



# Testing in production

Losing fear, one test at a time

Jorge Marín

October 14th, 2020 – Testcon Europe @ Vilnius Virtual 🌐

[@chipironcin](https://jorgemarin.me)

# Hidden slide

\*breathe\*

# Testing in production

My name is Jorge, and I am an engineer 🖐️🇪🇸

# Poll time - Raise your hands if you...

~~Poll time - Raise your hands if you...~~  
Think about these for a minute



# Think about these for a minute

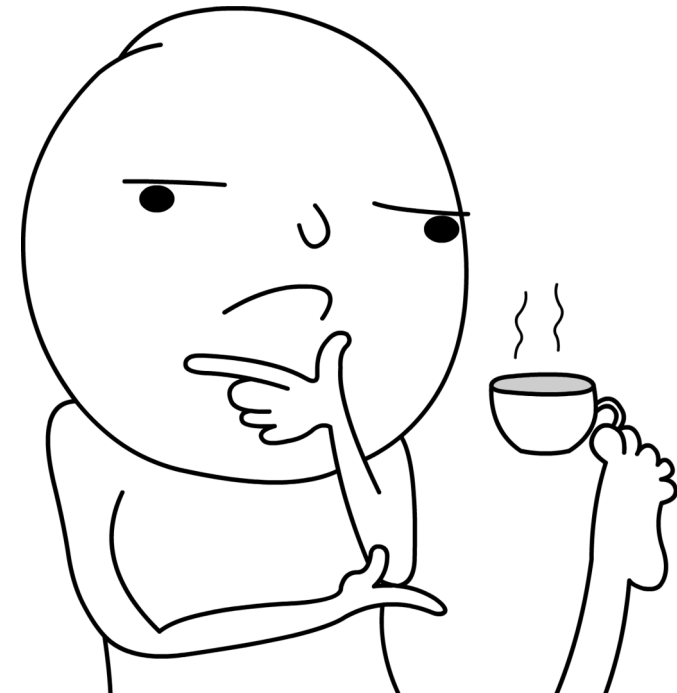
- Are having a good day?
- Do you think you know what testing is?
- Have you ever thought there are more testing types than programming languages out there?
- Testing your services in production
- If something goes “without saying”, why do people still say it?

# What is this talk about

- Testing backend services in production
- Answers to Frequently Asked Questions
- General recommendations
- Based on experience

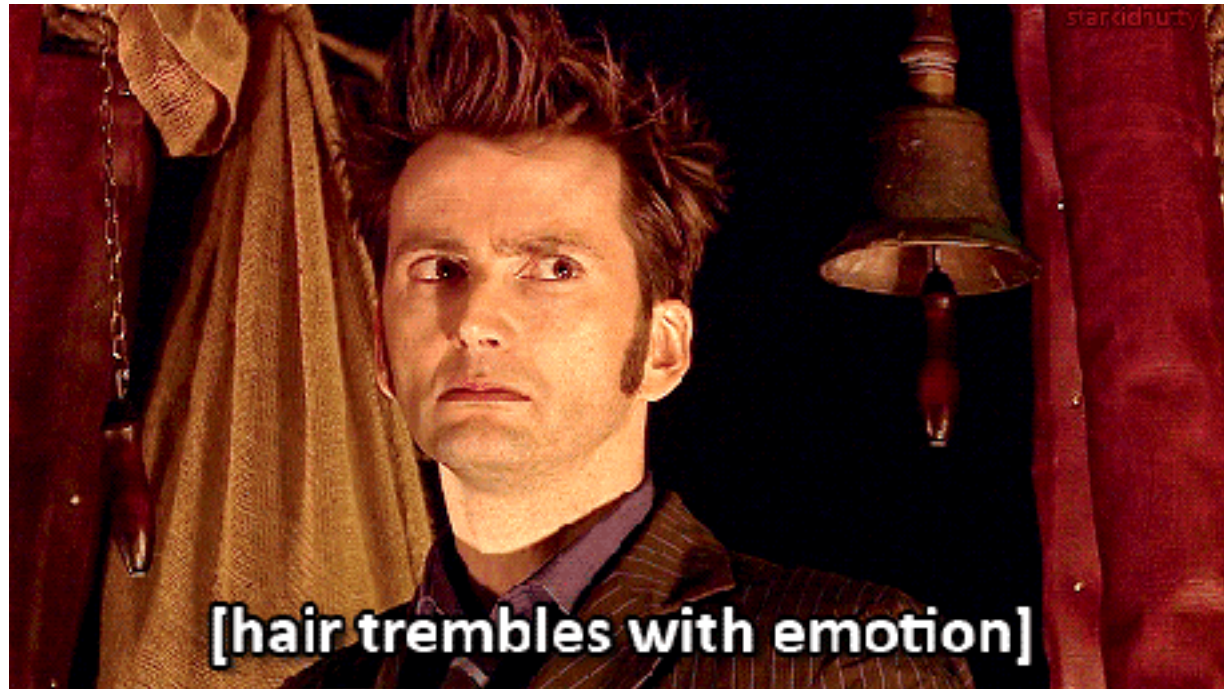
# Main takeaways

- Why do you need tests in production
- What kind of testing is the appropriate for production
- How to build end to end tests
- How to fight dragons





Allons-y!



Why do you need testing in production? #1

What do you mean by “testing in production”?

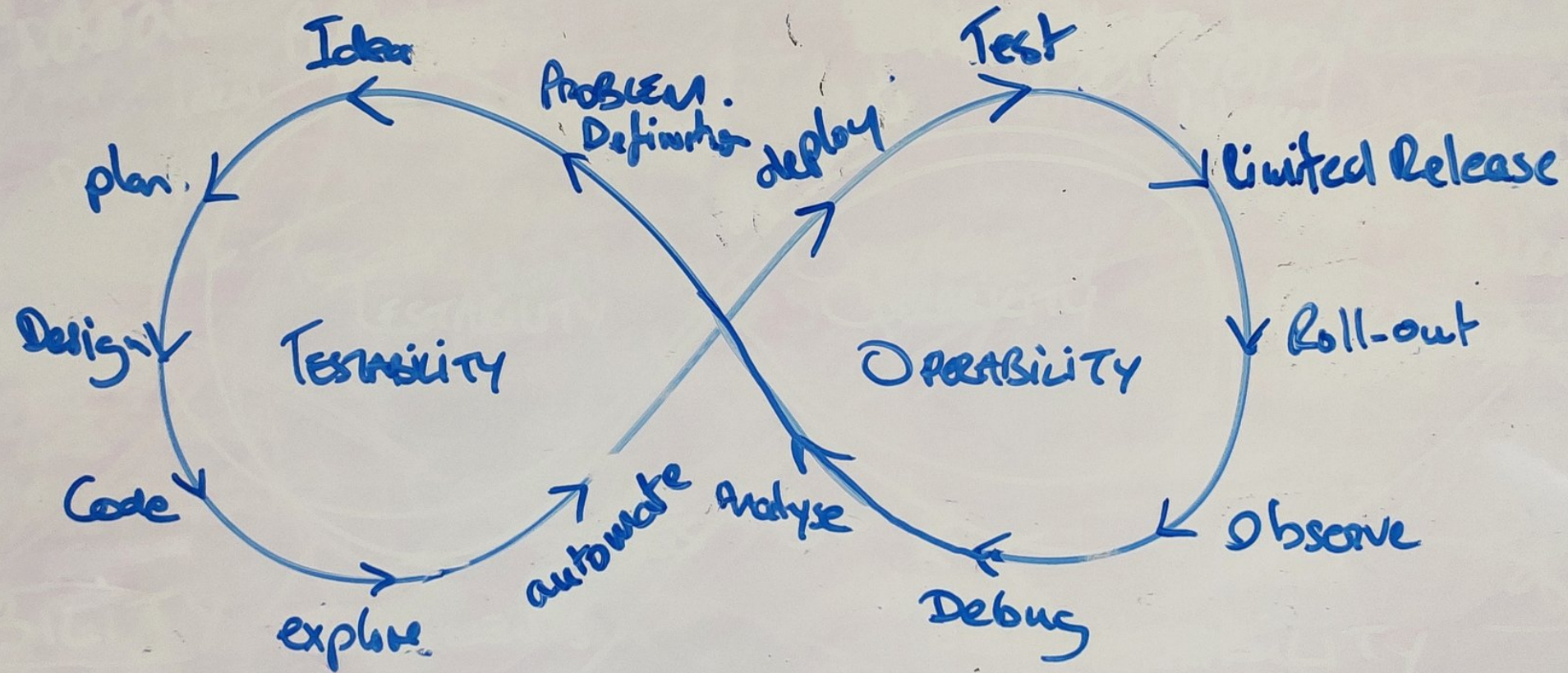
# What do you mean by “testing in production”?

