effectiveness

Solution: AB Tests

Traffic Allocation

Specify the percentage of traffic that should be included in this experiment and be tracked for conversions. Visitors that are not included will not count towards your monthly quota.

50%

Included in experiment.

0%

Excluded from experiment.



Percentage of experiment visitors:

Original - red button

50

%

Unpause

New - green button

50

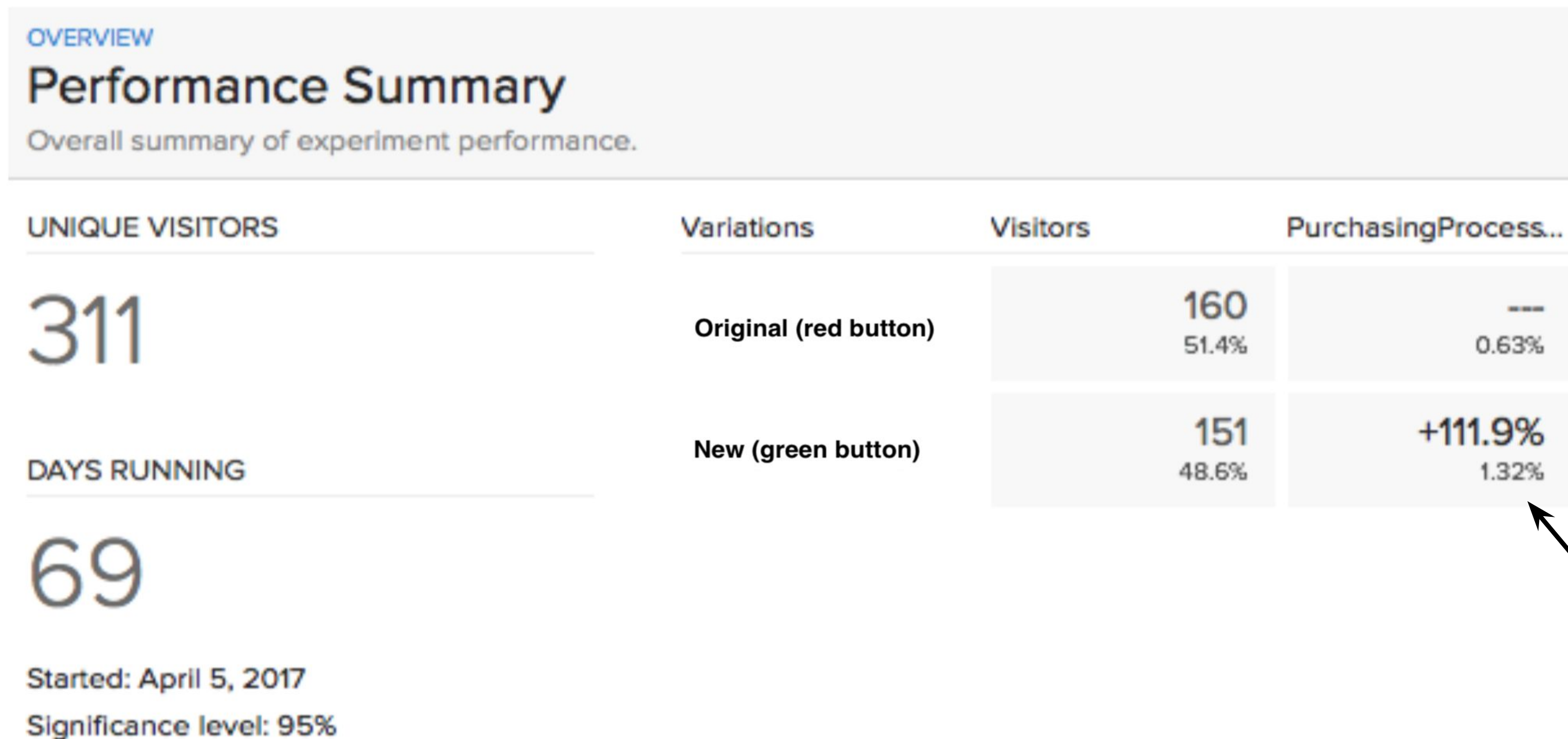
%

Unpause

[Learn more about Traffic Allocation](#)

