

# Testing 101

```
let author = "Konstantin Portnov"  
let github = "github.com/x0000ff"
```

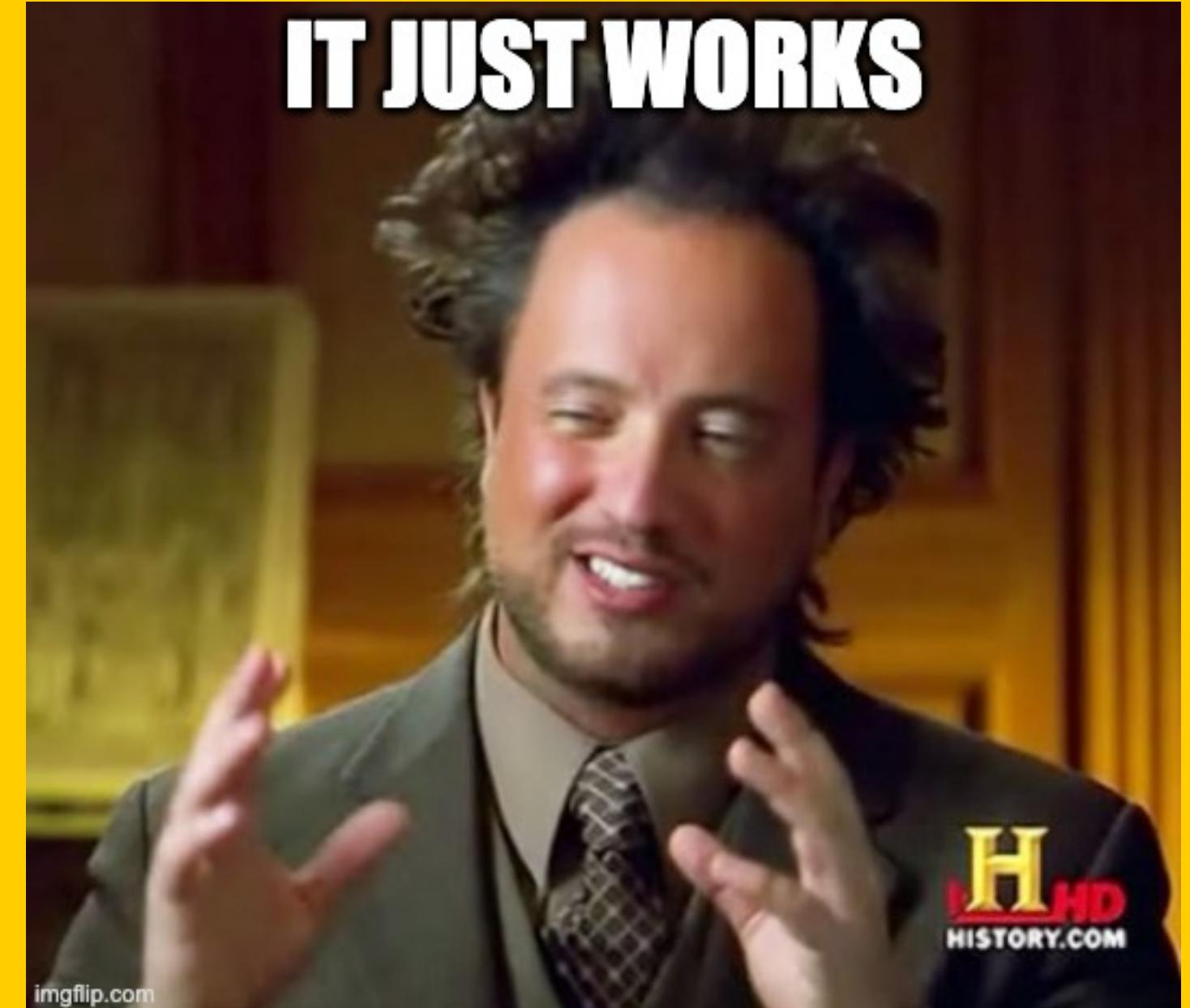


# First things first

1. Ask at any time