lambda  
service

To guarantee that your **application** continues to work  
consistently after the release/deployment



# MANAGING Risk in A Continuous World!



RISKS we can  
Anticipate

Realities we can not  
Anticipate

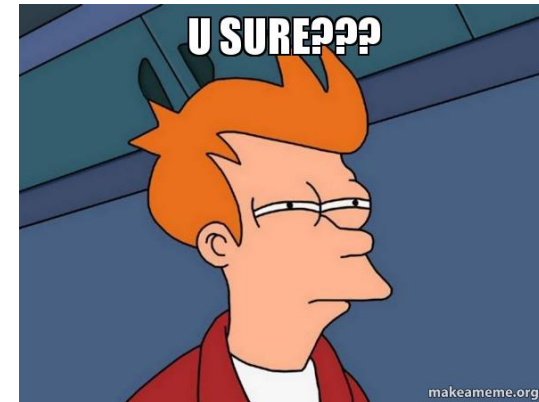
# Why do you need testing in production? #1

- But I have health dashboards that will tell me when something is wrong



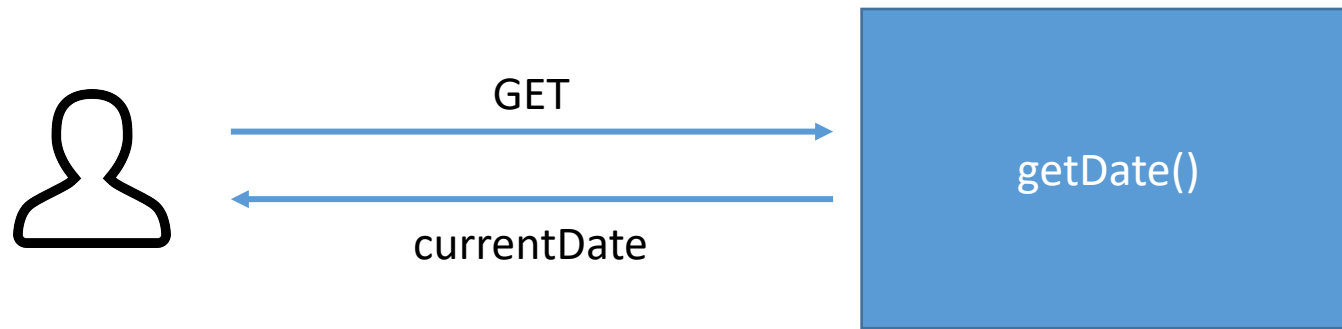
# Why do you need testing in production? #1

- But I have health dashboards that will tell me when something is wrong



# Why do you need testing in production? #1

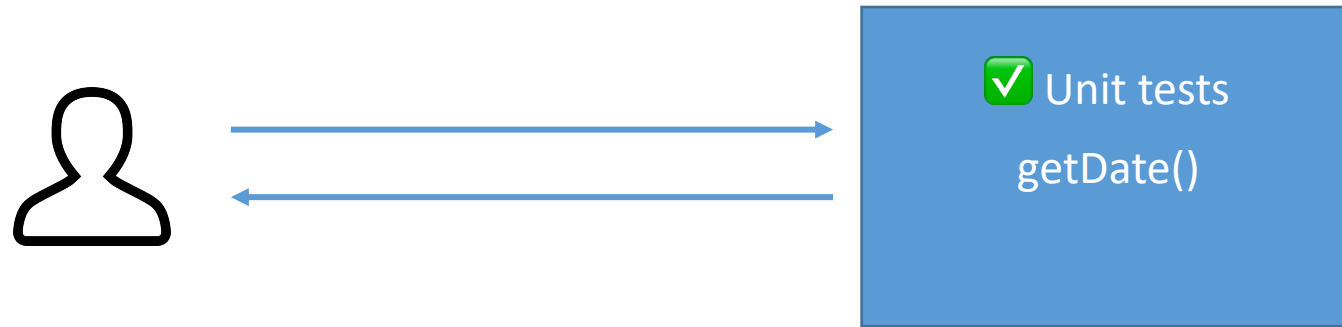
Imagine a function (serverless)





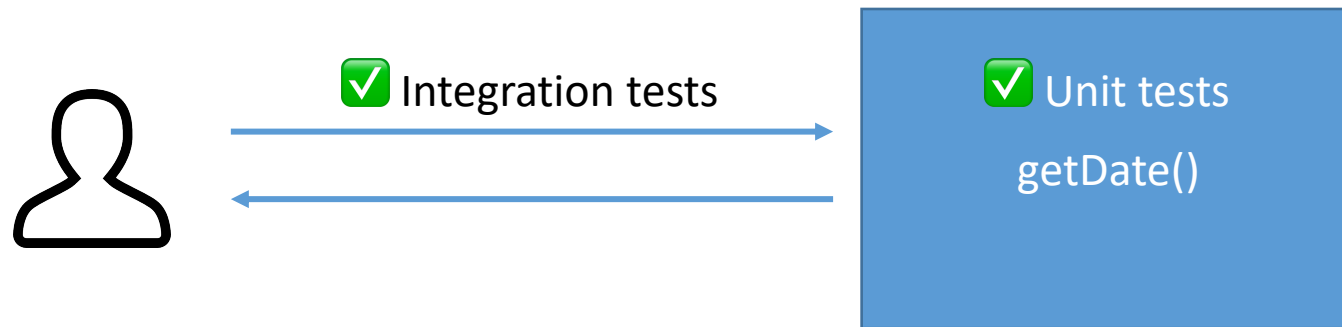
# Why do you need testing in production? #1

Imagine a function (serverless)



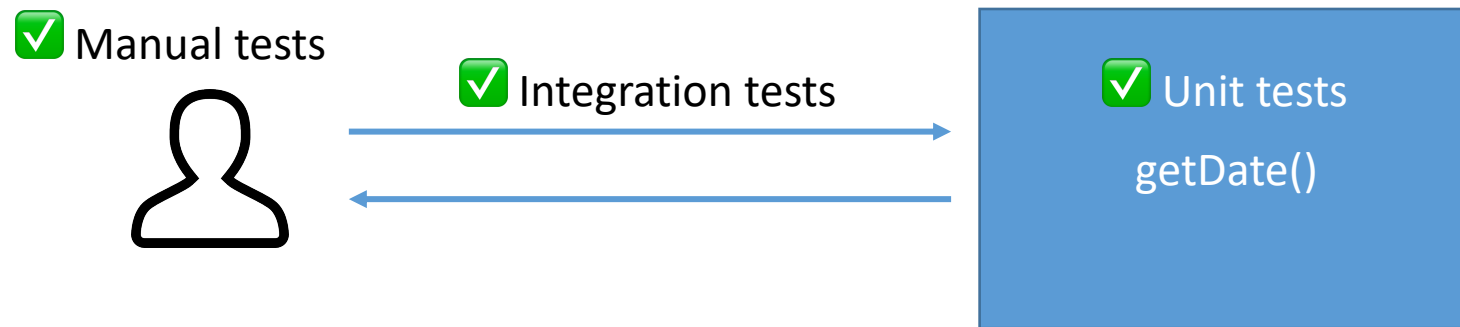
# Why do you need testing in production? #1

