

From **Dread** to **Delight**:

Why You Should Embrace Writing Tests

(Even If You've Never Written One)



Leah Vogel
Swift Connection
2024

THANK YOU!



\$whoami

Leah Vogel

- Started iOS development in 2012 🦖
- Director of Engineering @ **Chegg** 🧐
- iOS Developer at ❤️



\$whoami

Leah Vogel

- 🖐️
- ✈️ 🇫🇷
- 👦 👧 👦 👦 👧 👦
- 🧑🏫
- 📖



Mostly Harmless

TL;DR

Today's Agenda

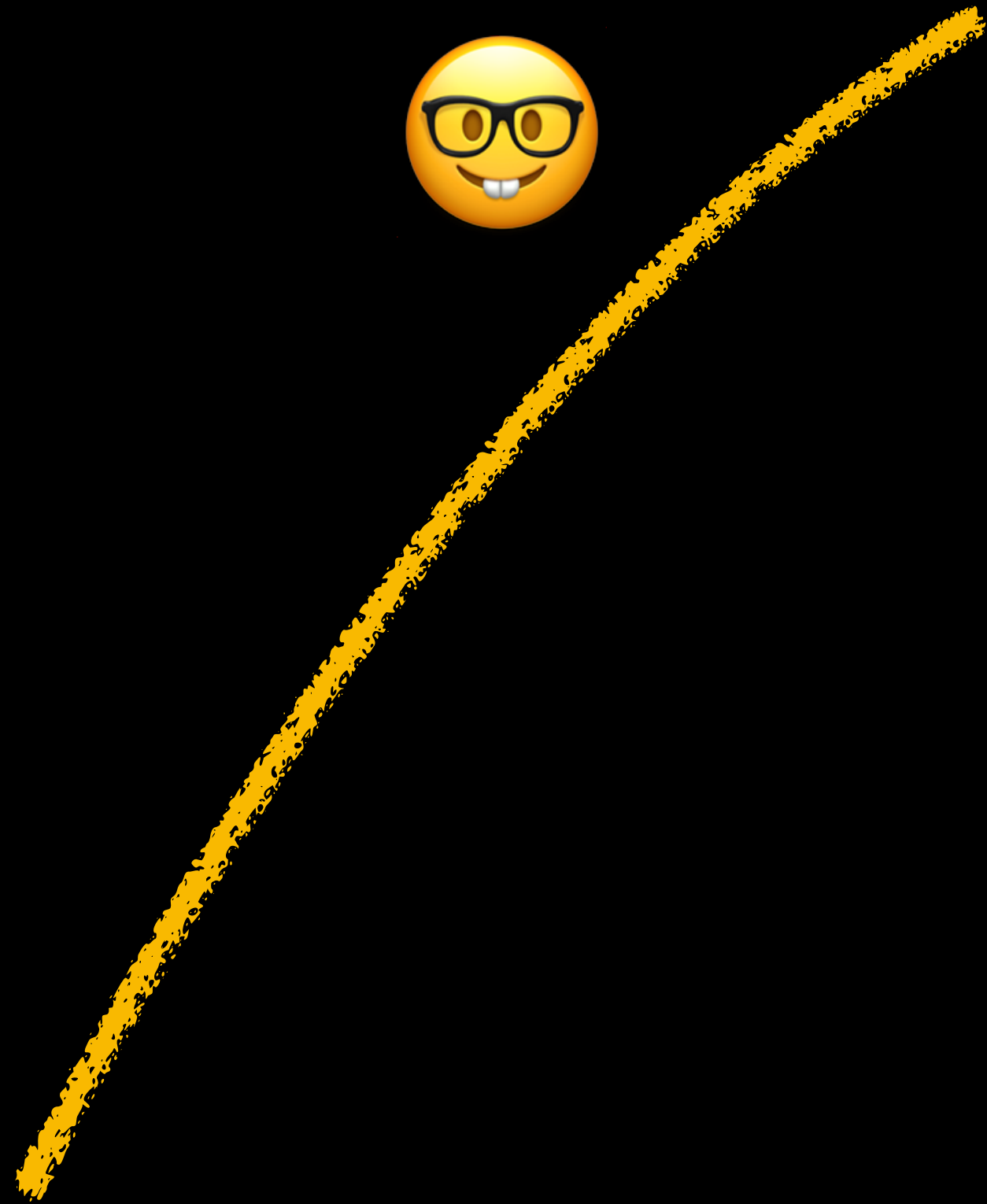
1. First Job (a.k.a. What's a Test?)
2. Testing - who, what and why
3. Next Level Goodness
4. New Technologies Ahead
5. Summing it All Up