2. Yes, I will share the slides
3. Yes, I know that you know about tests

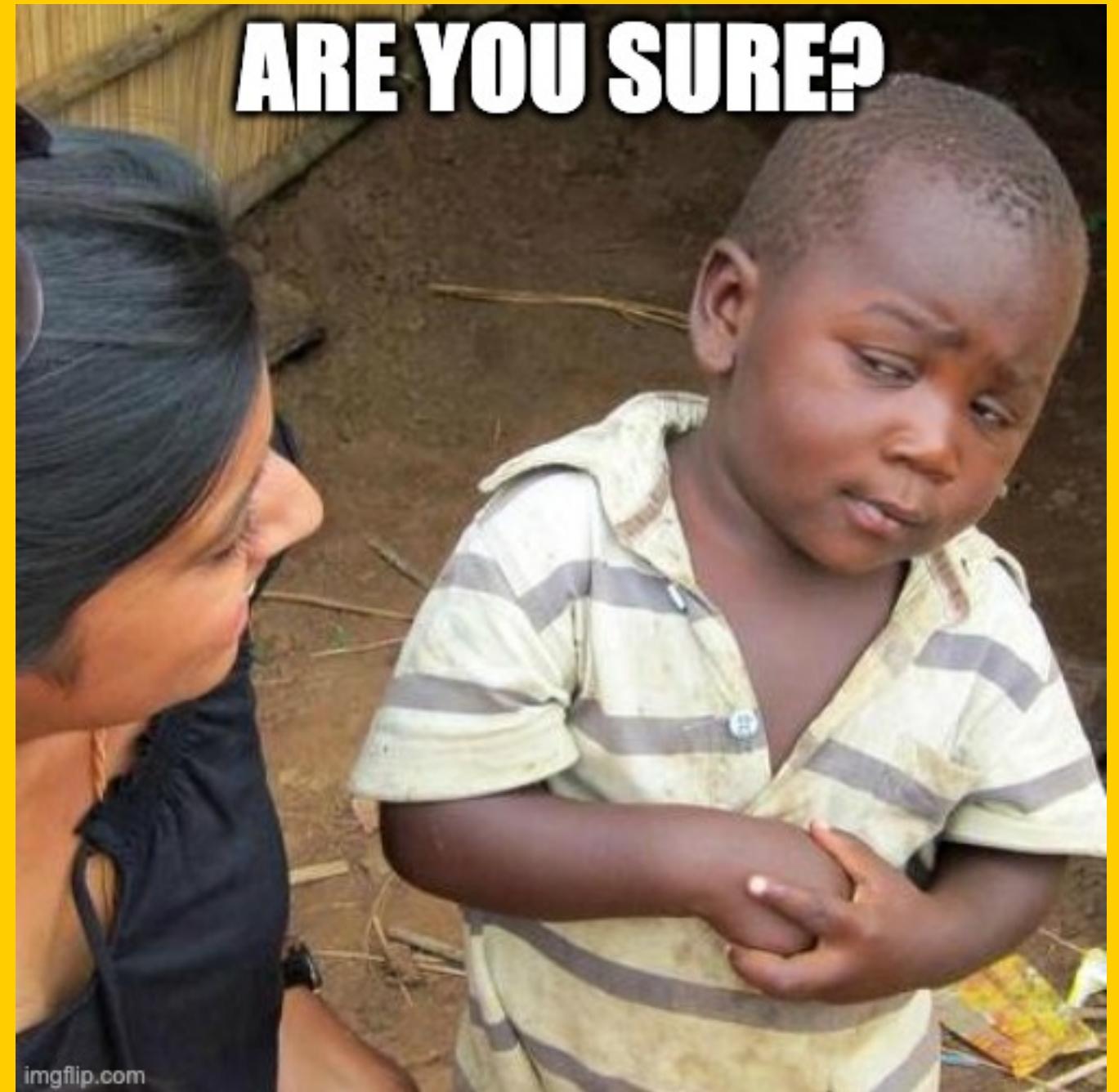
**If you didn't  
test how do you  
know it  
works?...**