Imagine a function (serverless)



# Why do you need testing in production? #1

Imagine a function (serverless)



# Why do you need testing in production? #1

Imagine a function (serverless)

**UNTIL YEAR 2038**



# y2k38

- Storing dates (seconds since Jan 01 1970) as a signed 32-bit binary integer

**00000000000000000000000000000000**

Thursday, 01<sup>st</sup> Jan 1970 00:00:00 UTC  
+0 seconds

y2k38

**0101111100001101101011010101100**

Wednesday, 14-Oct-2020 10:45:00 UTC  
+1602672300 seconds

y2k38

**0111111111111111111111111111111111111111**

????

y2k38

**01111111111111111111111111111111111111**

Tuesday, 19st Jan 2038 03:14:07 UTC  
+2147483647 seconds





y2k38

**100000000000000000000000000000000**

y2k38

**100000000000000000000000000000000**

-2147483648 seconds

y2k38

**100000000000000000000000000000000**

Friday, **13st Dec 1901** 20:45:52 UTC  
-2147483648 seconds

y2k38



# Why do you need testing in production? #1

- Extrapolate to a real service
- If something goes wrong, your customers will make sure you know:
  - Cannot login
  - Cannot register
  - Cannot change profile picture
  - Cannot launch the app
  - Cannot order chocolate milkshakes to takeaway

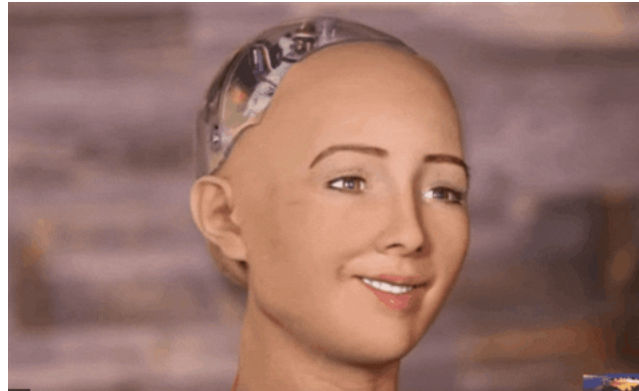
# Why do you need testing in production? #1

Real angry **paying** customers  
calling you  
because what they paid for is not working



# Why do you need testing in production? #1

An automated process  
that acting as a real user  
notices the issue and notifies the team





# Why do you need testing in production? #1

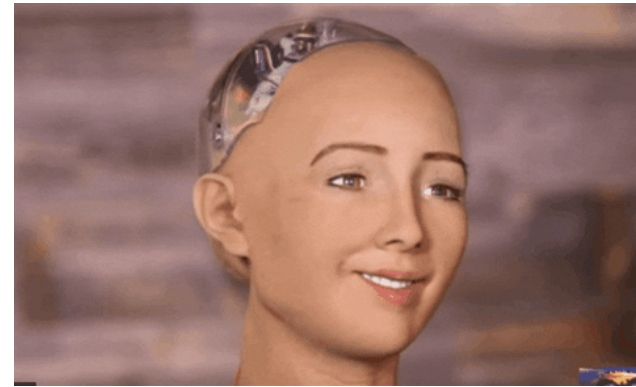
Still not convinced?



# Why do you need testing in production? #1



VS.



# Why do you need testing in production? #1

Still not convinced?

**SPEED**

# Why do you need testing in production? #1

Still not convinced?

**SPEED**

**LOSES**

# Why do you need testing in production? #1

Still not convinced?

**SPEED**

**LOSES**

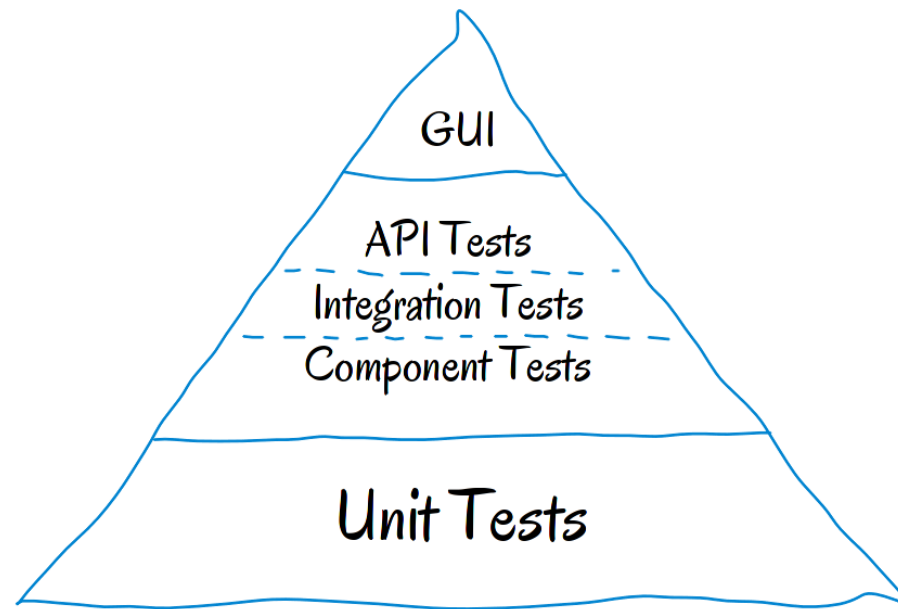
**REPUTATION**

# Why do you need testing in production? #1

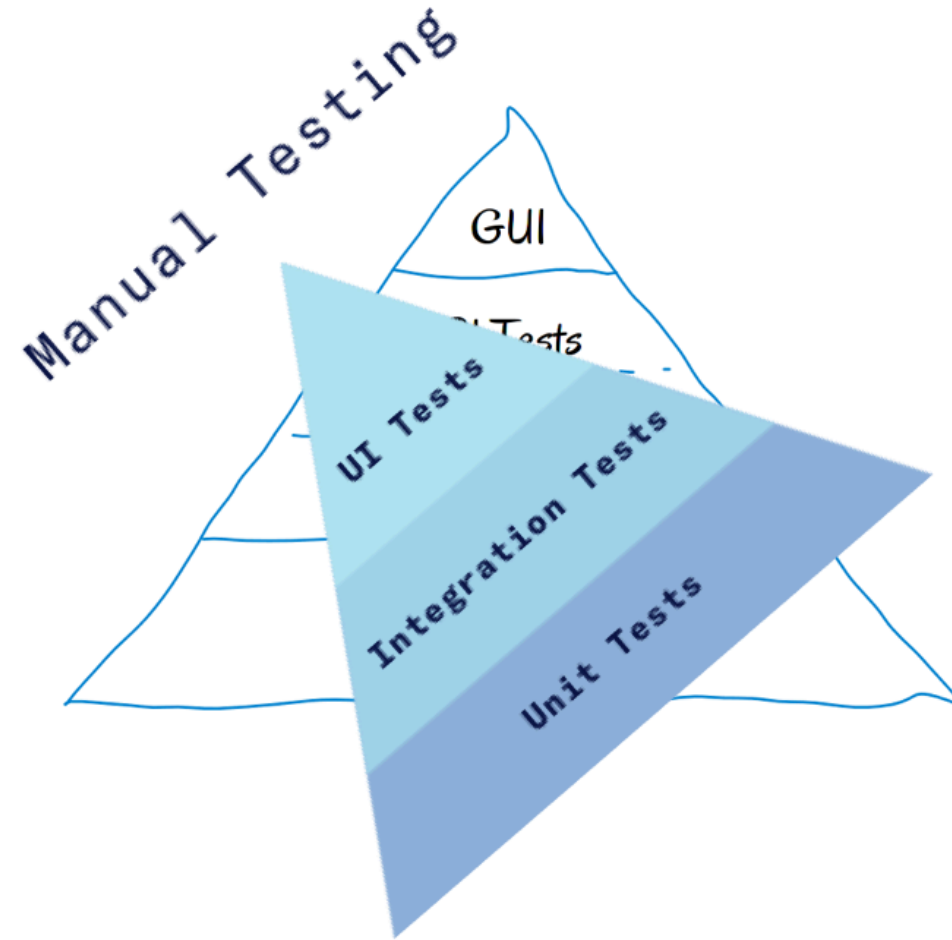
Are you convinced already?