Problem 7: Testing effectiveness

Solution: AB Tests



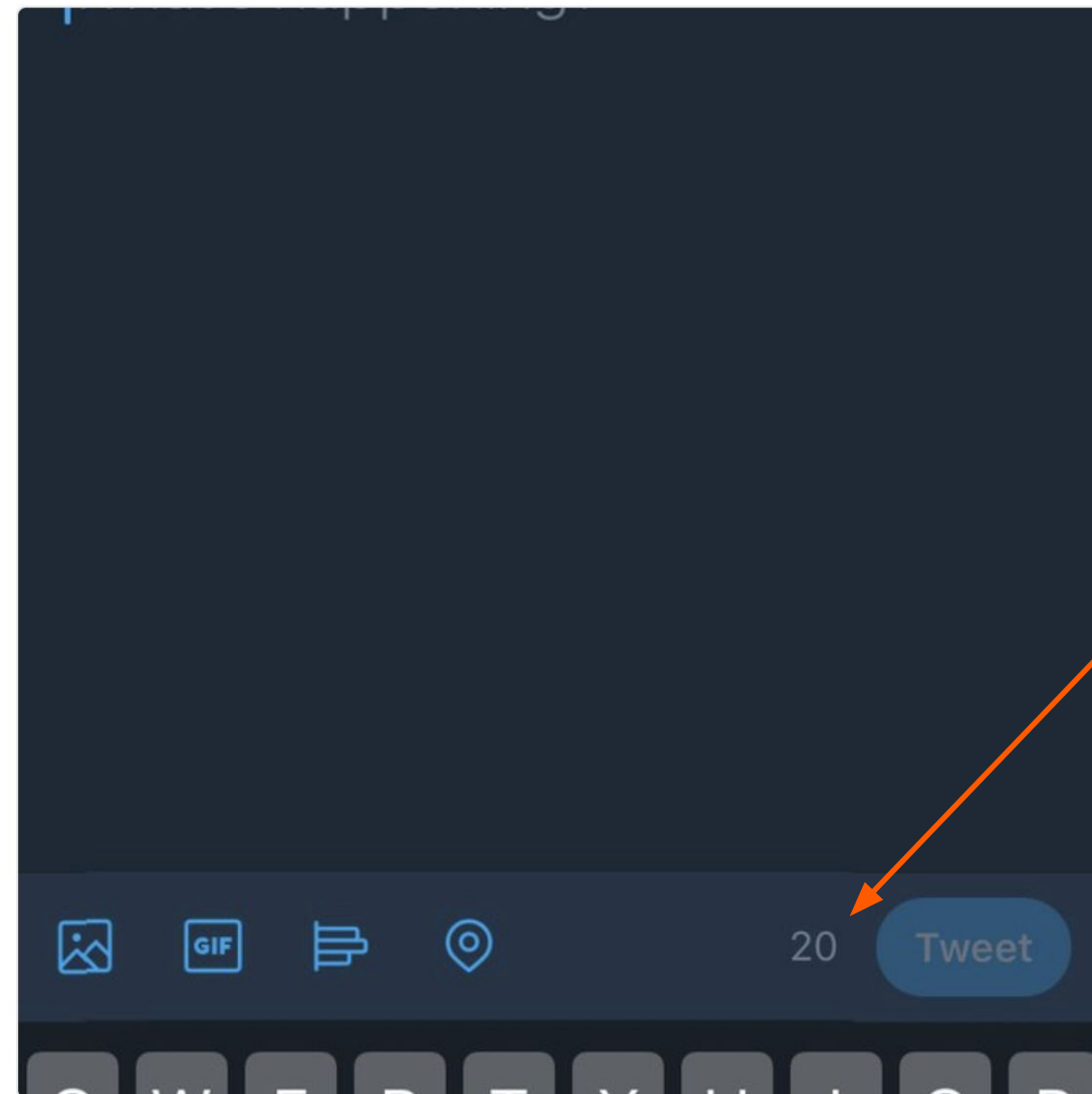


Chad Etzel 
@jazzychad

Follow



hlp im in bad ab tst



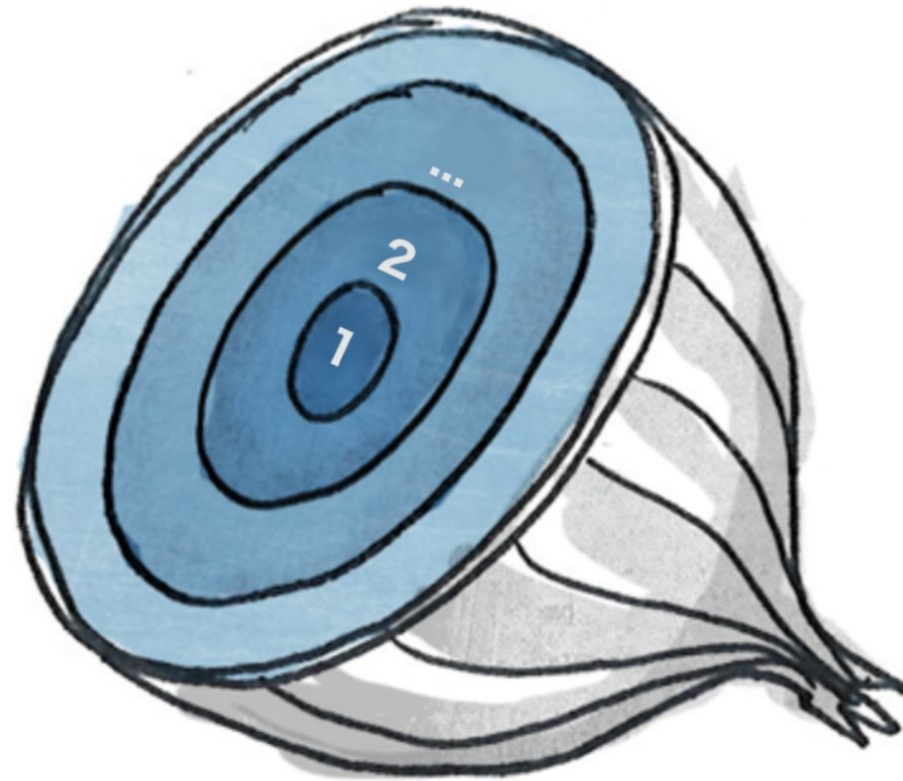
1:41 PM - 28 Sep 2017

79 Retweets 241 Likes



Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Problem 8:

We released app update to AppStore

We didn't see any significant crashes in Crashlytics 