**IT JUST WORKS**



imgflip.com

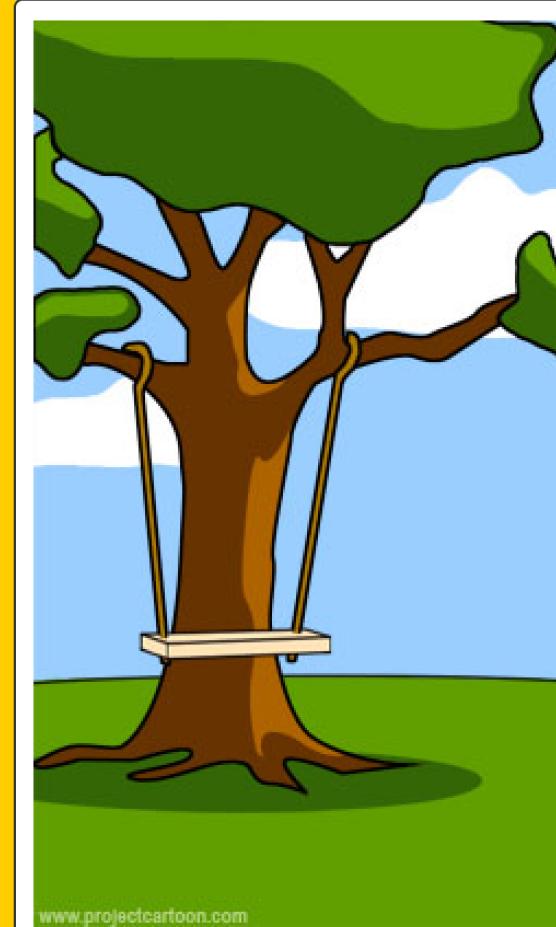
**Do you  
understand  
what needs to  
be done?**