# What is the right testing level against production systems? #2

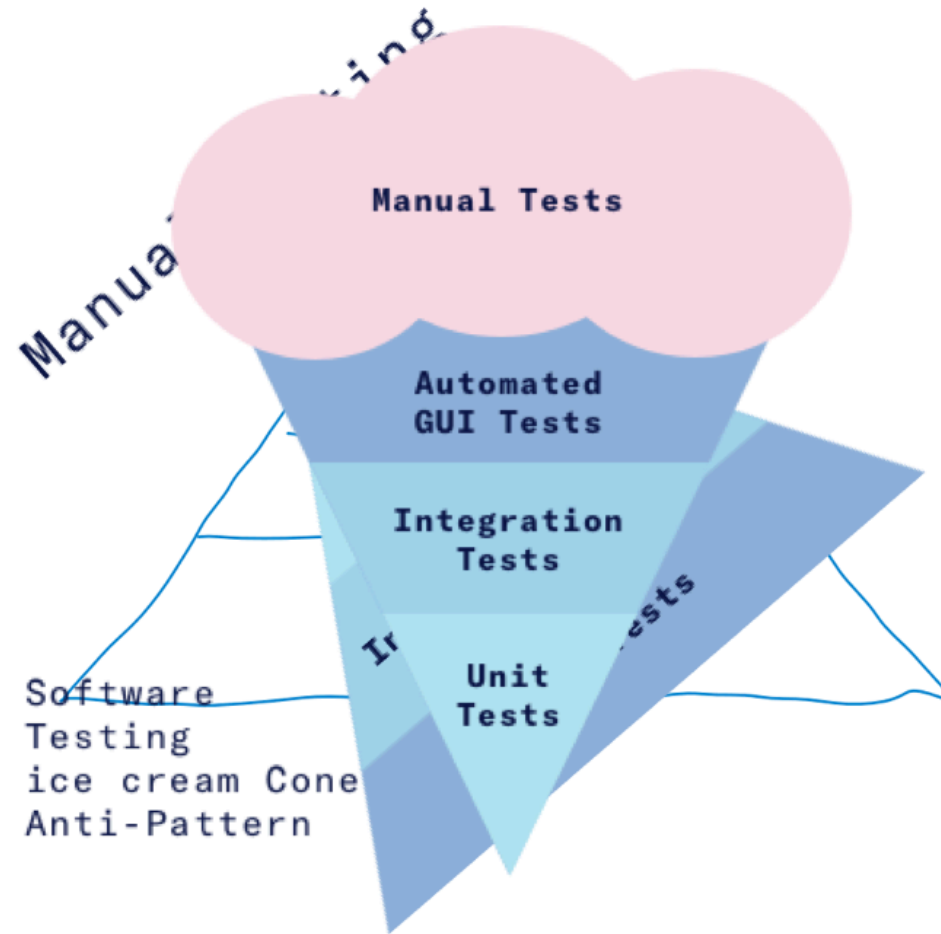


# What is the right testing level against production systems? #2

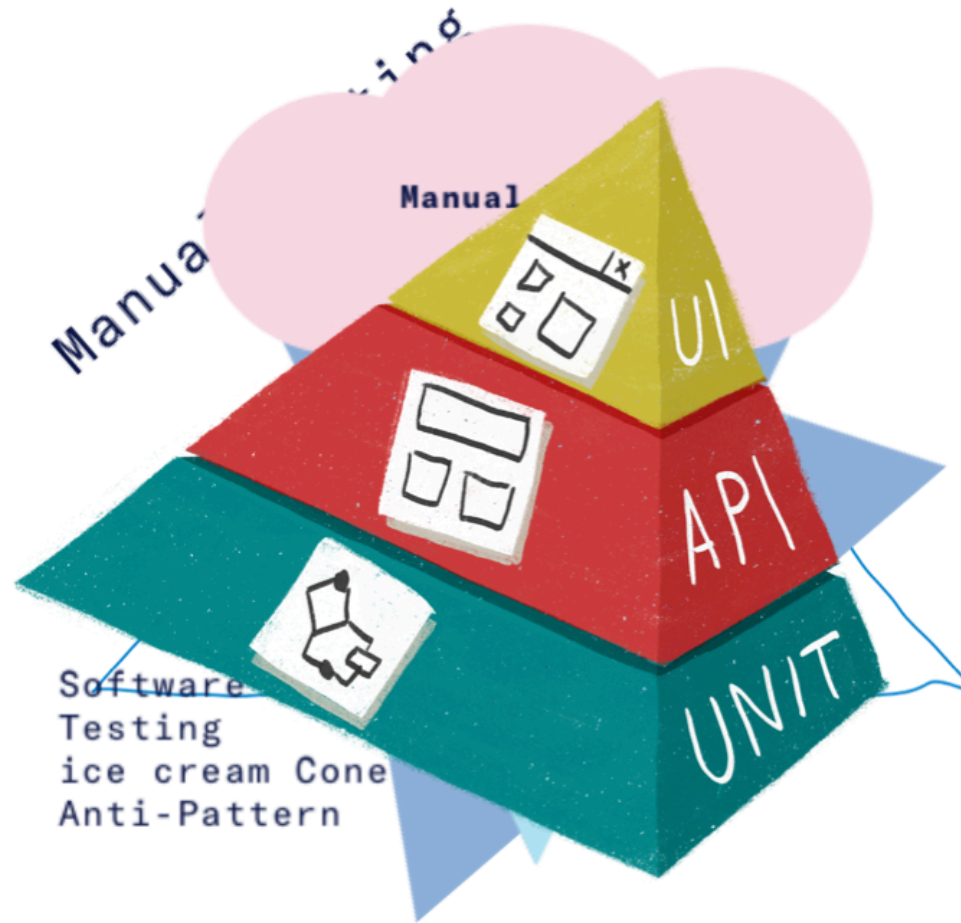


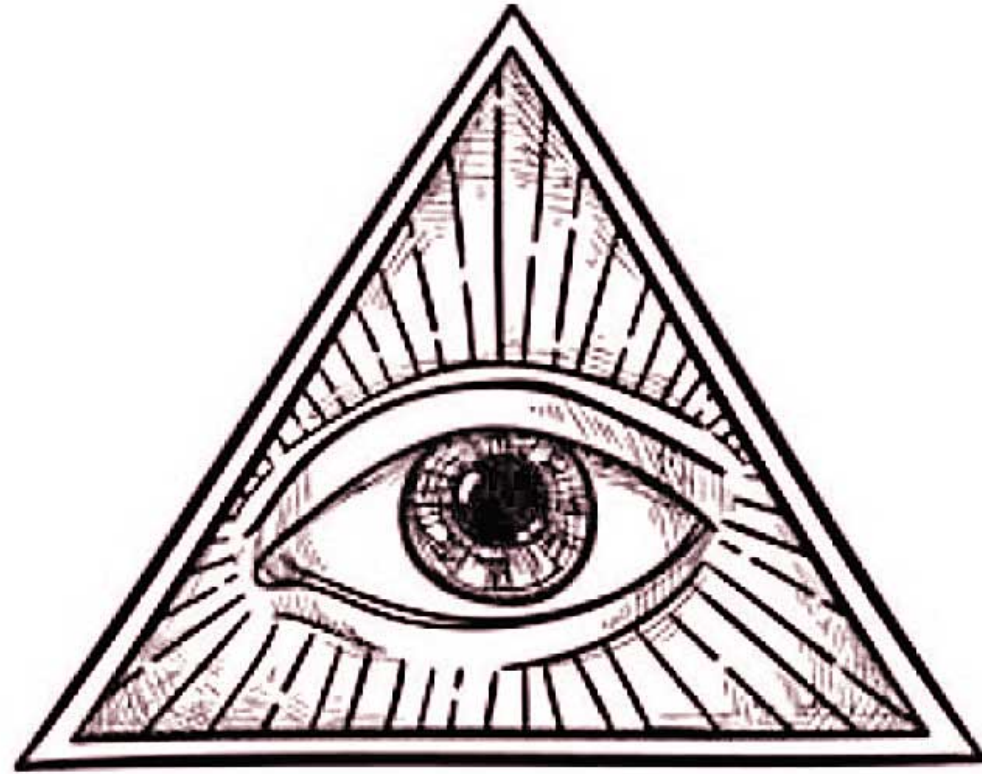


# What is the right testing level against production systems? #2

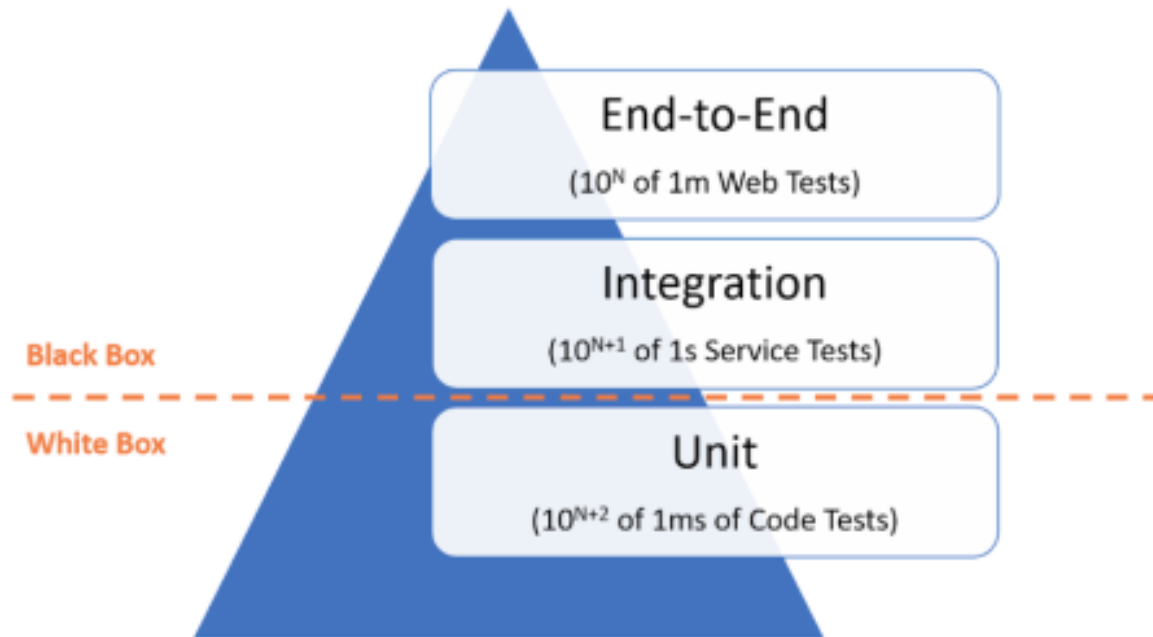


# What is the right testing level against production systems? #2

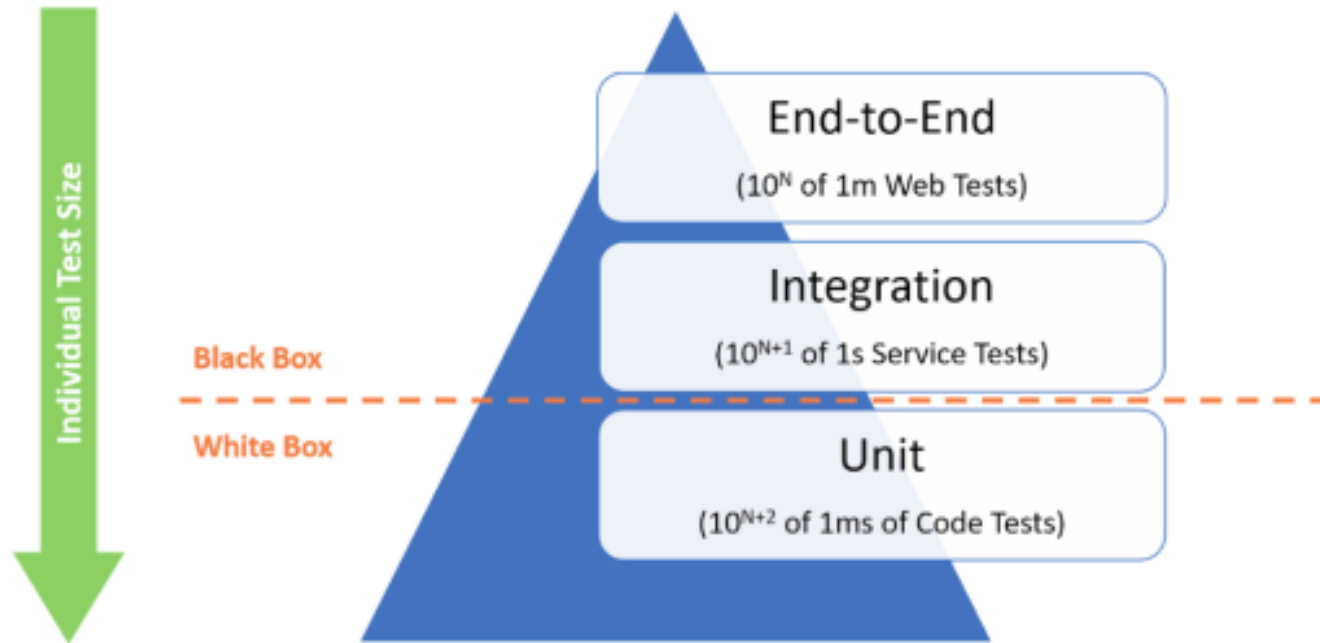