Then we saw this comment on AppStore...

<https://fabric.io/kits/ios/crashlytics>

Problem 8: Decision making

Do bani

3 Sep

★★★★☆ by kastrol XXX

Gdzie zniknęły kategorie? Bardzo słaba aktualizacja

← Developer Response

Kategorie nie zniknęły, znajdują się teraz pod ikoną pasku na dole ekranu.



Problem 8: Decision making

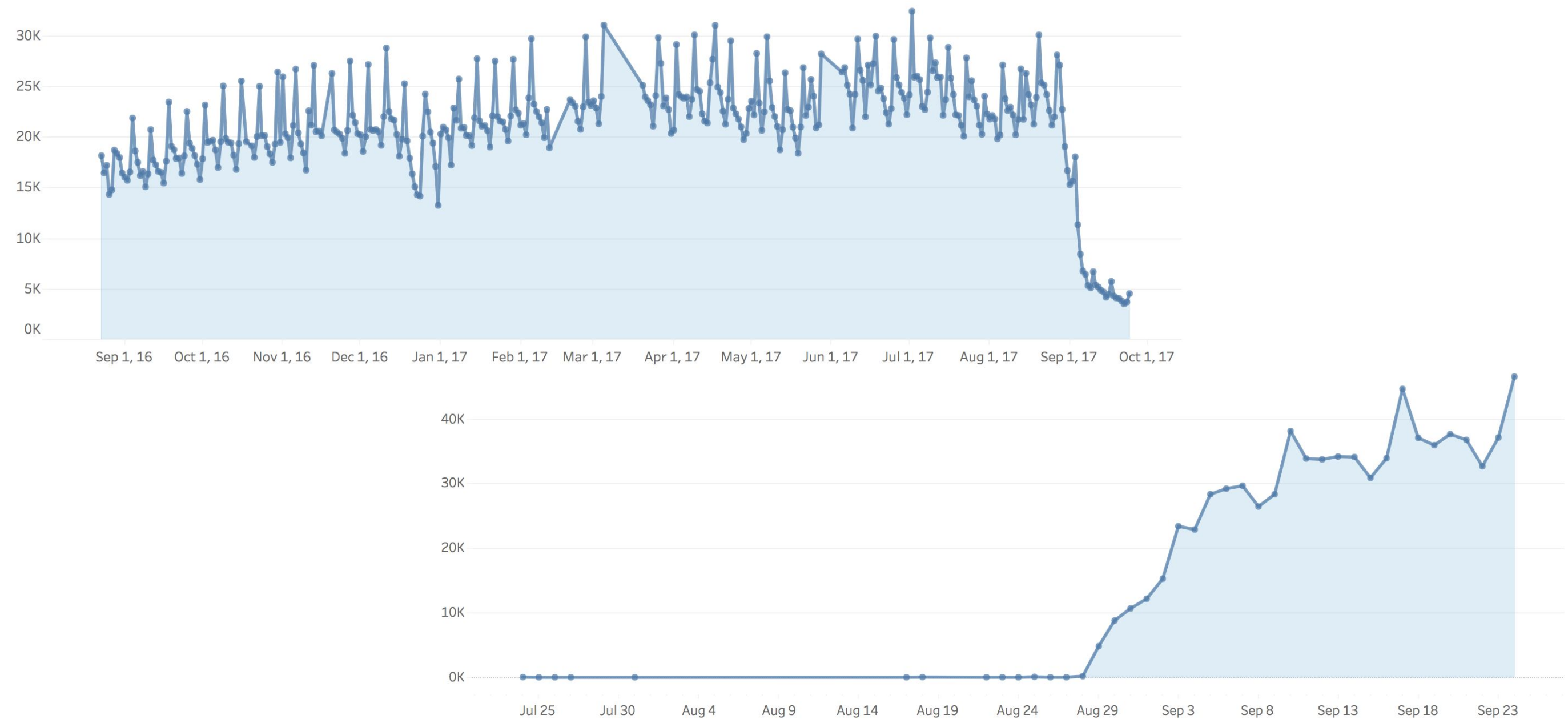
Solution: Make use of app analytical data