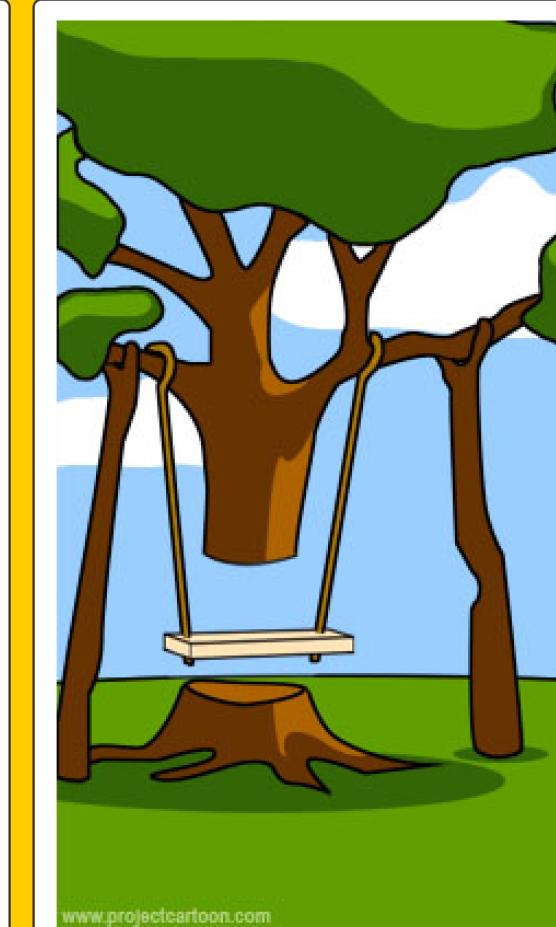
imgflip.com



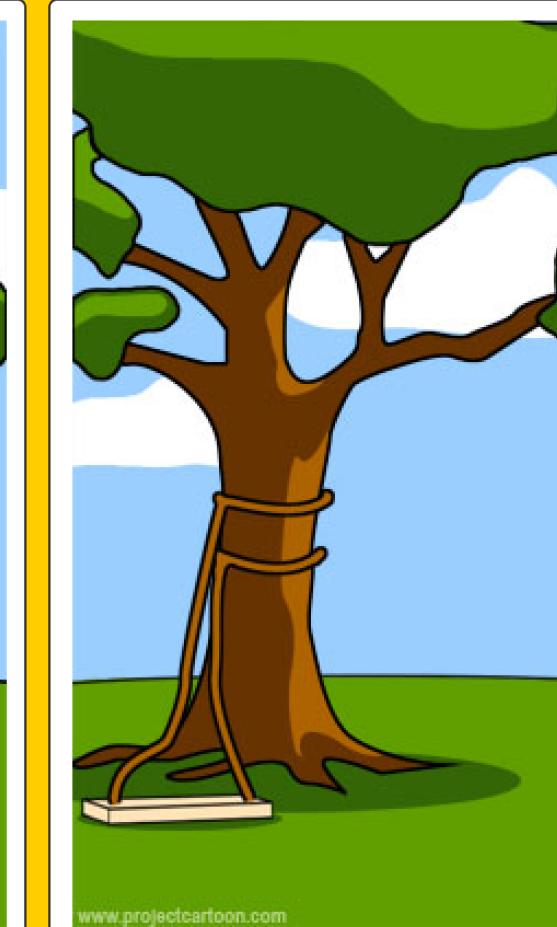
**WHAT CLIENT DESCRIBED**



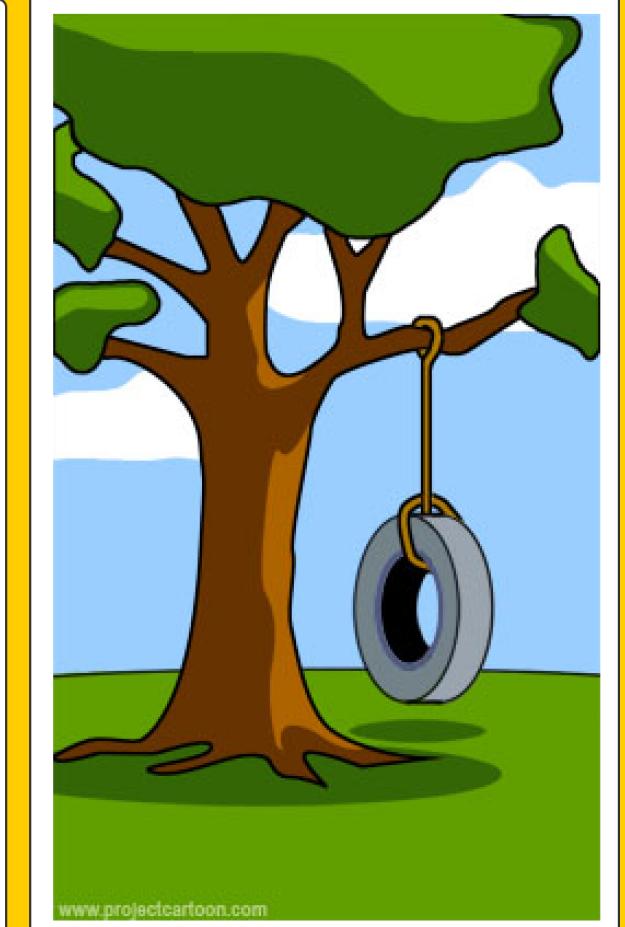
**WHAT PROJECT MANAGER  
UNDERSTOOD**



**WHAT DESIGNER MADE**



**WHAT PROGRAMMERS  
WROTE**



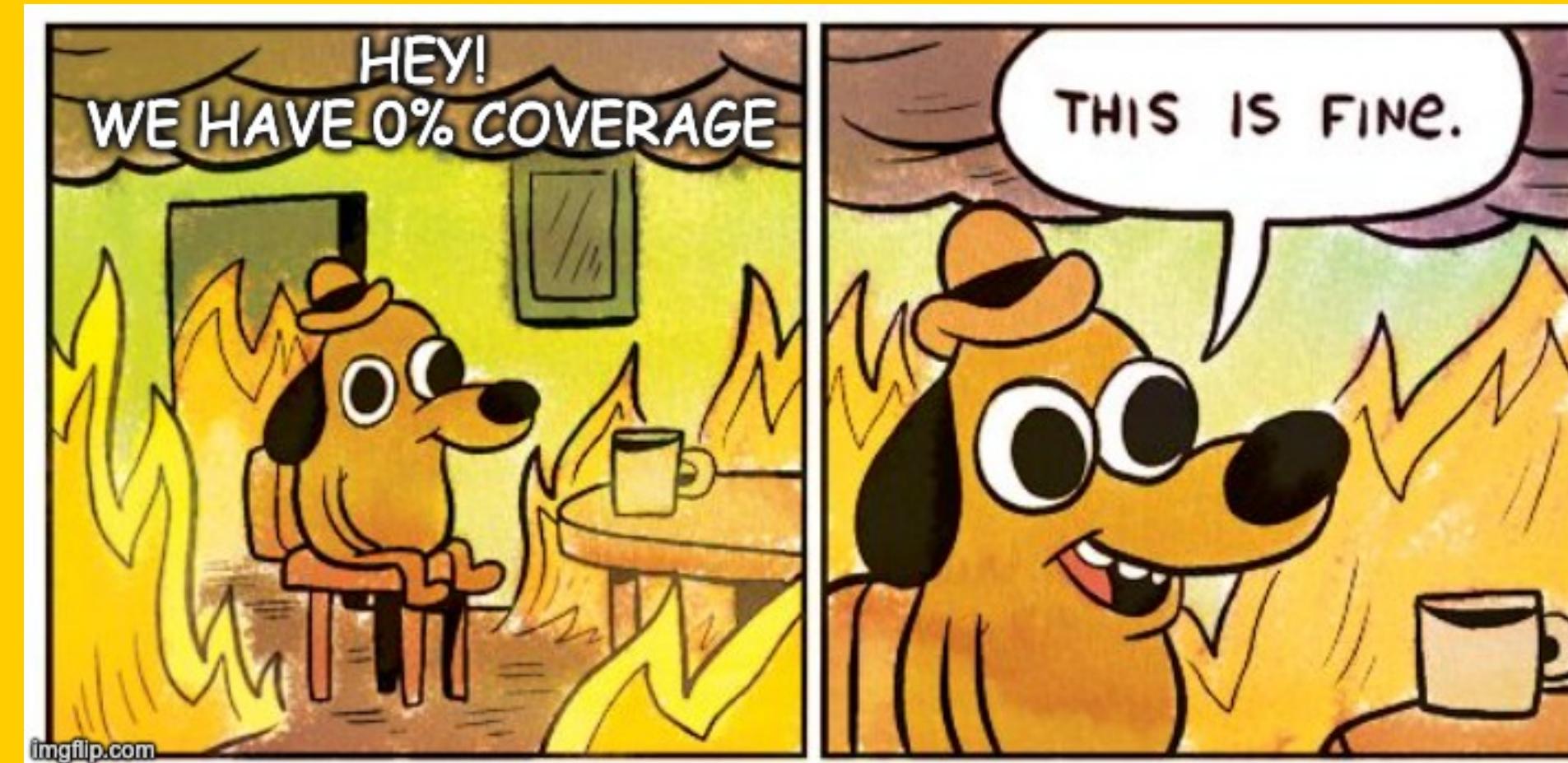
**WHAT CLIENT WANTED**

# REQUIREMENTS

NOTHING

CODE