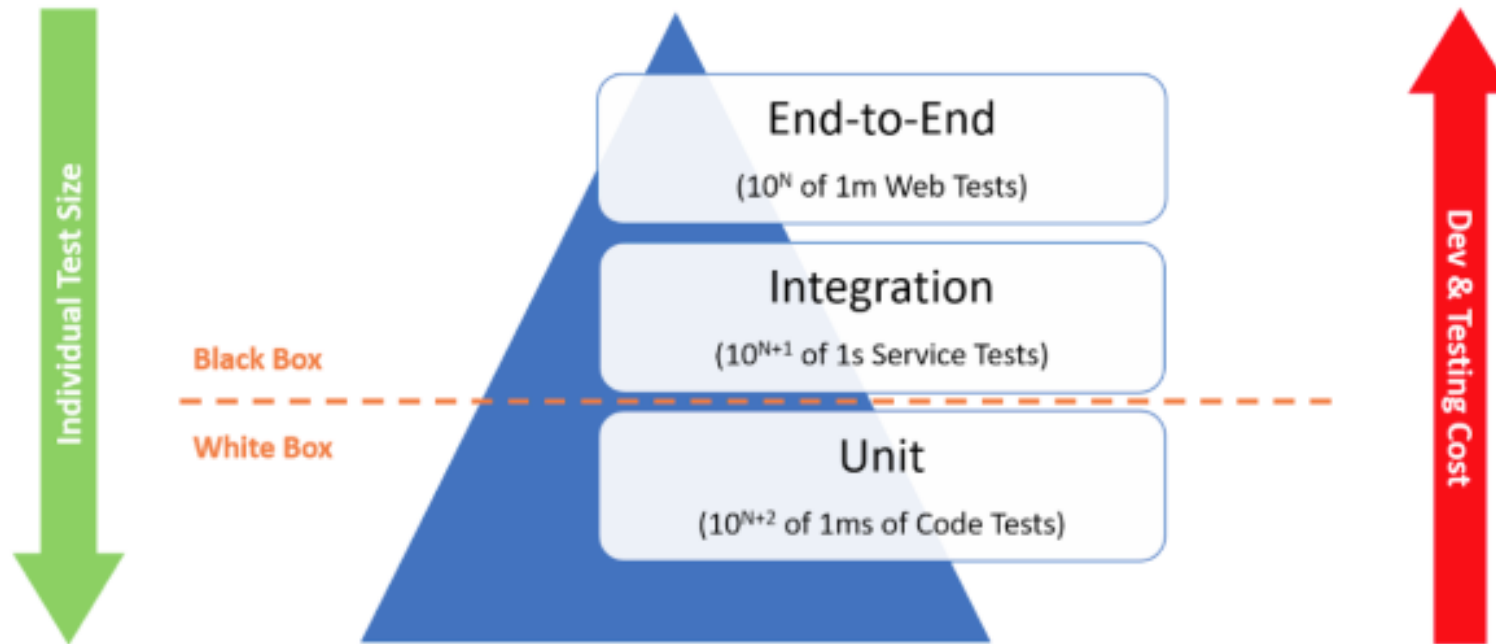
# What is the right testing level against production systems? #2



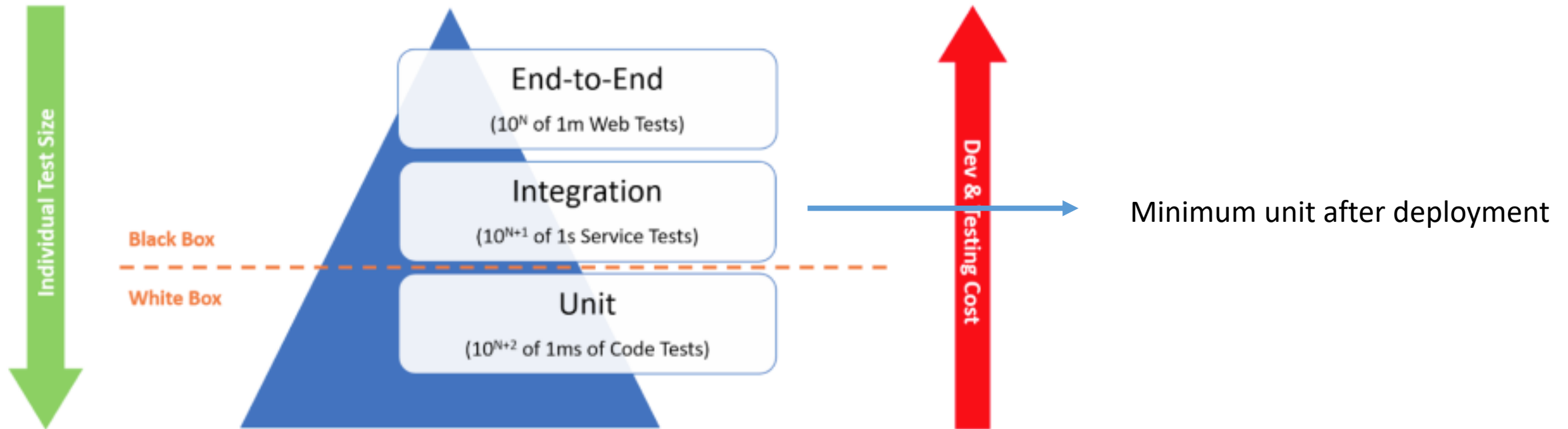
# What is the right testing level against production systems? #2



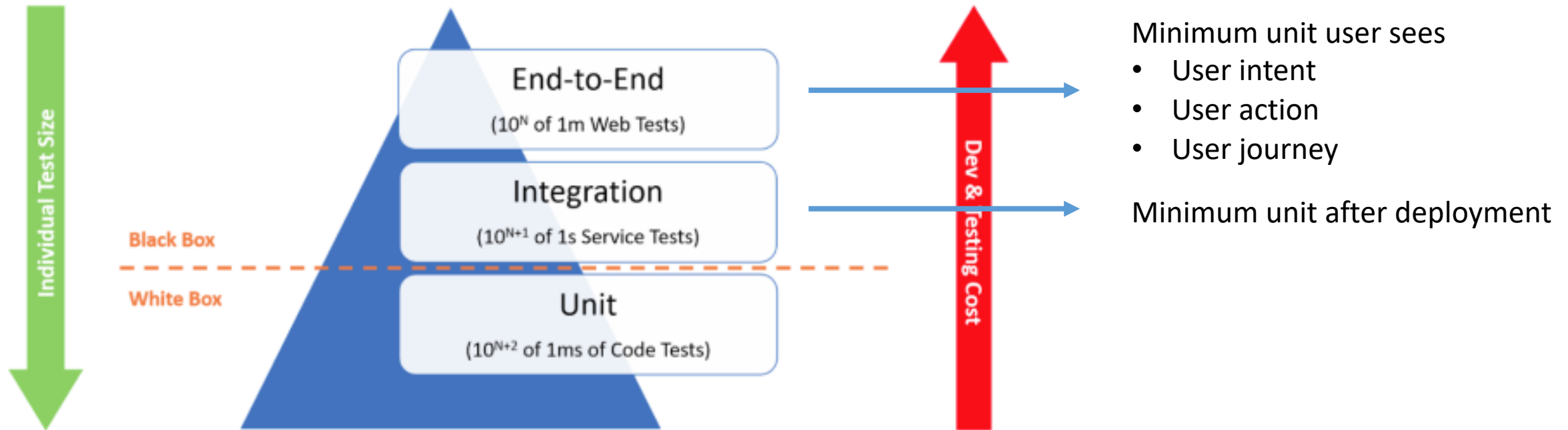
# What is the right testing level against production systems? #2



# What is the right testing level against production systems? #2



# What is the right testing level against production systems? #2





# Smoke Tests

Allows us to quickly **asses the status of an application** by running a set of end-to-end tests targeted at checking the most important user flows

## Example of end to end tests #3

# Example of end to end tests #3

- More than 2 million connected products
- Users all around the globe

The Dyson logo, consisting of the word "dyson" in a bold, lowercase, sans-serif typeface.