Excitement



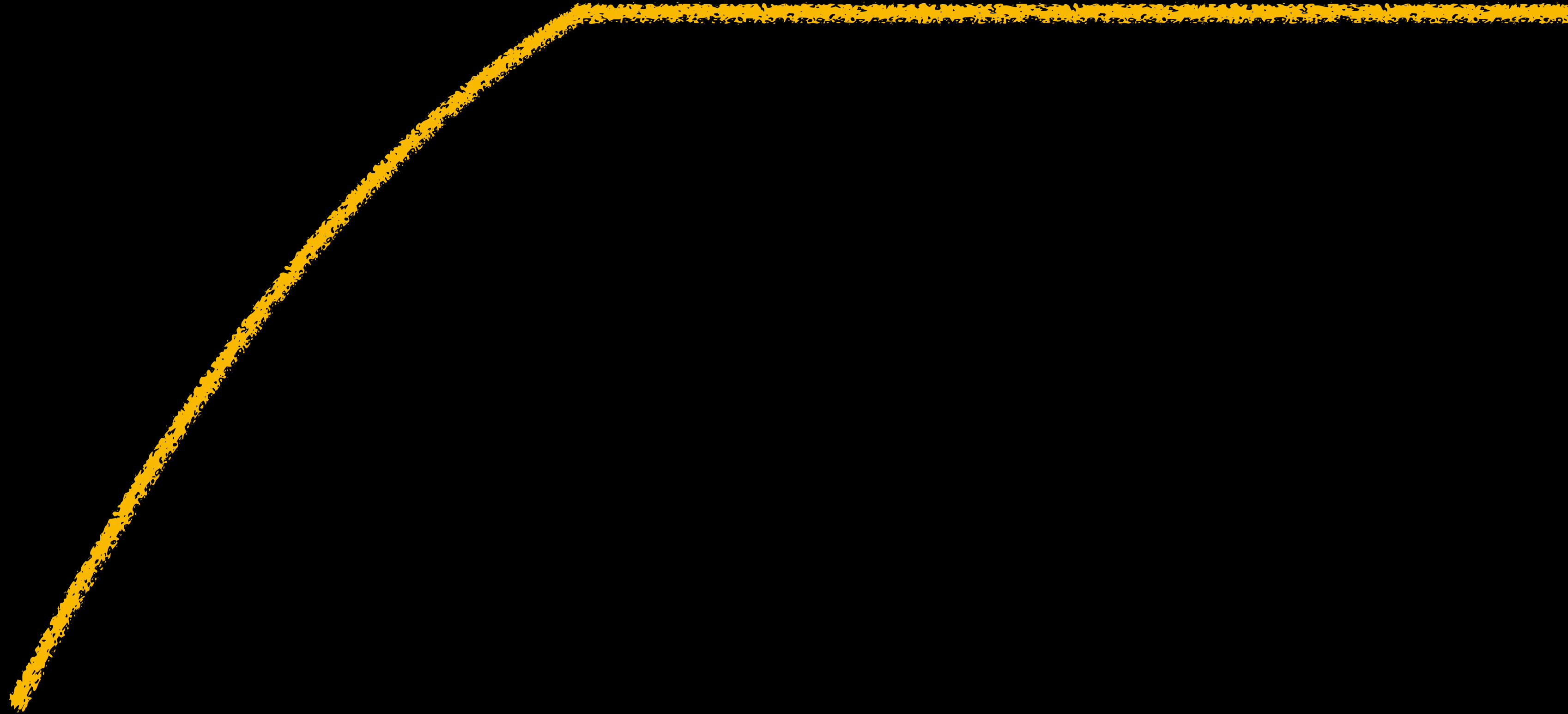
Development Lifecycle

Excitement



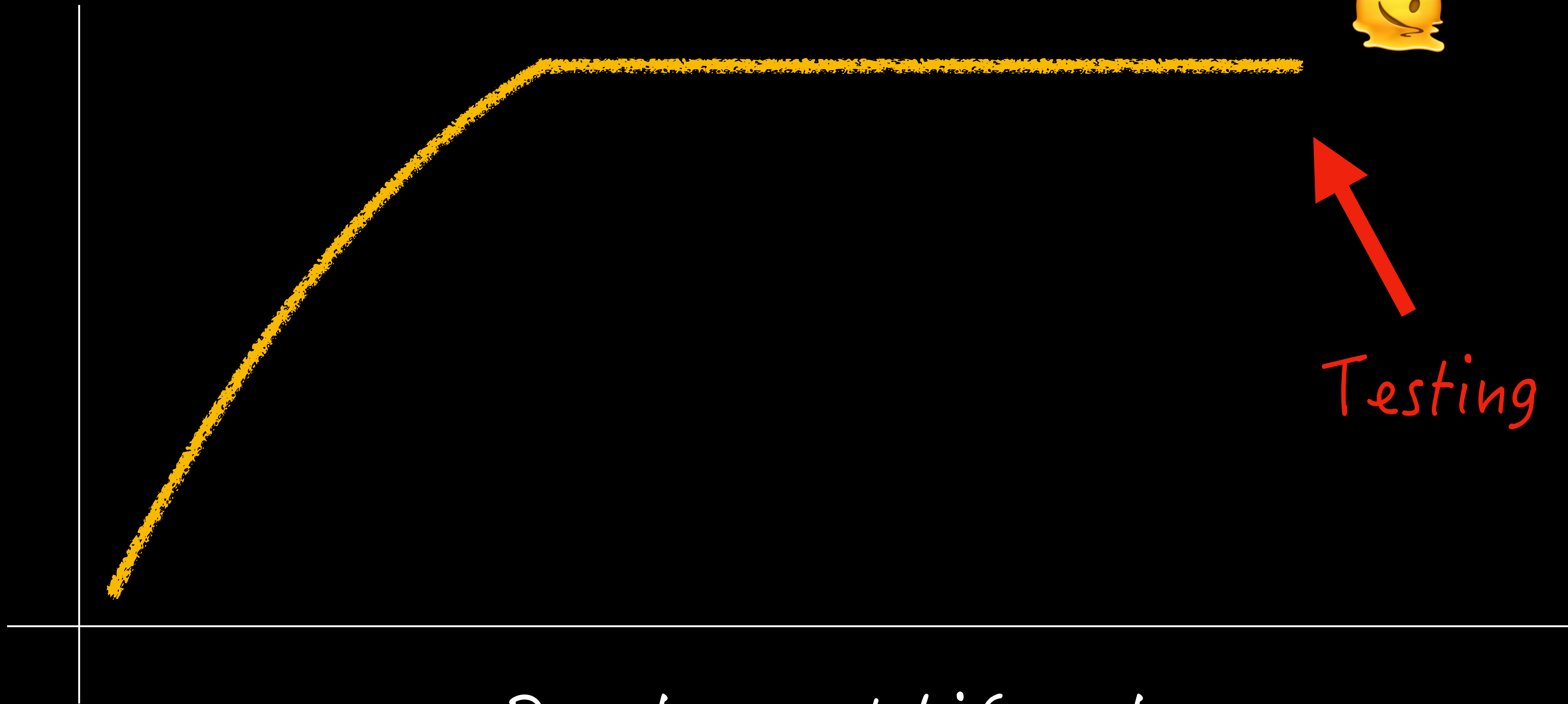
Development Lifecycle

Excitement



Development Lifecycle

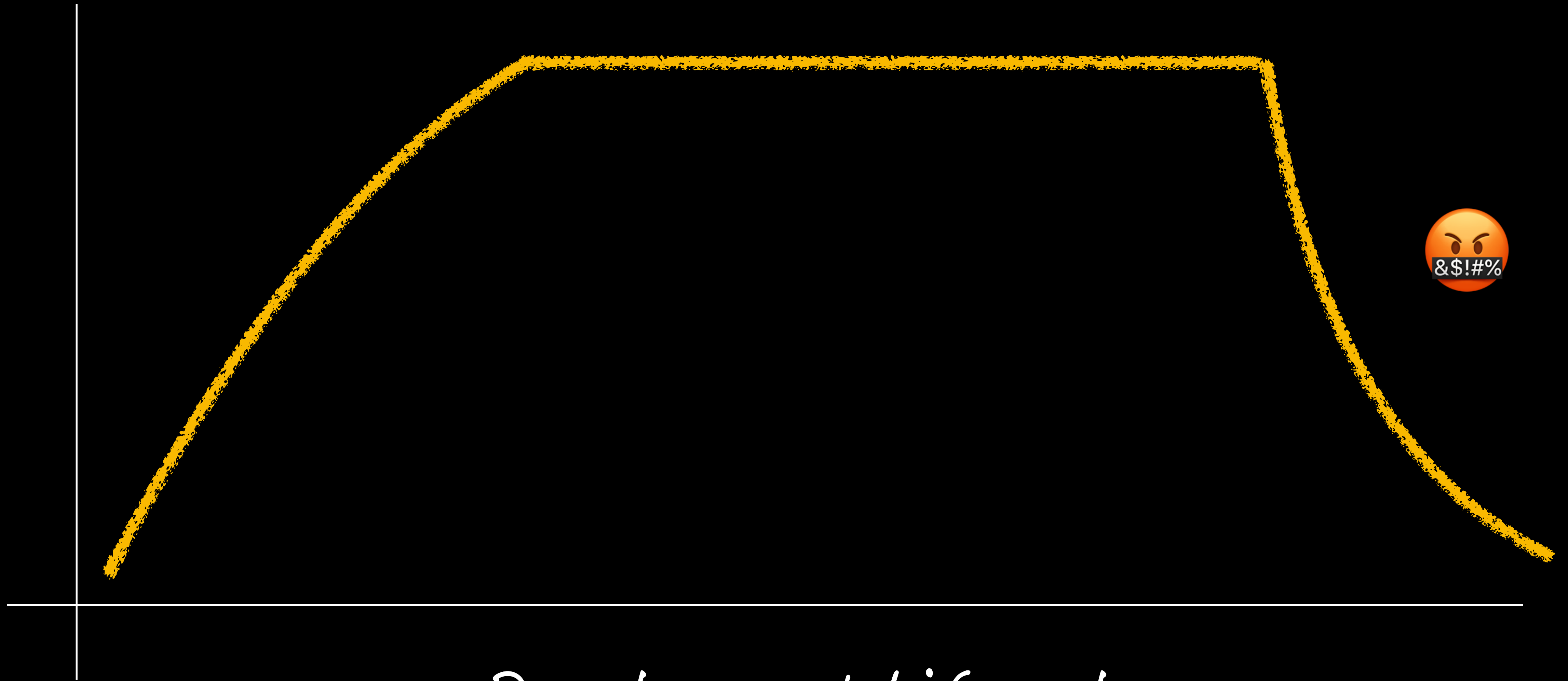
Excitement



Development Lifecycle

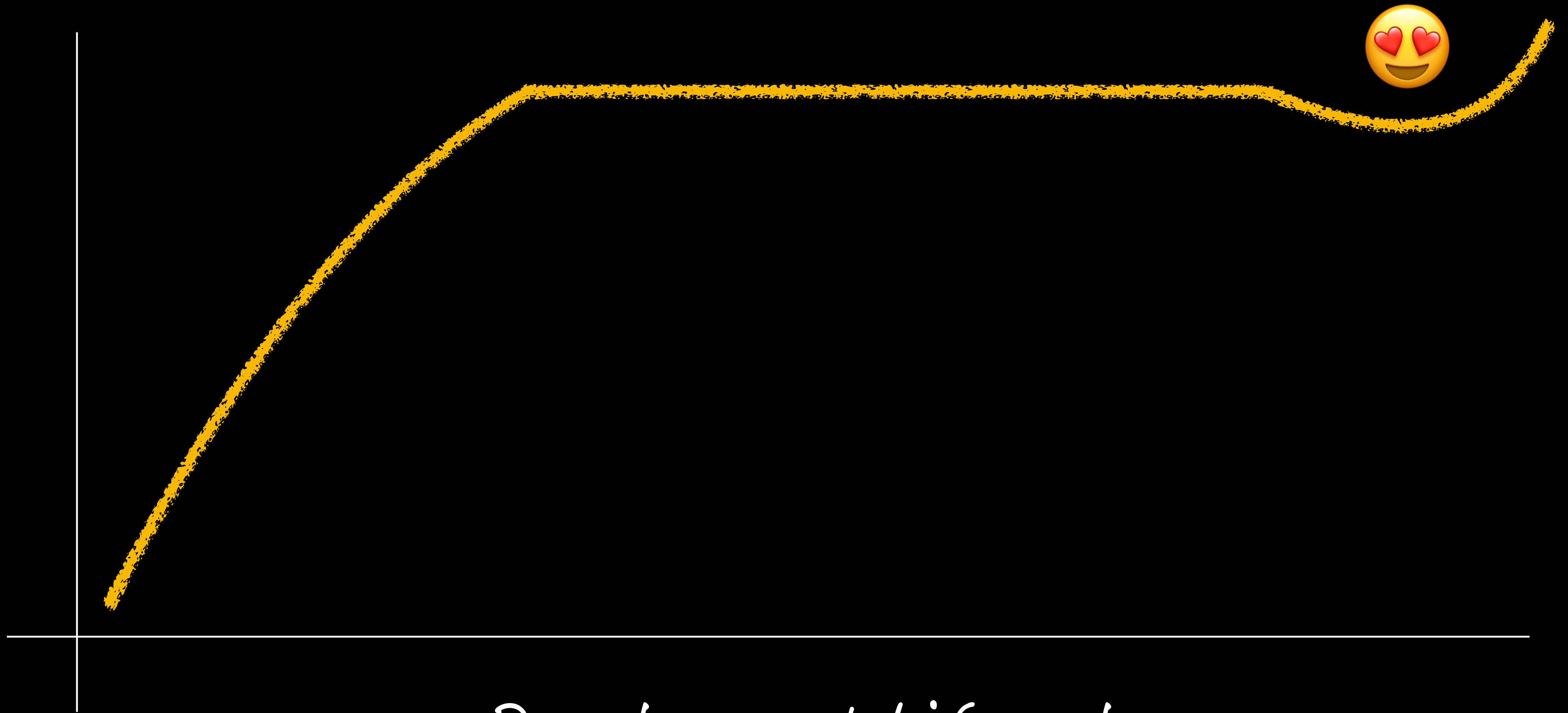
Testing

Excitement



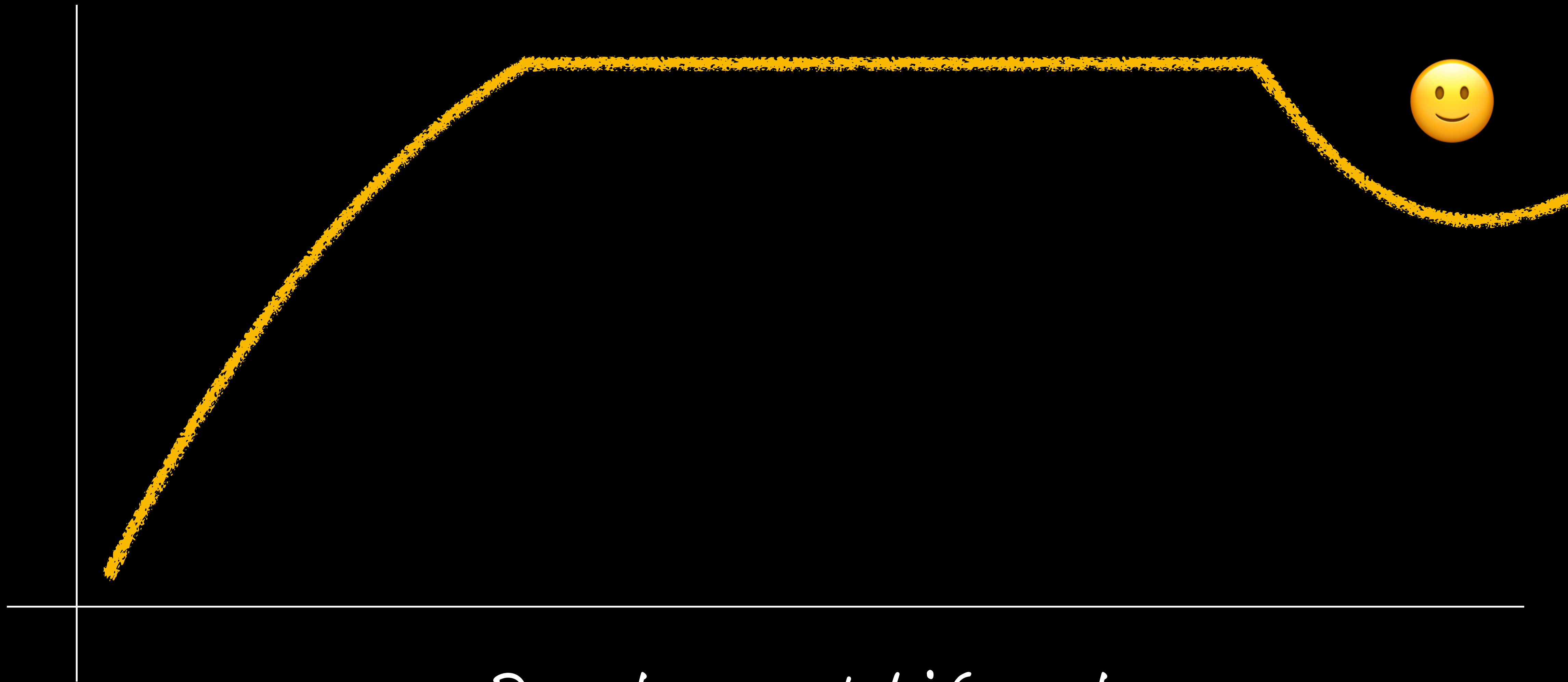
Development Lifecycle

Excitement



Development Lifecycle

Excitement



Development Lifecycle

DISCLAIMER



Chapter 1

First Job

(a.k.a. What's a Test?)



Important Learnings !!



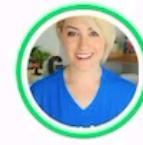
- Storyboards are the devil
- Git UI is for sissies
- Bookmark [f**kingblocksyntax.com](https://fkingblocksyntax.com)
- Always blame Server
- Developers don't test code

Glide
Video Texting

Glide



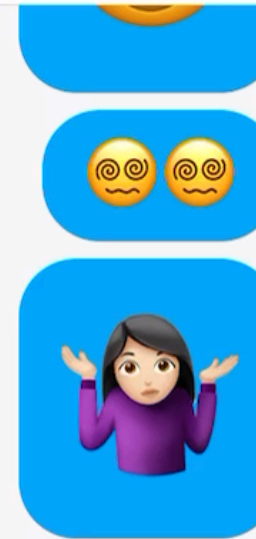
4G 85%



•

•

•



Send

🔍 Search Emoji

FREQUENTLY USED



SMILEYS &

ABC



```
- (BOOL) isSingleEmoji: (NSString*) inputText {  
    }
```

```
- (BOOL) isSingleEmoji: (NSString*) inputText {  
    // check if the length is exactly 1  
    if ([inputText length] == 1) {  
  
    } else {  
        return NO;  
    }  
}
```

```
- (BOOL) isSingleEmoji: (NSString*) inputText {  
    // check if the length is exactly 1  
    if ([inputText length] == 1) {  
        // check if it's an emoji  
        if (complicatedCodeHere) {  
            return YES;  
        } else {  
            return NO;  
        }  
    } else {  
        return NO;  
    }  
}
```




```
- (BOOL) isSingleEmoji: (NSString*) inputText {  
    // check if the length is exactly 2 because  
    apparently emojis are different?  
    if [inputText length] == 2 {  
  
    } else {  
        return NO;  
    }  
}
```





Me



Team Lead







Teaching Moment ✨

**DON'T CREATE AND
PASS MANAGED
OBJECT CONTEXTS
WITHOUT FIRST
SAVING IT AS A
VARIABLE**



Let's write a test!