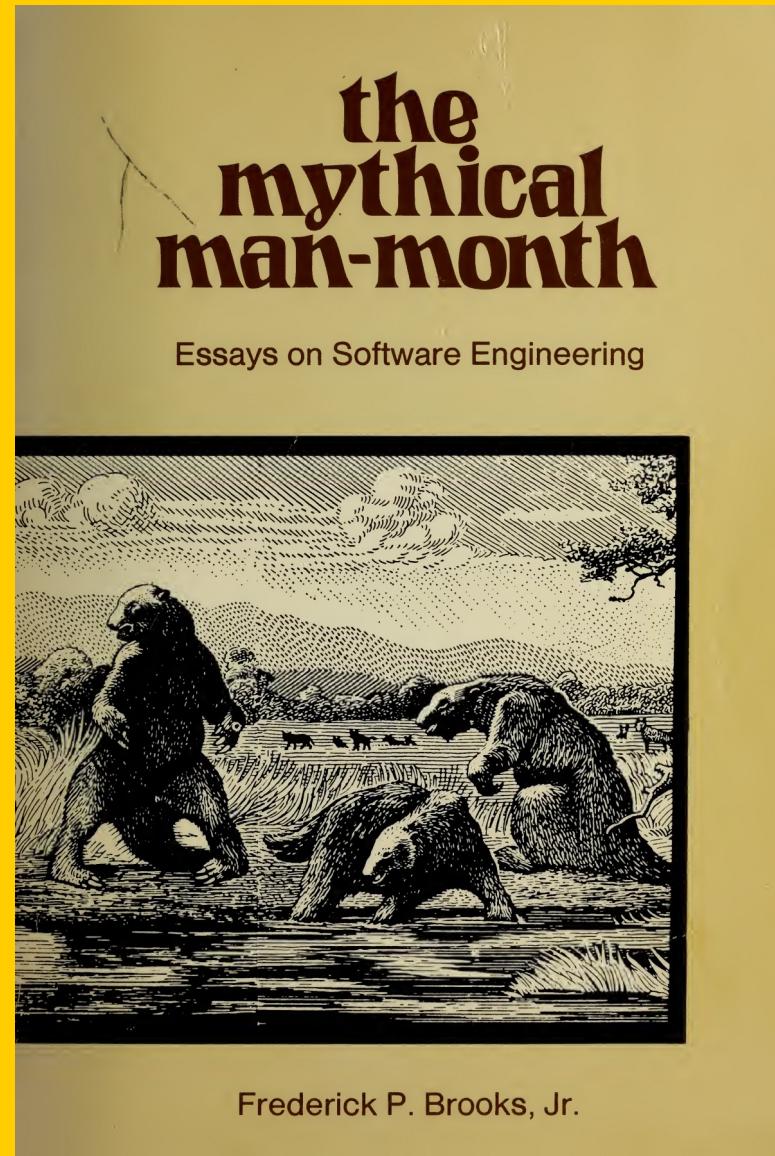




# La Brea Tar Pits



# The Mythical Man-Month





# Why to test?

1. We don't like sh\*tty code





# Why to test?

1. We don't like sh\*tty code
2. To sleep well





# Why to test?

1. We don't like sh\*tty code
  2. To sleep well
  3. Discover requirements
- 
- Dividable by 3 -> "Fizz"
  - Dividable by 5 -> "Buzz"
  - by 3 and 5 -> "FizzBuzz"