# Example of end to end tests #3

- More than 2 million connected products
- Users all around the globe
  - Link new product to my account
  - Rename cleaning robot
  - Start a scheduled clean
  - Return to base
  - Turn on the fan

The Dyson logo, consisting of the word "dyson" in a bold, lowercase, sans-serif typeface.

# How can we define an end to end test? #3

- Starting point
- User action
- User action
- Expected outcome

# How can we define an end to end test? #3

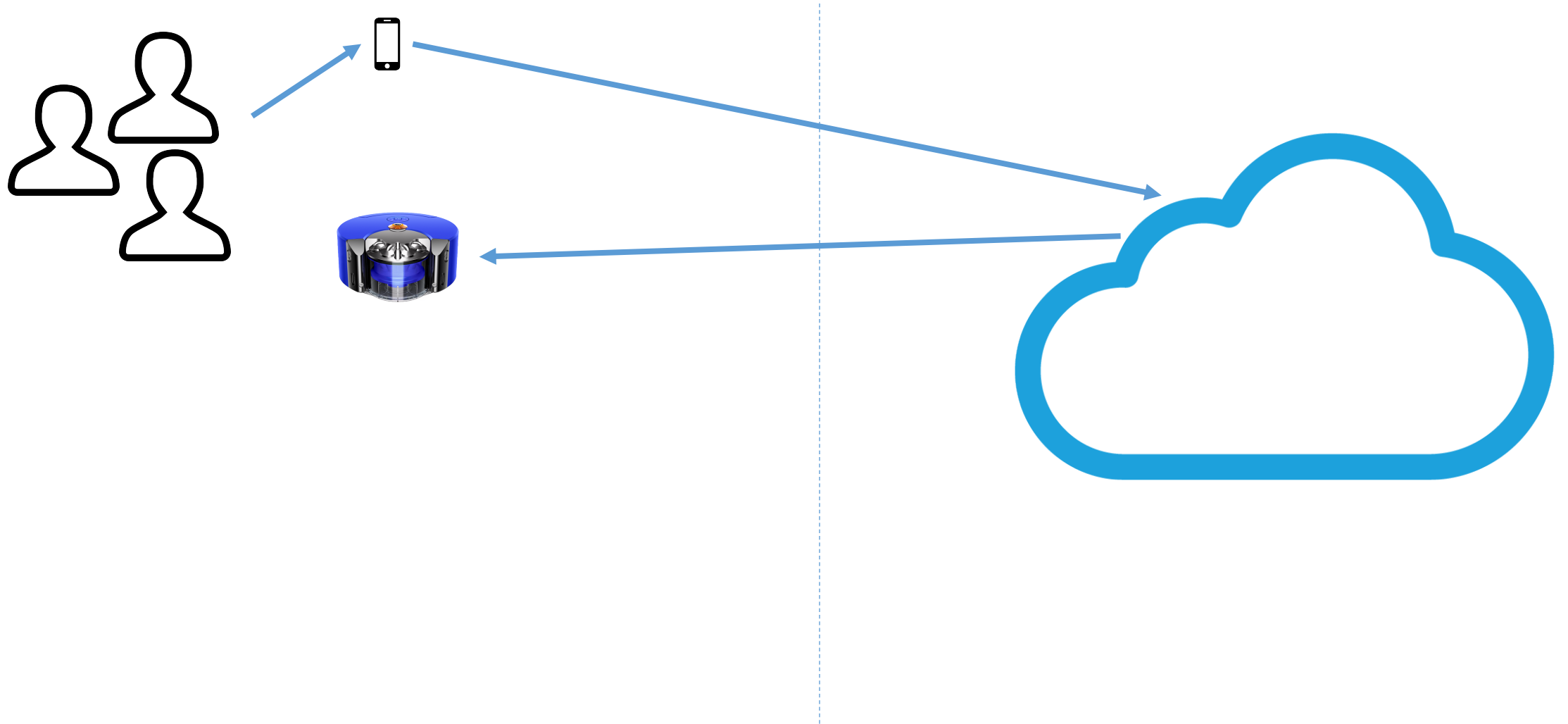
- Given an existing user with this configuration
- And the user logs in
- And the user changes robot name to 'RoboMop9000'
- Then the robot name has changed to 'RoboMop9000'

# How can we define an end to end test? #3

- Given an existing user with this configuration
- And the user logs in
- And the user changes robot name to 'RoboMop9000'
- Then the robot name has changed to 'RoboMop9000'

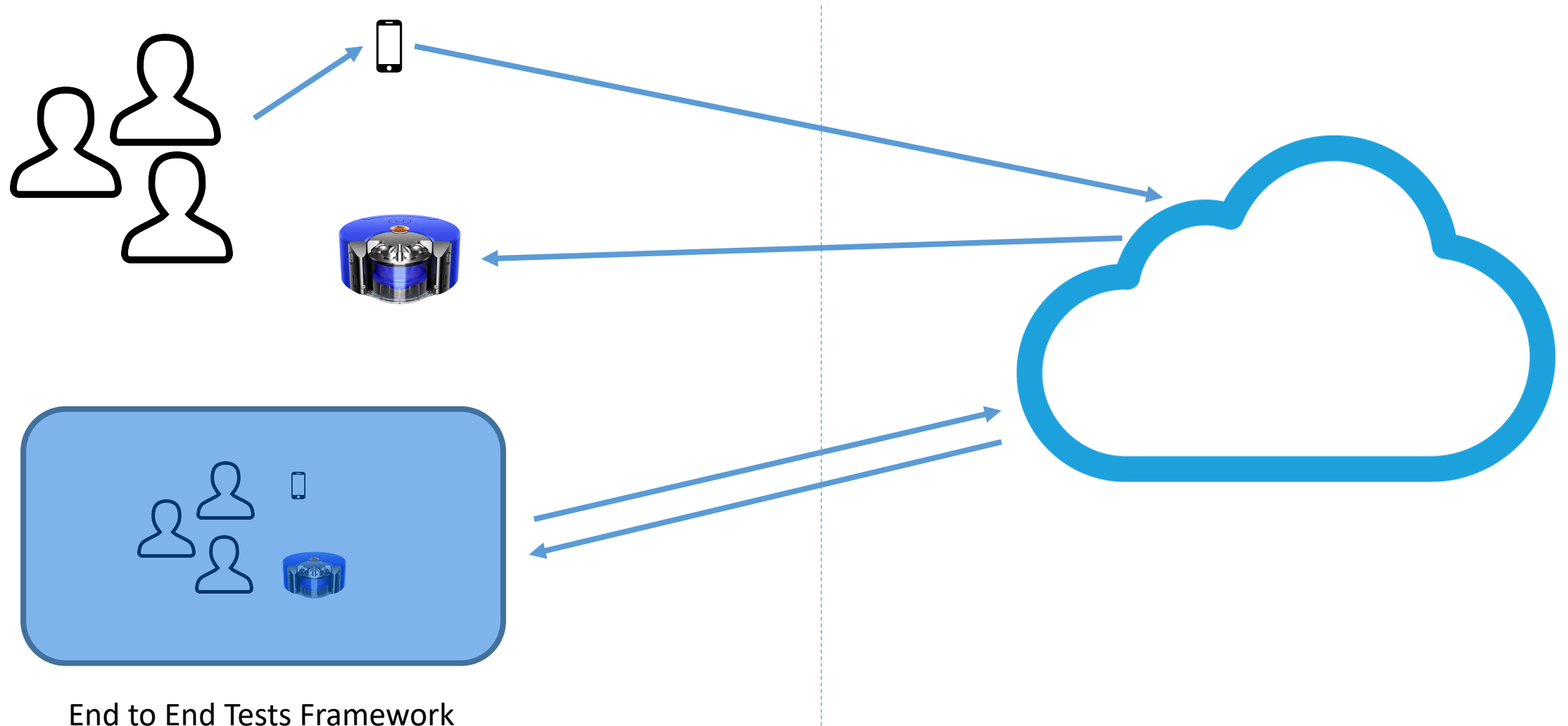


# How can we define an end to end test? #3





# How can we define an end to end test? #3



# How can we define an end to end test? #3

- Combination of:

- CucumberJS
- Request & Assert library



- Lots of single steps defining calls to APIs and data transformations
- Grouped together under user journeys or scenarios

# How can we define an end to end test? #3

- Given an existing user with this configuration
- And the user logs in
- And the user changes robot name to 'RoboMop9000'
- Then the robot name has changed to 'RoboMop9000'

# How can we define an end to end test? #3

- Given an existing user with this configuration
- And the user logs in
- And the user changes robot name to 'RoboMop9000'
- Then the robot name has changed to 'RoboMop9000'

- Given an existing user with this configuration
- And the user logs in
- And the user schedules a clean
- Then the robot starts a clean within 10 seconds

- Given an existing user with this configuration
- And the user logs in
- And the user stops the clean
- Then the robot returns to the base within 10 seconds

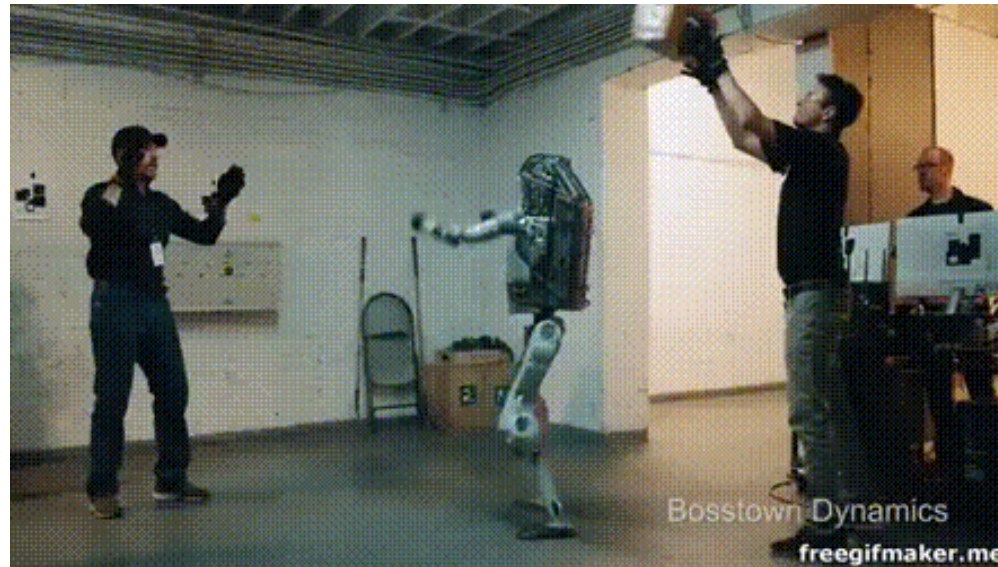
BUT...

# How avoid disrupting real users? #4 problem

- If your test is too close to real users and real actions...

# How avoid disrupting real users? #4 problem

- If your test is too close to real users and real actions...



# How avoid disrupting real users? #4 solution

- Spacing these end to end tests
  - The more often you run them the faster you will notice an issue
  - Noise
  - Slowing down your production systems
  - Balance

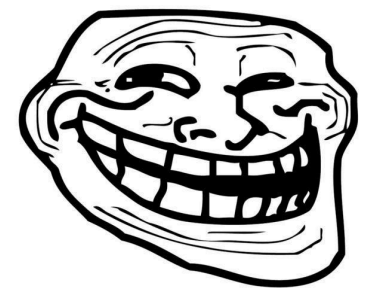


# How avoid disrupting real users? #4 solution

- Spacing these end to end tests
  - The more often you run them the faster you will notice an issue
  - Noise
  - Slowing down your production systems
  - Balance
- Virtual users registered in production like real users with their own configuration

How to keep your tests out of statistics? #5

# How to keep your tests out of statistics? #5

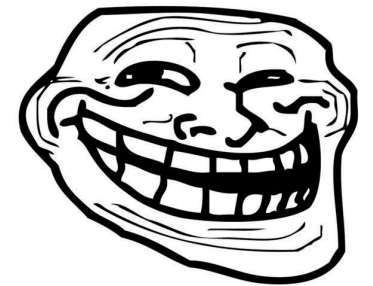


**problem?**

# How to keep your tests out of statistics? #5

problem

- They can also skew some of the metrics you record like:
  - Number of users interacting with your system
  - API usage
  - Error count
- Even mixed logs
  - How to discern what was part of a test or part of a real interaction?



**problem?**

# How to keep your tests out of statistics? #5

solution

- Correlation id

ffffffff-0000-0000-0000-000000000000

- User agent

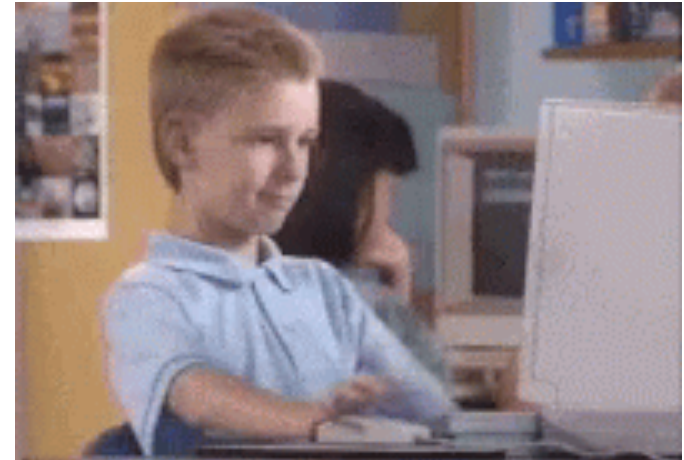
AppId/AppVersionId (Language=LanguageNameAndOptionallyVersion)

- Custom HTTP headers

TestToolVersion=1.1.2

# Cleanup? #6

- Running tests in production 👍



# Cleanup? #6

- But... what about the all the fake data they generated?



# Cleanup? #6

- Automatically clean as much testing items as possible after each test
- Objective: Leave the system as you found it





# Cleanup? #6

- Helps starting tests from a clean state
  - Avoiding false negatives
  - Each test is independent
- Idempotency
- Avoids cluttering your production databases
- Avoids exhausting other limited resources
- Avoids making real data more difficult to retrieve and search



# That's interesting

- Can a system failure or bug go unnoticed to your E2E tests?

- By their nature E2E Tests span multiple services and teams

Who is responsible for them and how do teams collaborate to write them?

- How do E2E Tests compare to dark or canary Deployments?

Do you need both?

- How do you avoid E2E Tests becoming fragile?



# Recap

- Testing in production is good and necessary 🎉
- Think like an user, act like an user 🧠
- CucumberJS is a good start
- Think about your system's capacity 📈
- Mark test intents to differentiate them from real user interactions
- Clear your test data and reset connections/status after each test

Use the comments section!

Follow me on Twitter!

Enjoy the rest of the conference!



and you?  
are you brave enough to test in production?

Questions round