– (BOOL) isEmoji: (NSString*) inputText {}

– (BOOL) isSingleEmoji: (NSString*) inputText {}

```
- (BOOL)isEmoji {
    BOOL returnValue = NO;
    const unichar hs = [self characterAtIndex:0];

    // surrogate pair
    if (0xd800 <= hs && hs <= 0xdbff) {
        if (self.length > 1) {
            const unichar ls = [self characterAtIndex:1];
            const int uc = ((hs - 0xd800) * 0x400) + (ls - 0xdc00) + 0x10000;
            if (0x1d000 <= uc && uc <= 0x1f77f) {
                returnValue = YES;
            }
        }
    } else if (0x2600 <= hs && hs <= 0x27bf) {
        returnValue = YES;
    } else if (self.length > 1) {
        const unichar ls = [self characterAtIndex:1];
        if (ls == 0x20e3) {
            returnValue = YES;
        }
    } else {
        // non surrogate
        if (0x2100 <= hs && hs <= 0x27ff) {
            returnValue = YES;
        } else if (0x2B05 <= hs && hs <= 0x2b07) {
            returnValue = YES;
        } else if (0x2934 <= hs && hs <= 0x2935) {
            returnValue = YES;
        } else if (0x3297 <= hs && hs <= 0x3299) {
            returnValue = YES;
        } else if (hs == 0xa9 || hs == 0xae || hs == 0x303d || hs == 0x3030 || hs == 0x2b55 || hs == 0x2b1c || hs == 0x2b1b || hs ==
0x2b50) {
            returnValue = YES;
        }
    }

    return returnValue;
}
```

```
- (void) testIsSingleEmoji {  
}
```



```
- (void) testIsSingleEmoji {  
    NSString *emoji = @"😊";  
    NSString *nonEmoji = @"Hello world!";  
}
```

```
- (void) testIsSingleEmoji {  
    NSString *emoji = @"😊";  
    NSString *nonEmoji = @"Hello world!";  
  
    XCTAssertTrue([self isSingleEmoji:emoji],  
"Expected single emoji to return true.");  
    XCTAssertFalse([self isSingleEmoji:nonEmoji],  
"Expected non-emoji text to return false.");  
}
```



```
- (void) testIsSingleEmoji {  
    NSString *emoji = @"🇫🇷";  
    NSString *nonEmoji = @"Hello world!";  
  
    XCTAssertTrue([self isSingleEmoji:emoji],  
"Expected single emoji to return true.");  
    XCTAssertFalse([self isSingleEmoji:nonEmoji],  
"Expected non-emoji text to return false.");  
}
```





$$2 \equiv 1$$

[@“🙄” length] == 2

[@“👤” length] == 7

[@“🇫🇷” length] == 4

[@“👥” length] == 11



size

glyph

length

utf-8

unicode scalar

utf-16

character count

invisible characters

Swift ❤️

“🤔”.count == 1

“🧔”.count == 1

“🇫🇷”.count == 1

“👥”.count == 1



I wrote a test.

I wrote a test.
Only once.

I wrote a test.
Only once.
Ever.



Wristcam



Video Calling.



Capture Photos.



Shoot Videos.



**The first and ONLY camera for the
Apple Watch.**



Chapter 2

Testing:

Who, Why and When?







TESTING

THEY KEEP USING THAT WORD







Me



*Little Brother Who
Knows Everything*









WHY?

WHY?

Why, indeed?