*Essential*

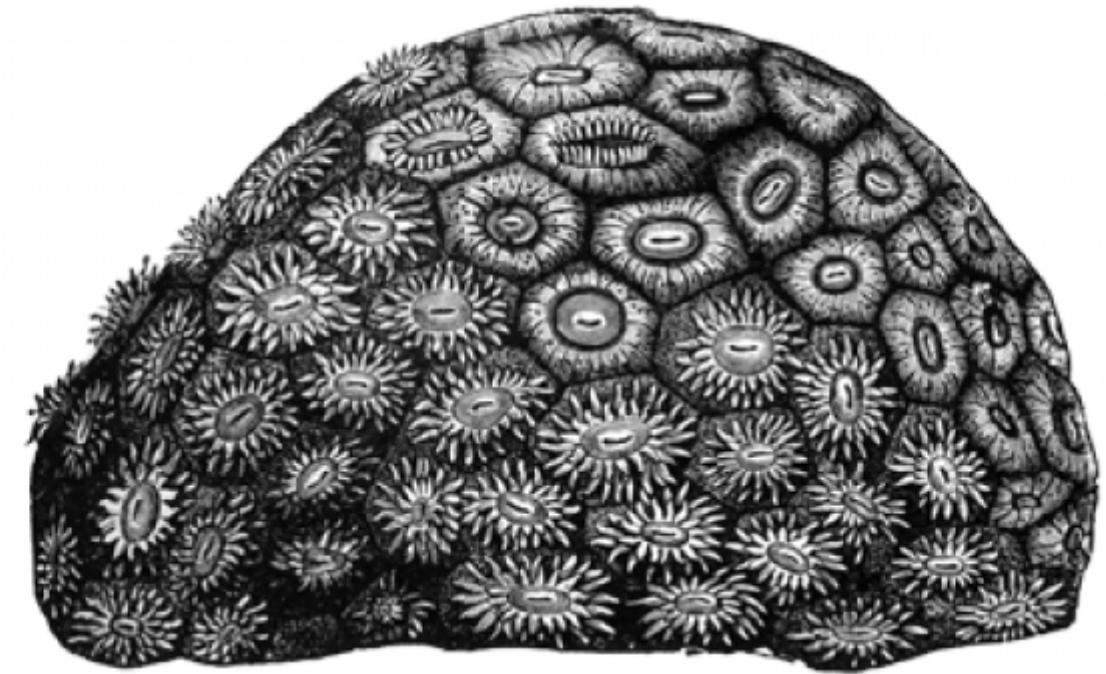
Fizz Buzz





# Why to test?

1. We don't like sh\*tty code
2. To sleep well
3. Discover requirements
4. Documentation



Forgetting How Your  
Own Code Works

*//TODO: Comment*



## Why to test?

1. We don't like sh\*tty code
2. To sleep well
3. Discover requirements
4. Documentation
5. Move fast or die
  - Who said Lotus?