1. data-based decisions
2. priority and severity of bug
3. access to analytical data to team members

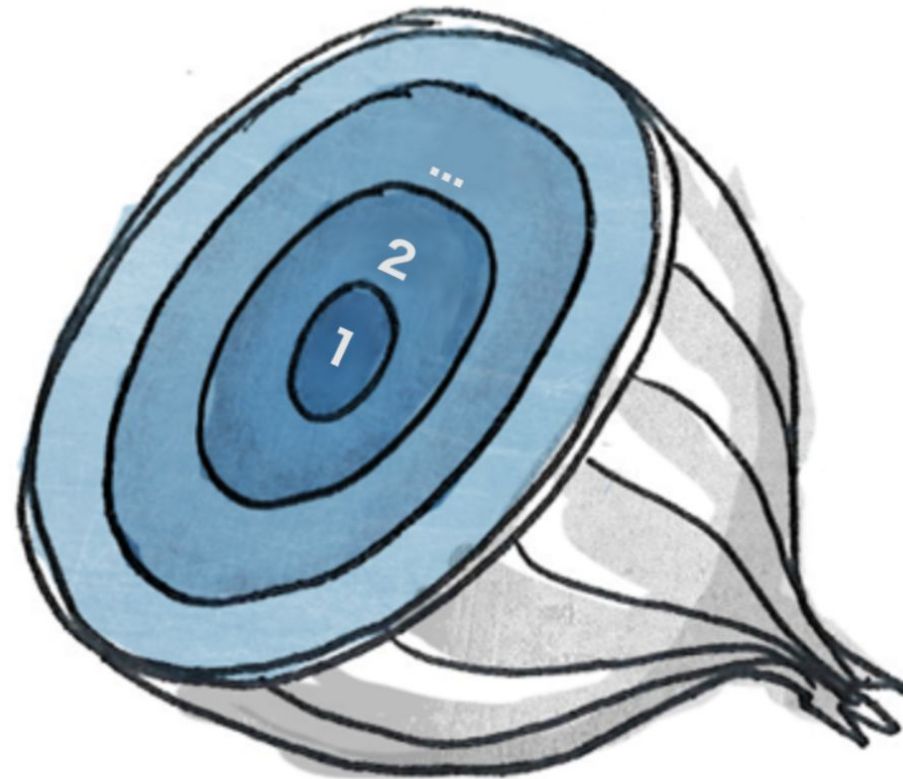
Problem 8: Decision making

Solution: Make use of app analytical data



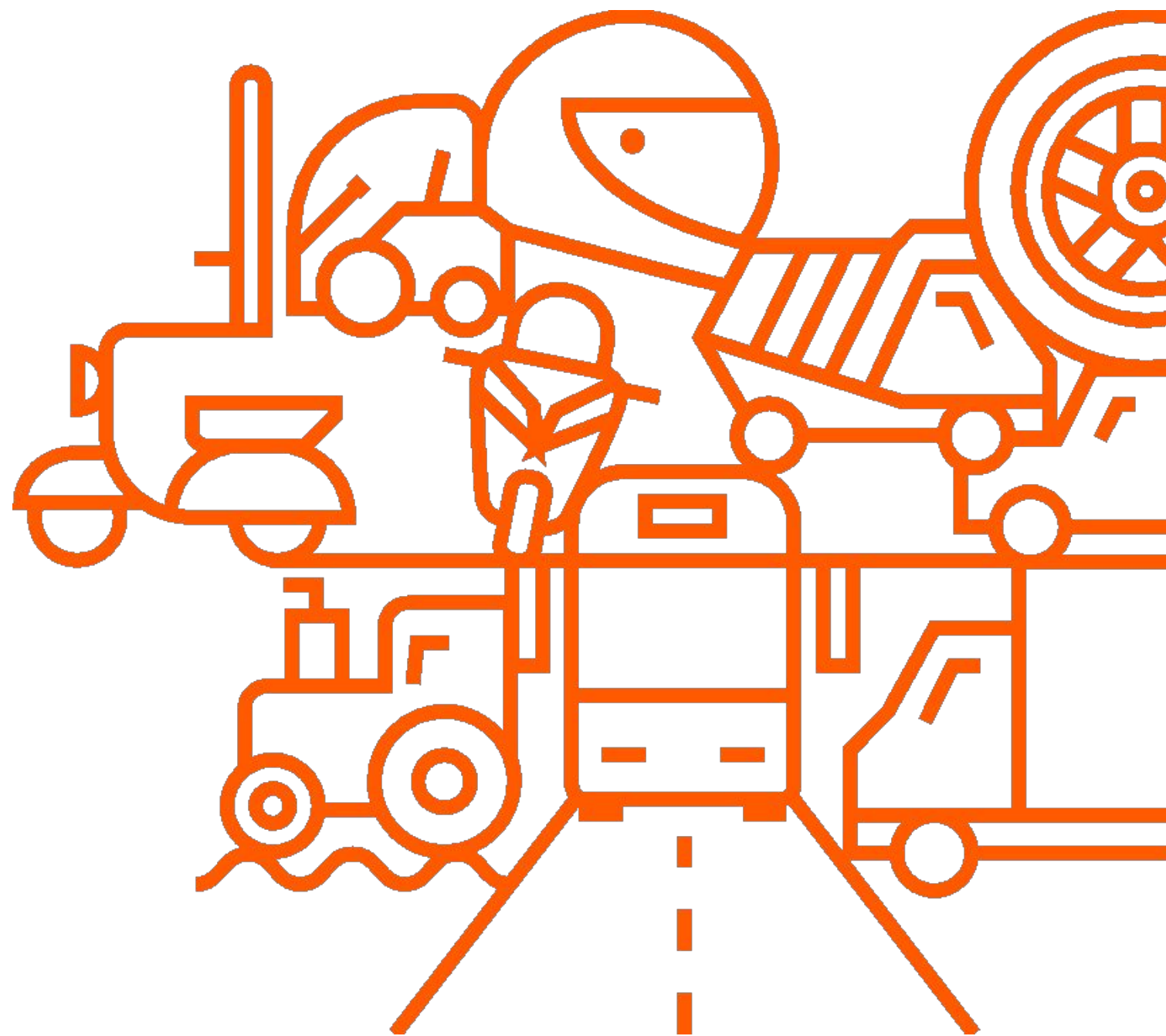
Tests are like onions

1. **Static** analysis
2. **Mutation** testing
3. **Snapshot** tests
4. **Functional** tests



5. **Continuous** integration
6. **Phased** release
7. **AB** Tests
8. Application **analytics**

Thank you for your attention





Ewa Ludwiczak

Senior Software Tester

ewa.ludwiczak@allegrogroup.com

 [@ewabielskapoz](https://twitter.com/ewabielskapoz)



allegro.tech



allegrotechblog



Kamil Pyć

Senior iOS Developer

kamil.pyc@allegrogroup.com

 [@KamilPyc](https://twitter.com/KamilPyc)



allegrotech