Some reasons we should write tests

Craftsmanship



Architecture



Reliability



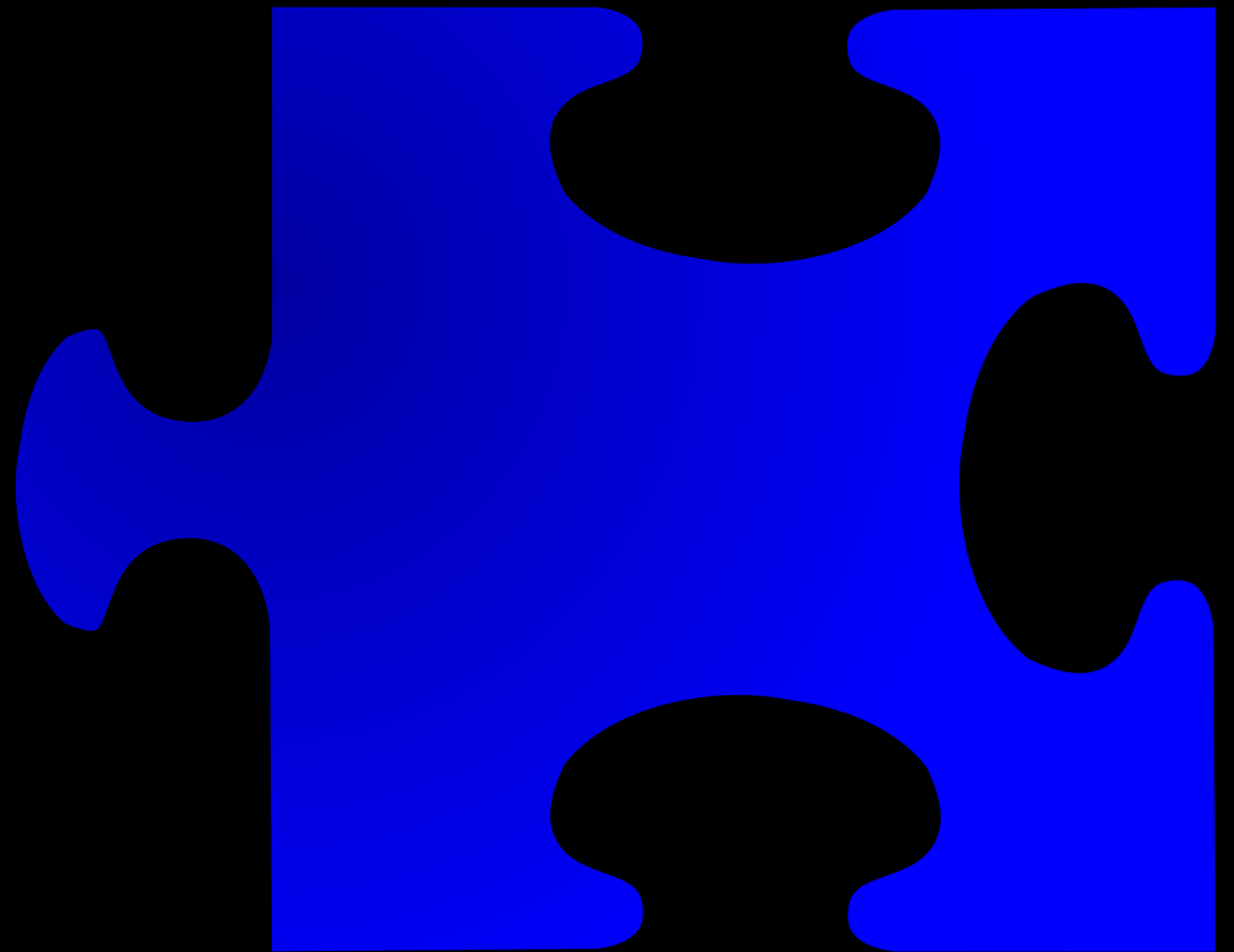
Ownership



Types of Tests

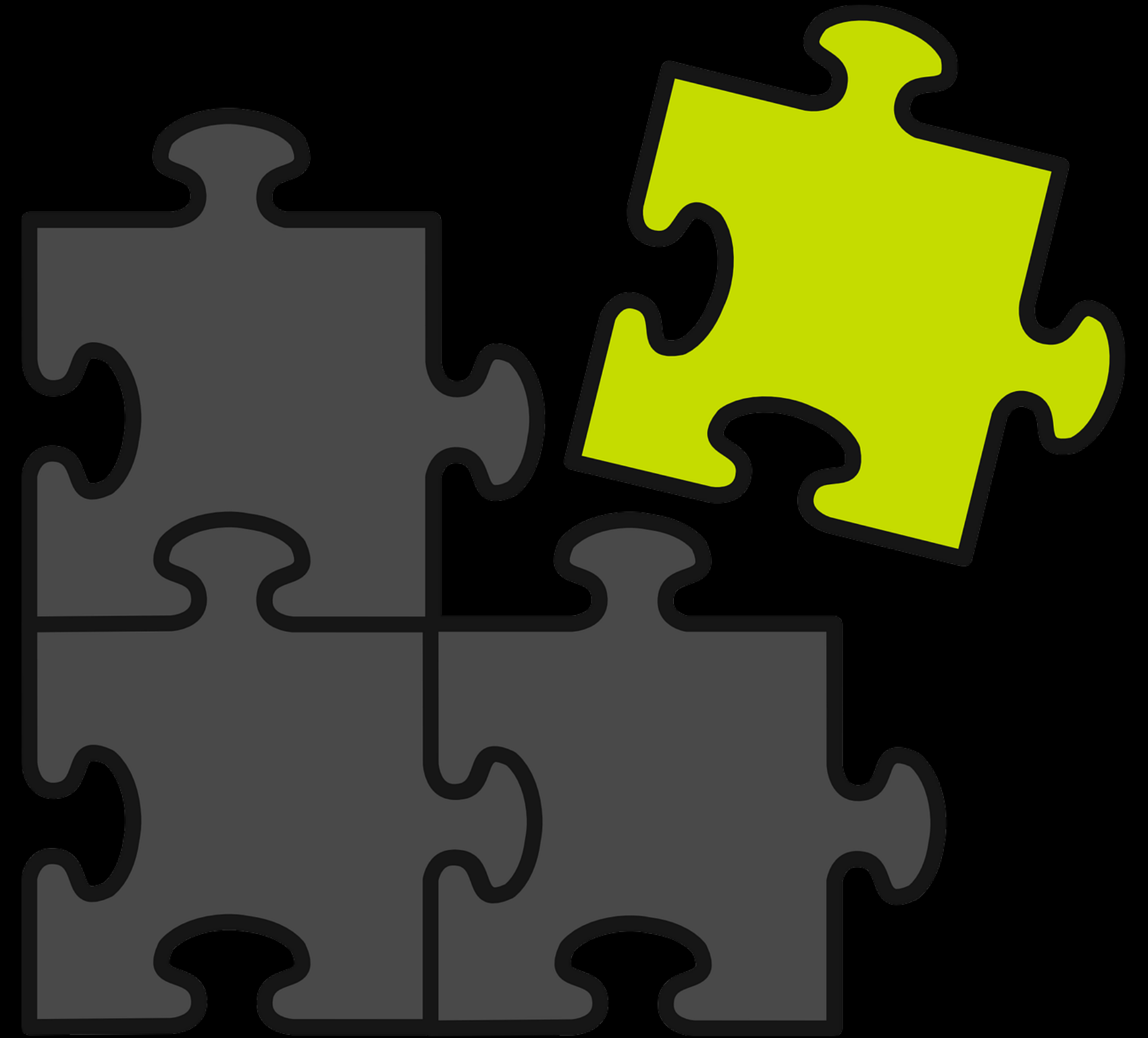
Unit Tests

Testing individual methods



Integration Tests

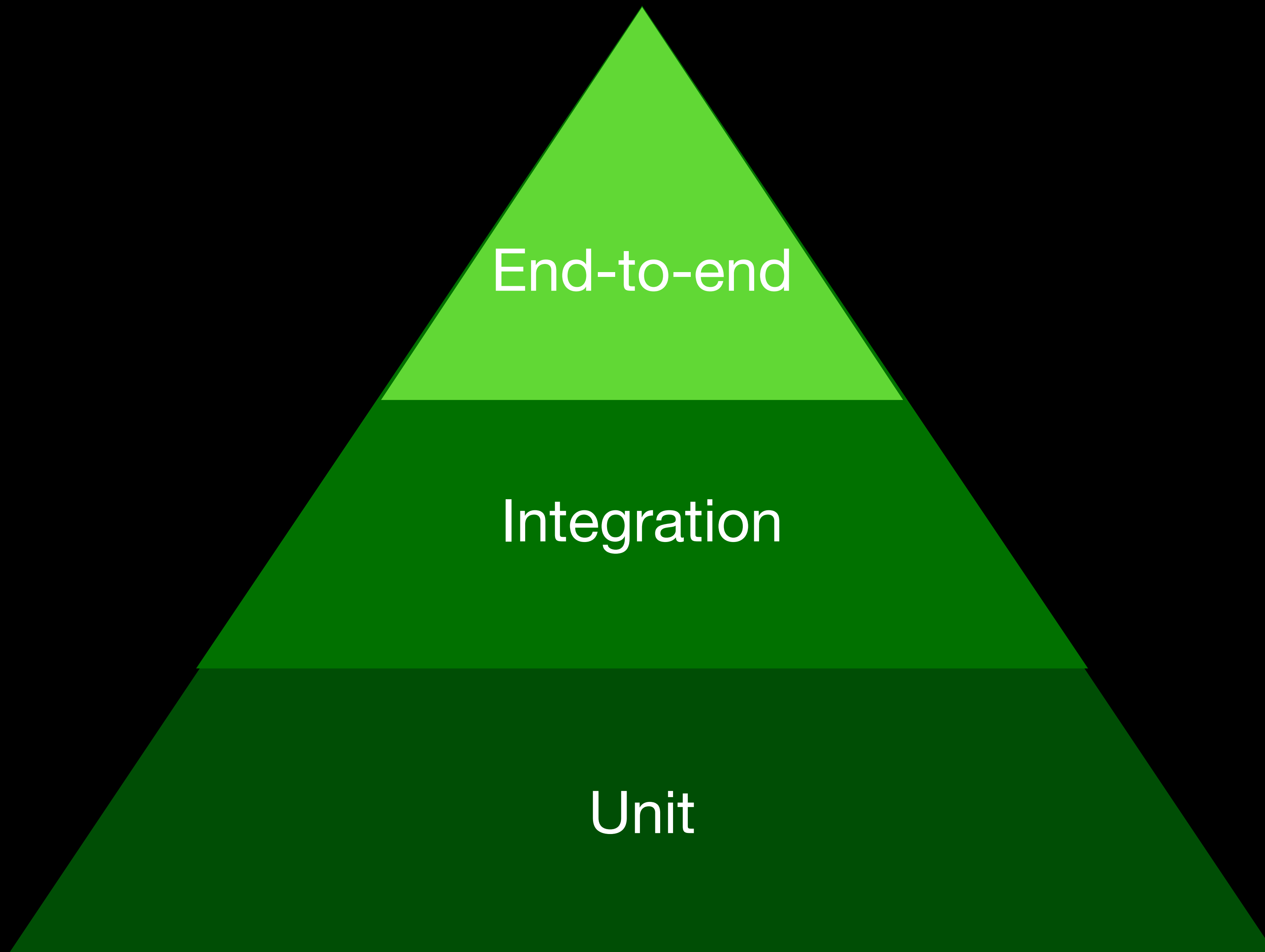
Testing flows



End-to-end Tests

Testing entire experiences





End-to-end

Integration

Unit



Unit

Integration

End-to-end



Unit

Integration

End-to-end



Let's do it! 

Where to start? 🙄



**SHARED
RESOURCE**

Bluetooth





```
protocol BluetoothProtocol {  
    func connect()  
    func disconnect()  
    func sendMessage(_ message: Data)  
    func stateChanged(newState: BluetoothState)  
}
```

```
class MockBluetoothManager: BluetoothProtocol {  
    var currentState: BluetoothState = .disconnected  
  
    func connect() {  
    }  
  
    func disconnect() {  
    }  
  
    func sendMessage(_ message: Data) {  
    }  
  
    func stateChanged(newState: BluetoothState) {  
    }  
}
```

```
class MockBluetoothManager: BluetoothProtocol {
    var currentState: BluetoothState = .disconnected

    func connect() {
        currentState = .connecting
    }

    func disconnect() {
        currentState = .disconnected
    }

    func sendMessage(_ message: Data) {
        print("Sending message: \(message)")
    }

    func stateChanged(newState: BluetoothState) {
        currentState = newState
        print("New state: \(newState)")
    }
}
```




Chapter 3

Next Level Goodness



Chegg



**Requisite photo of
the new stuff**



Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Tests

Group jobs by

Stage

Job dependencies

build

- ✓ mathway:build_for_testing ↺
- ✓ study:build_for_testing ↺

test

- ✓ mathway:run_system_test ↺
- ✓ mathway:run_unit_test ↺
- ✓ mathway_trigger_e2e_tests_mr ↺
- ✓ study:run_system_test ↺
- ✗ study:run_unit_test ↺
- ✓ study_trigger_e2e_tests_mr ↺

Downstream

- ✓ Mobile-Automation #853683510
Multi-project >
- ✗ Mobile-Automation #853681946
Multi-project ↺ >