**WHAT'S THE ETA?**



**SOME MANAGER...**



# When to test?

1. You have regression bugs





# When to test?

1. You have regression bugs
2. You are scared to change the code

*Essential*

Hoping This  
Works





# When to test?

1. You have regression bugs
2. You are scared to change the code
3. High cost of failure

# 🙄 When to test?

1. You have regression bugs
2. You are scared to change the code
3. High cost of failure
4. Long Time To Market
  - Mobile Apps
  - Desktop Apps
  - Embedded



Turning Coffee  
Into Code

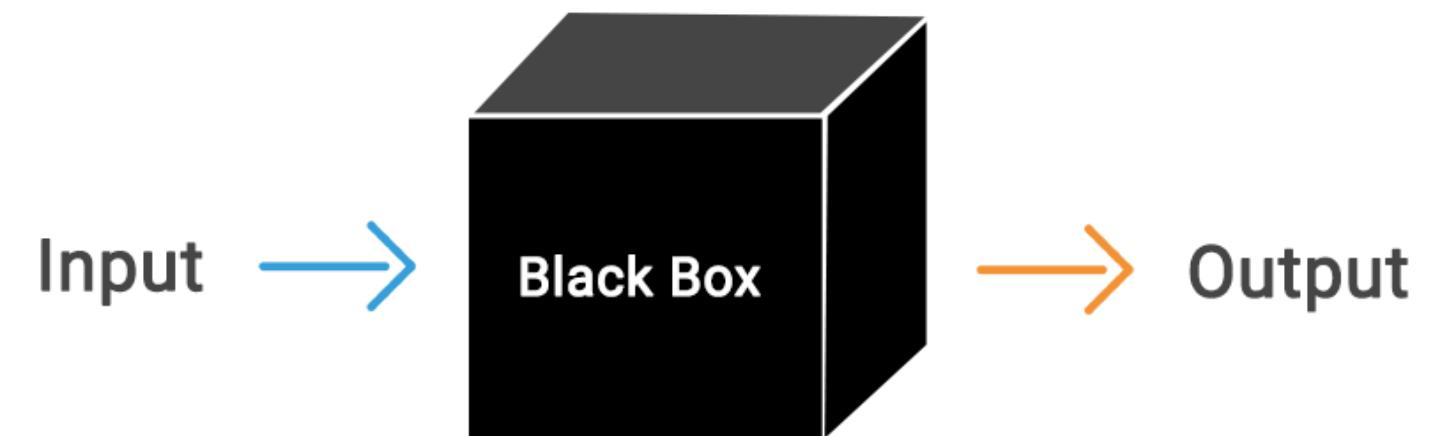
*The Definitive Guide*



# How to test?

## 1. Unit testing

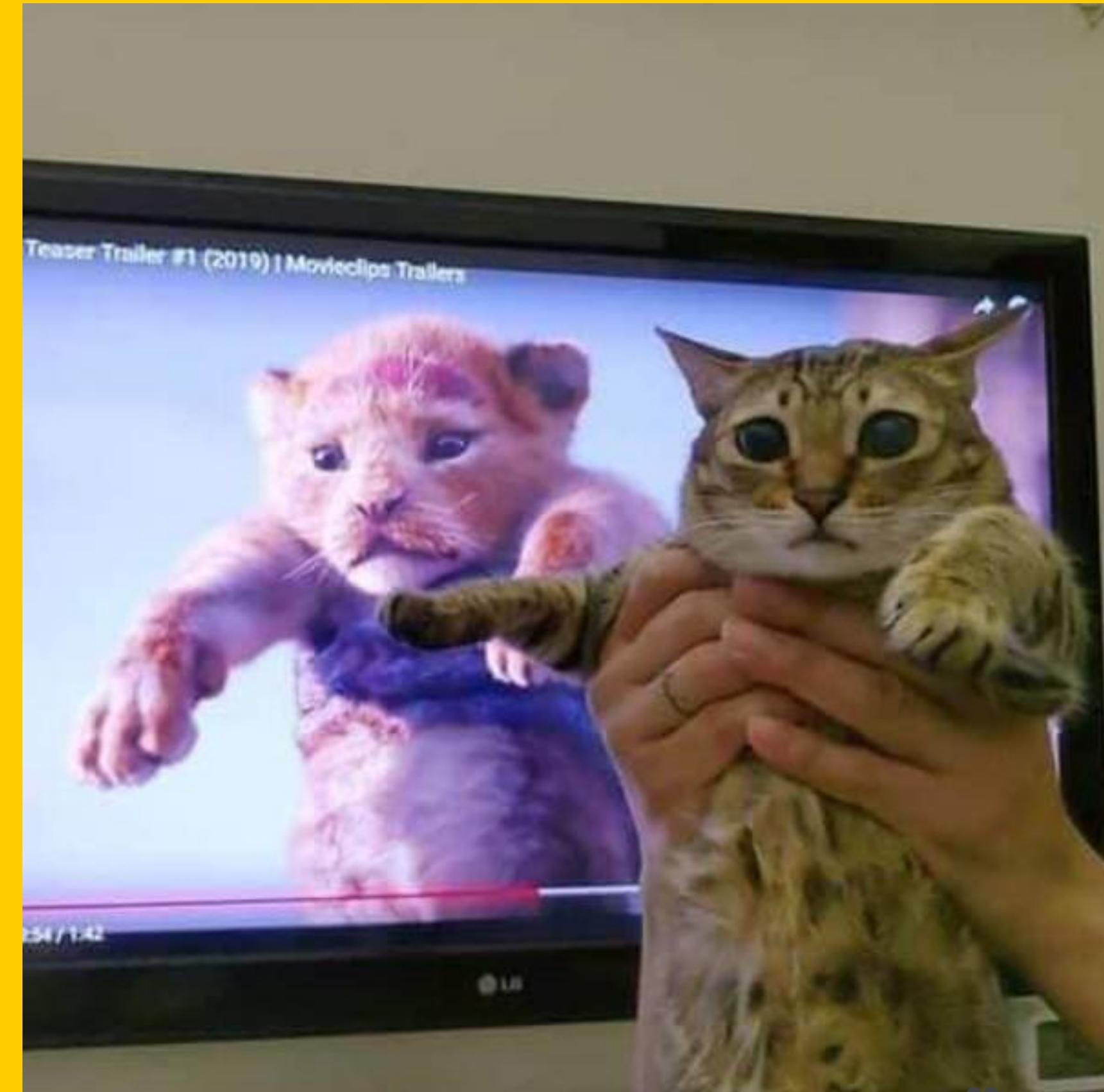
### Black Box Testing





# How to test?

1. Unit testing
2. UI / Screenshot testing





# How to test?

1. Unit testing
2. UI / Screenshot testing
3. Integration testing

LEFT JOIN

RIGHT JOIN

INNER JOIN

FULL OUTER JOIN





# How to test?

1. Unit testing
2. UI / Screenshot testing
3. Integration testing
4. End to End testing





# How to test?

1. Unit testing
2. UI / Screenshot testing
3. Integration testing
4. End to End testing
5. Test Driven Development





**Yes!  
It hurts at the  
beginning!**



**Testing it's like  
investing**



**Tests force you  
to create better code**



**Comment  
something and a test  
should fail**



**100% coverage can  
be 100% fake**

**It doesn't make sense a test which just "executes" the line**

# Simple Code

=

# Simple Test



In S.O.L.I.D. We  
Trust



**Humans shouldn't  
test things which  
machines can test**



**The test doesn't  
find the bug**

**A human finds the  
bug**





**The 1st thing we  
need to do is change  
the way we think**



# Ready for some practice?\*

$1 \text{ UF} = 28716.52 @ 03-06-2020$

\* <https://mindicador.cl/api/uf/DD-MM-YYYY>

# Divide et impera

1. Network operation
  2. Process Network response
  3. Convert
    - 3.1. **UF ↔ CLP**
    - 3.2. **CLP ↔ UF**
  4. Format amount
- ...
- Cache network communication
  - Localization
  - Notify user on errors:
    - Networking
    - Math ( divide by 0 etc.)
  - UI / Layout

1	I	UF
28716,52		CLP

$1 \text{ UF} = 28716.52 @ 03-06-2020$



# Q & A

# Me...

-  Konstantin Portnov
-  <http://about.me/x000ff>
-  <https://github.com/x000ff>
-  <https://twitter.com/x000ff>
-  [https://www.linkedin.com/in/  
KonstantinPortnov](https://www.linkedin.com/in/KonstantinPortnov)





**You can find  
this talk here:**

`http://  
bit.ly/  
3dvFvFQ**`



---

\*\* <https://github.com/x0000ff/Testing101>



**Thanks a lot!**



# EOF