Pipeline Needs Jobs 8 Failed Jobs 1 Tests 1264

Group jobs by Stage Job dependencies

build

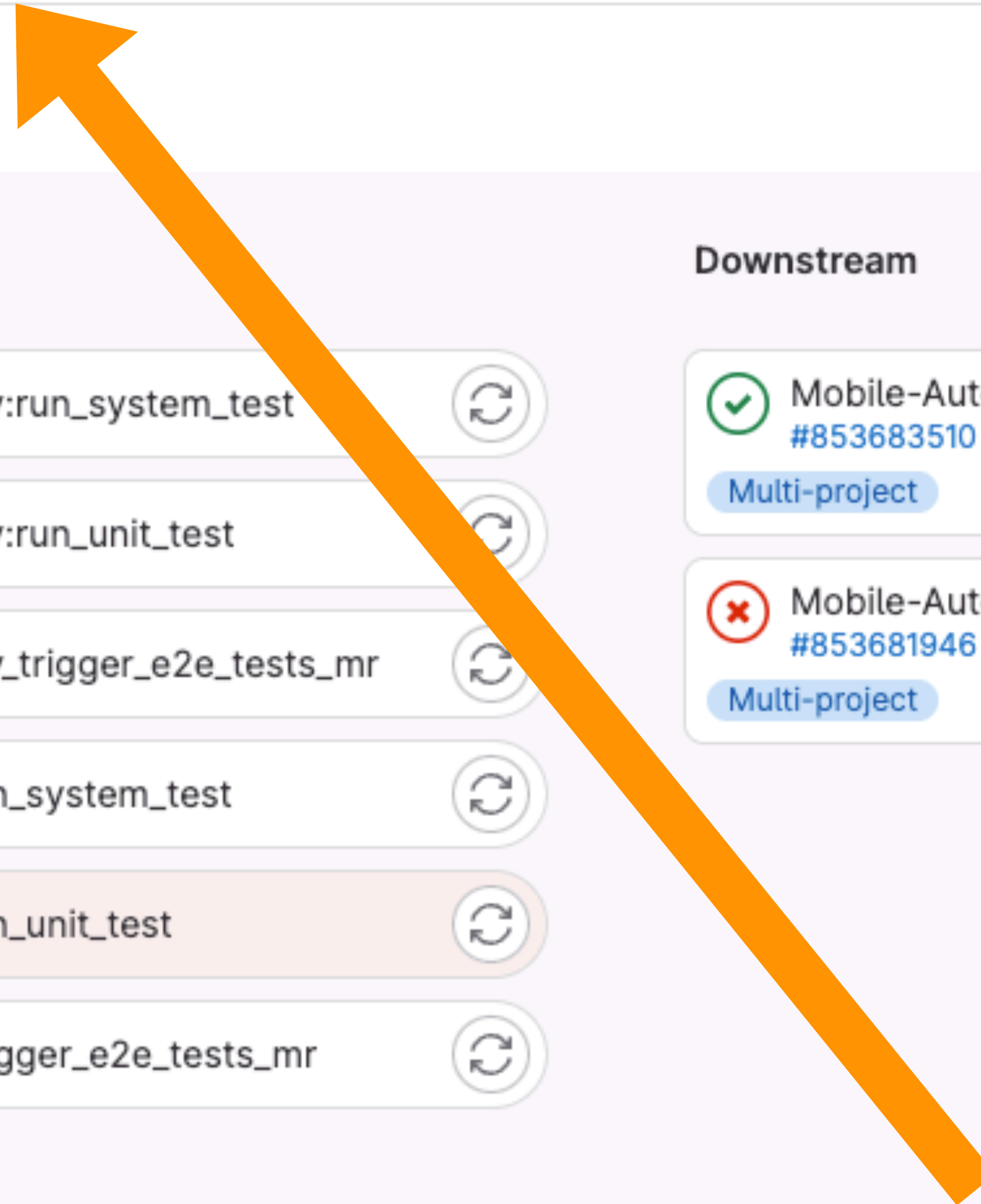
- ✓ mathway:build_for_testing
- ✓ study:build_for_testing

test

- ✓ mathway:run_system_test
- ✓ mathway:run_unit_test
- ✓ mathway_trigger_e2e_tests_mr
- ✓ study:run_system_test
- ✗ study:run_unit_test
- ✓ study_trigger_e2e_tests_mr

Downstream

- ✓ Mobile-Automation #853683510
Multi-project
- ✗ Mobile-Automation #853681946
Multi-project



CI/CD:

Fastlane

Bitrise

GitHub Actions

ios-e2e-automation-reports 🔔



Automation Reports APP 10:22 AM

GitLab job [triggered_smoke](#) finished

Application info: study iOS

Launched by: [@Alon Zilbershtein](#)

Branch name: feature/NPX-313-icons-template

Test Run summary:



Total test run: 35



Passed = 29



Failed = 2



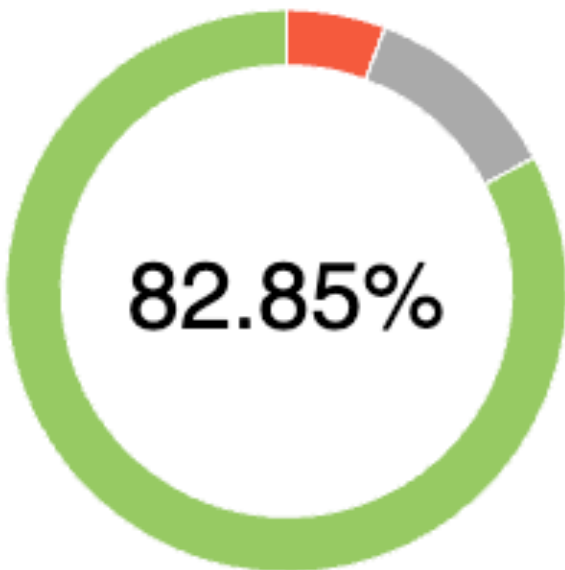
Skipped = 4

To view details follow the [link with report](#)

ALLURE REPORT 4/25/2023
9:07:23 - 9:22:43 (15m 20s)

35

test cases

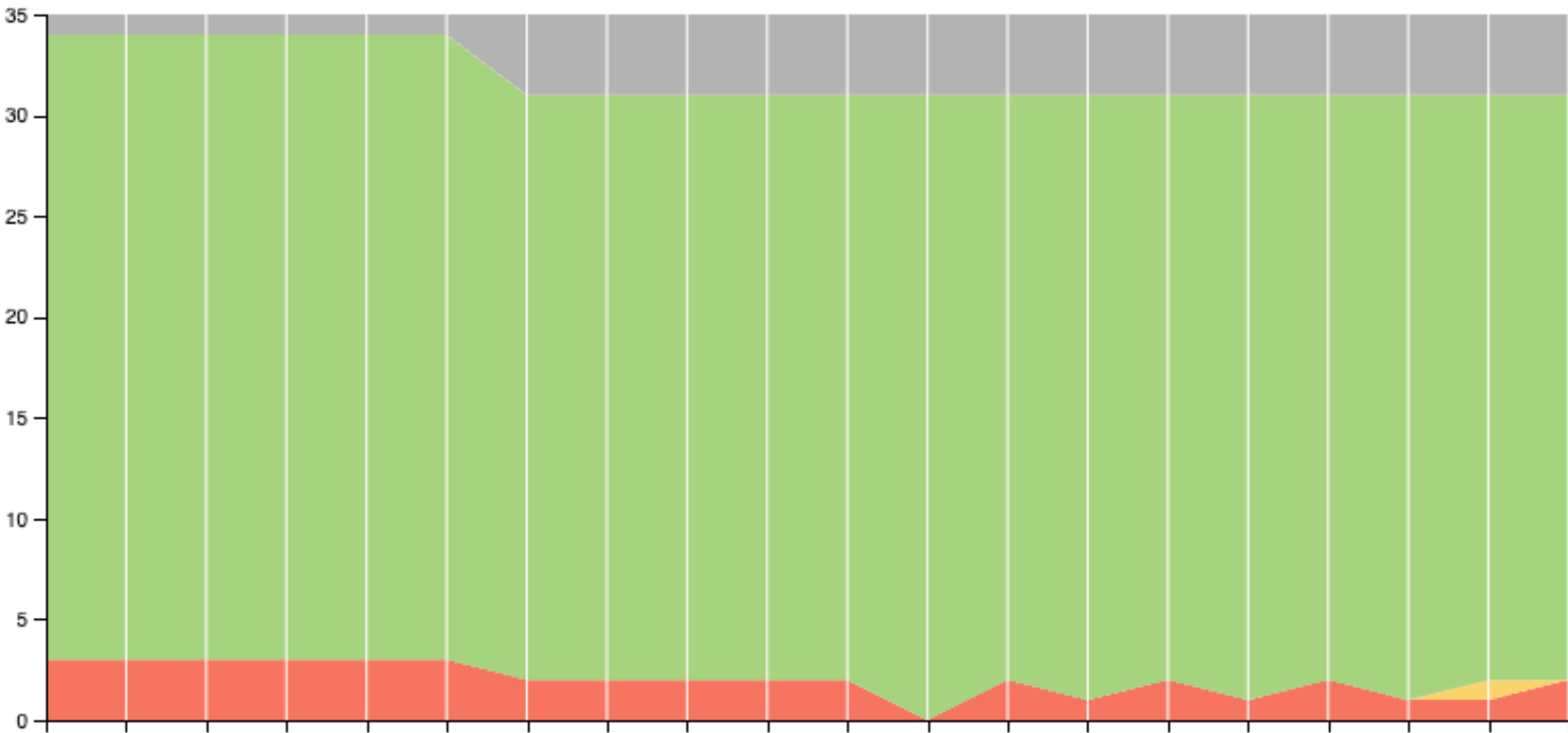


SUITES 13 items total

| | | |
|---|---|---|
| com.hegg.study.testRun.StudyPNQTestsIT | 2 | 1 |
| com.hegg.study.testRun.StudyFlashCardsTestsIT | 5 | |
| com.hegg.study.testRun.StudySignInUpTestsIT | 4 | |
| com.hegg.study.testRun.StudyHomeTestsIT | 3 | 1 |
| com.hegg.study.testRun.StudySearchTestsIT | 3 | |
| com.hegg.study.testRun.StudyToolsTestsIT | 3 | |
| com.hegg.study.testRun.StudyCourseTestsIT | 2 | 2 |
| com.hegg.study.testRun.StudyTBSTestsIT | 2 | |
| com.hegg.study.testRun.StudyExamPrepTestsIT | 2 | |
| com.hegg.study.testRun.StudyQnATestsIT | 2 | |

Show all

TREND



CATEGORIES 3 items total

| | |
|-----------------|----|
| Product defects | 2 |
| Passed | 29 |
| Ignored tests | 4 |

Show all

EXECUTORS

 Testing Culture

 **Warning**

**“Be careful what you incentivize,
because that’s what you’re going to get”**



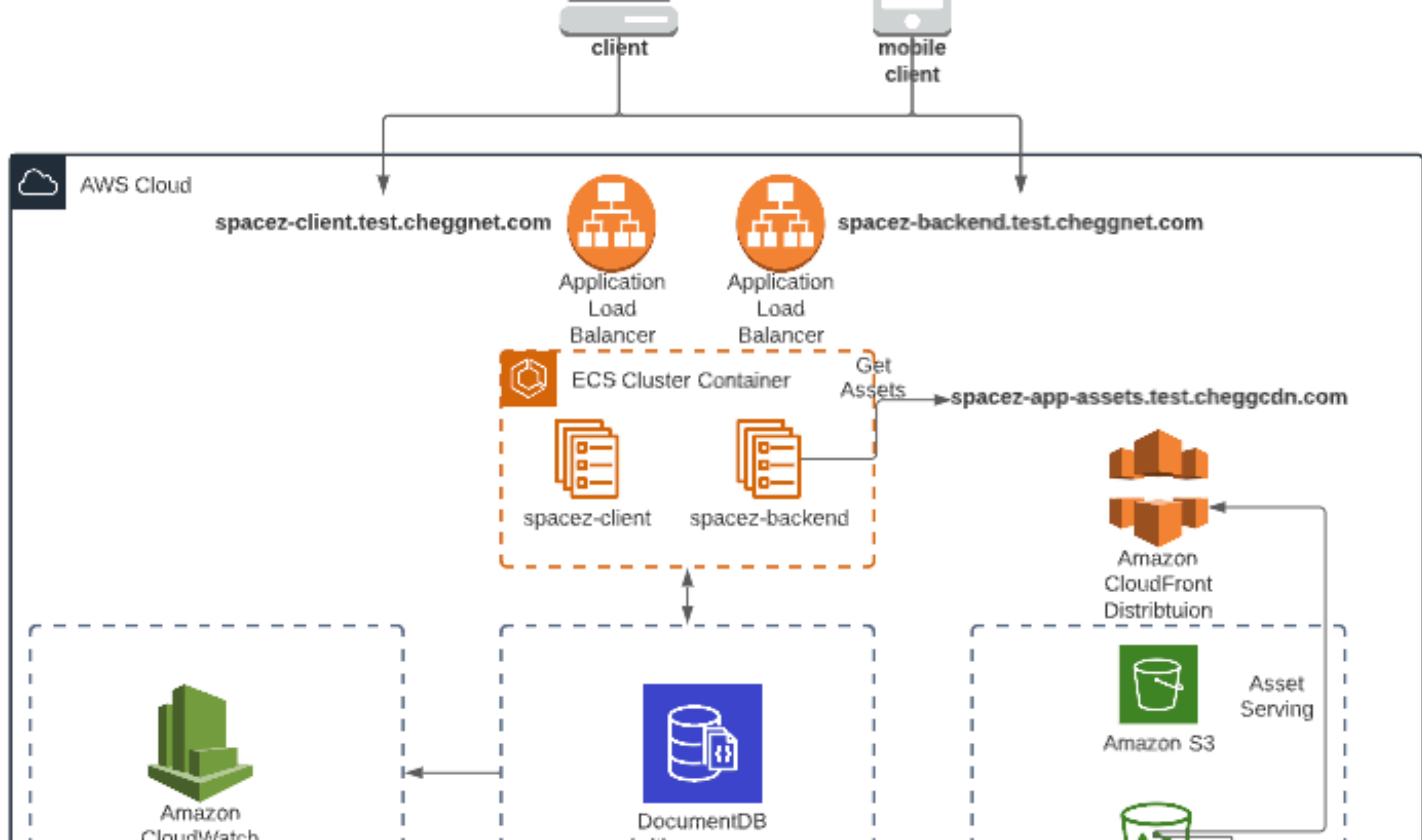
Hackathon







ACE YOUR SPACE!



Which place
do you like?



Duration (Hours):



3



How many people:



4



Special features

Snooker



FIND ME A SPACEZ



Home



Spaces



My Spacez



Assistant



Music room

4 People



Near the meeting room on the north, Near the
meeting room on the east, Near the meeting room.

Sunday, Jan 31, 2020

Site Capacity: 42/100, 42%



Start time

End time

How many people:



4



RESERVE



Houston,
we have a problem...

Something went wrong, but we are
working on a solution.

Back to homepage

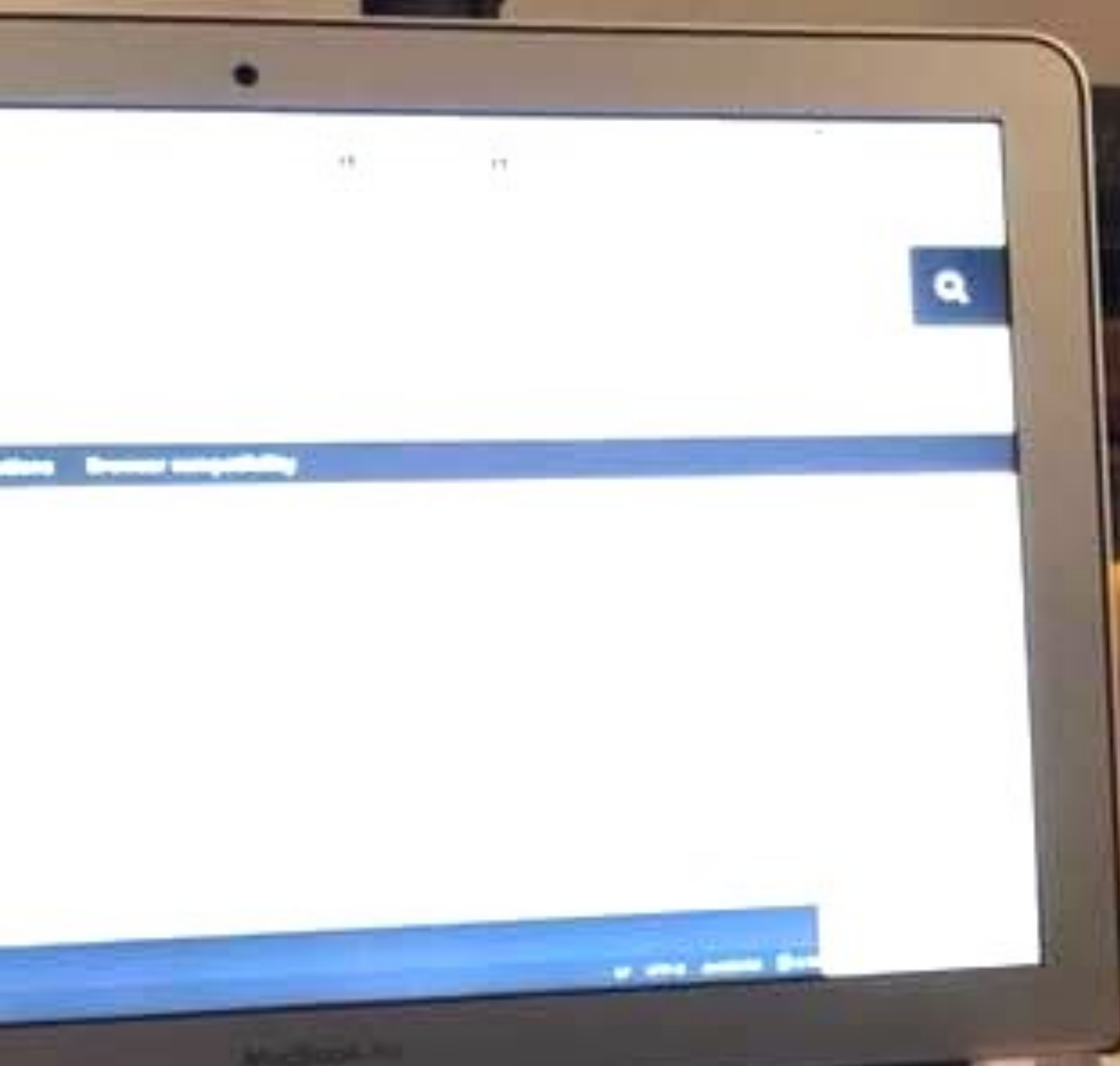
TDD

Tequila

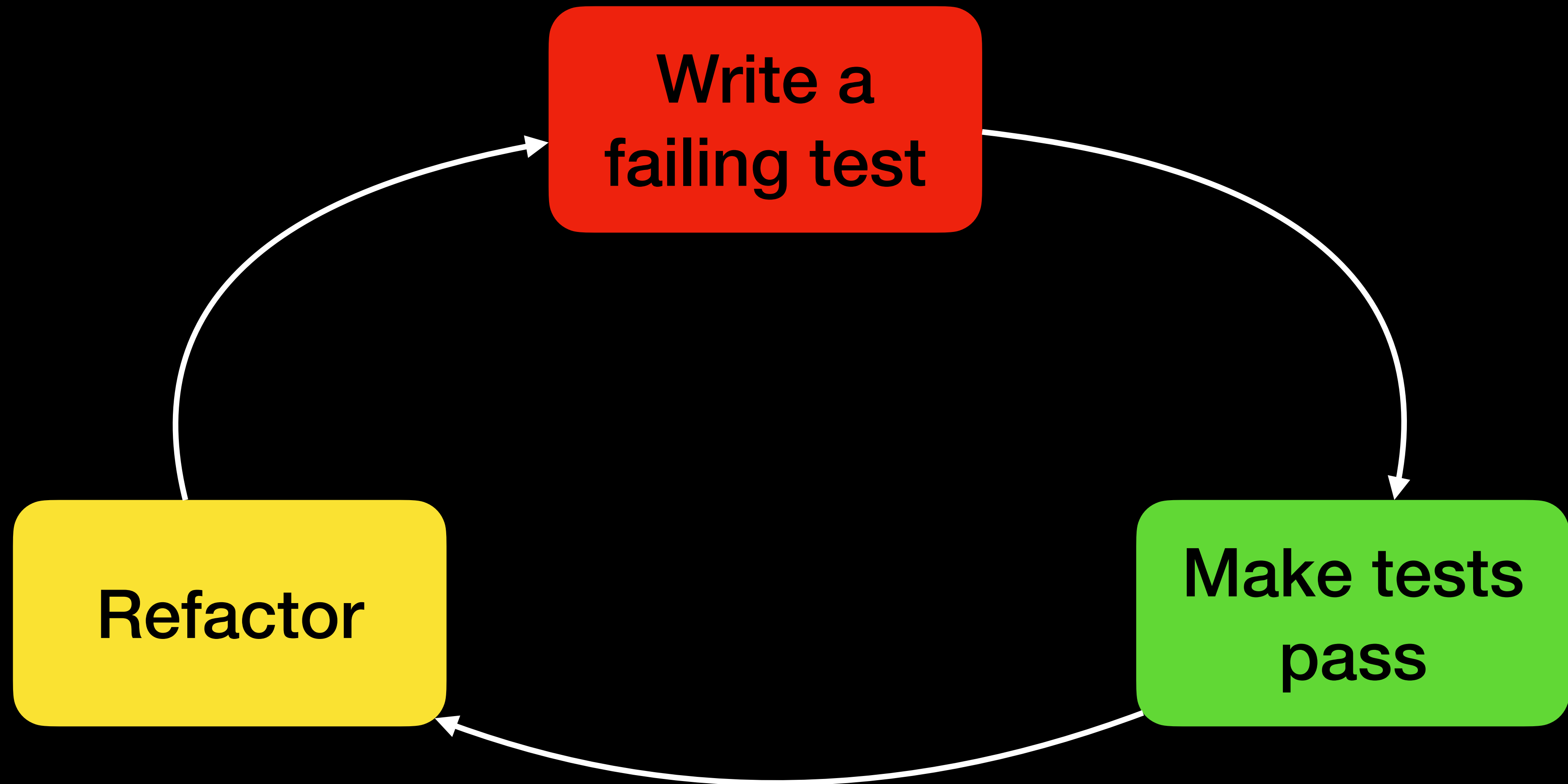
Driven

Development





Test Driven Development



Unblocking



MOCK ALL OF THE THINGS



Mocking:

OCMock

Cuckoo

Mockito

Mockingbird

loser

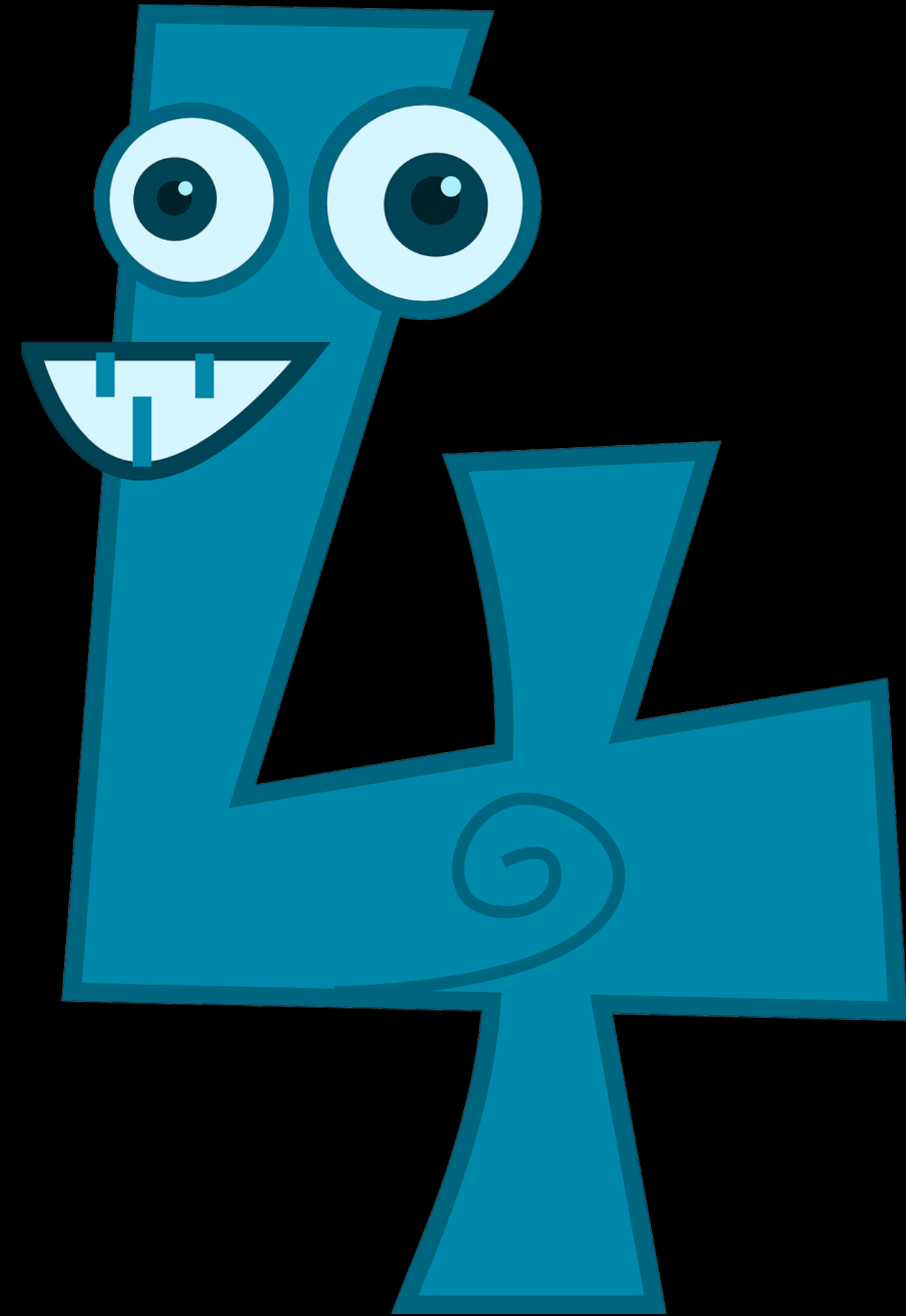
**Achievement
Unlocked**



When not to test?

Chapter 4

New Technologies Ahead



Swift Testing



AI Testing



Chapter 5

Summing it All Up



Why am I giving this talk?

*a.k.a. **WIIFM***

Bottom line:

Testing makes you a better developer



**KEEP CALM
AND
WRITE TESTS**

