

JavaScript Testing 2019

@ codecentric

@MarcoEmrich

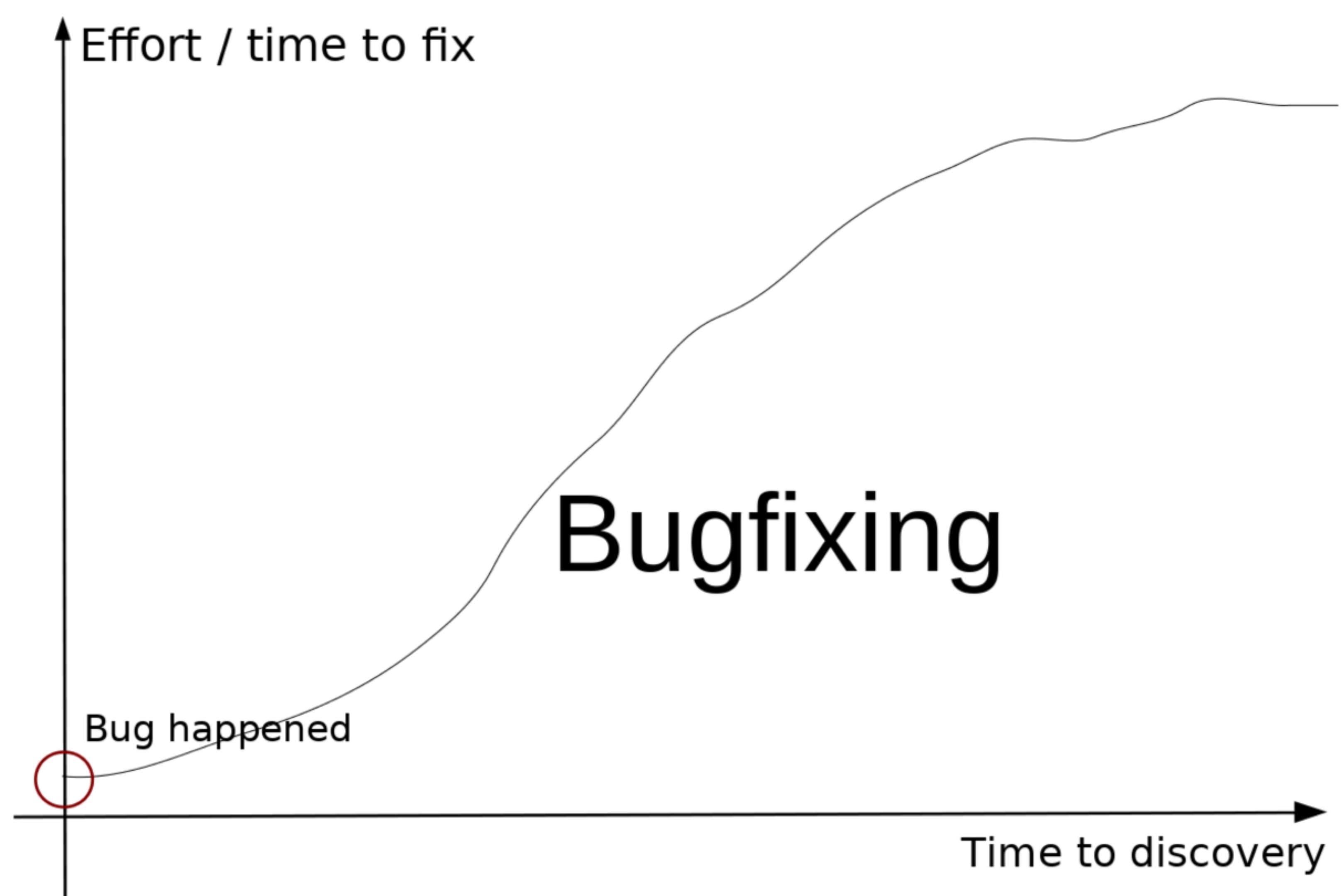
TOPIC VOTING

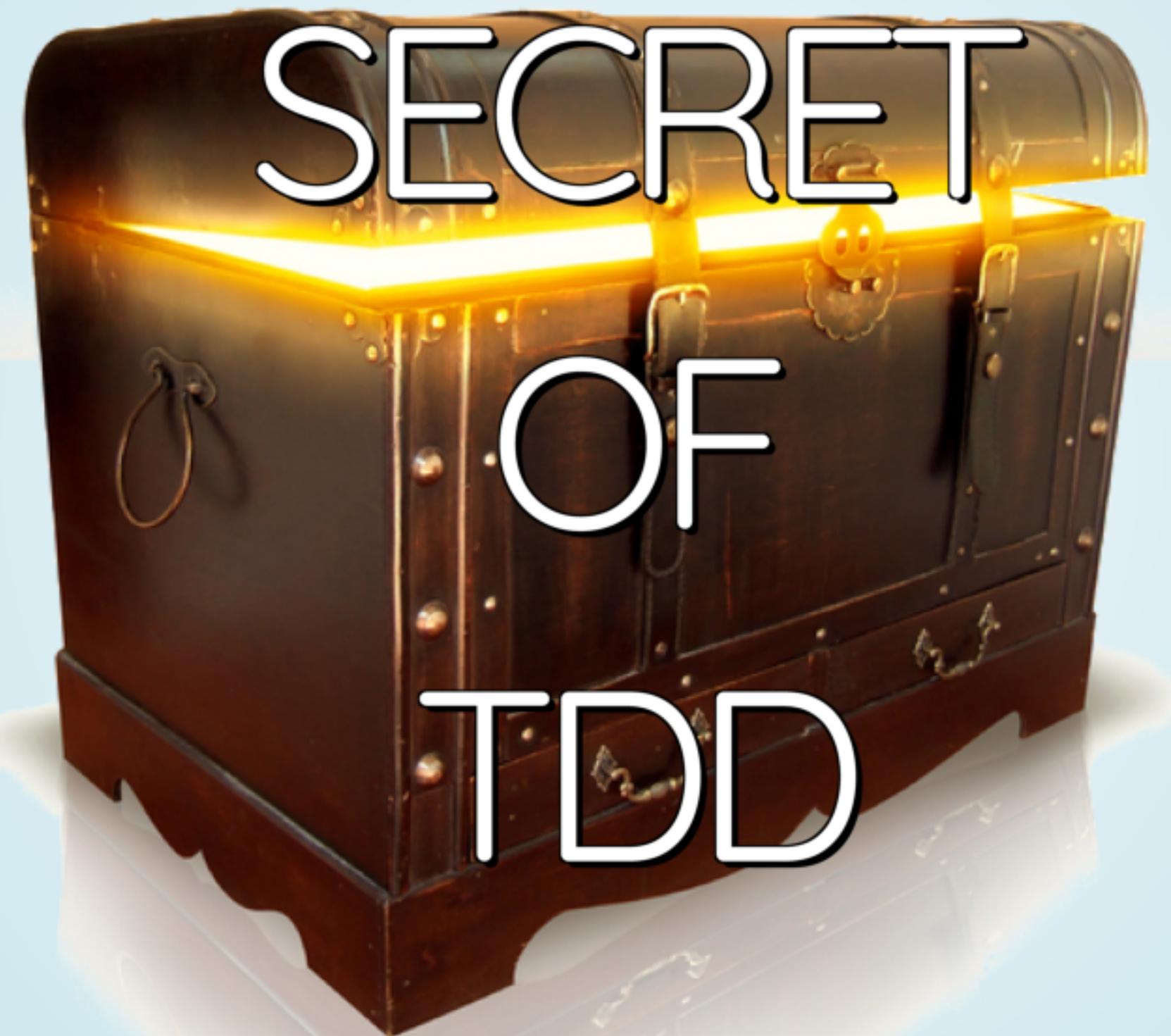
1. Basics / TDD: **Jest**
2. Assertions: **PowerAssert, Unexpected**
3. Parametrized Tests: **Jest**
4. Test Doubles: **Jest**
5. Mutation Based Testing: **Stryker**
6. Unit-Testing the GUI: **Enzyme**
7. End2End-Gui Testing: **Cypress**

What is the most
time consuming activity
in programming?

- A** - New features
- B** - Refactoring
- C** - Hunt for bugs

- A** - New features
- B** - Refactoring
- C** - Hunt for bugs





QA?

PRODUCTIVITY!



RESEARCH?

*The results [...] indicate that [...] the resulting quality was higher than teams that adopted a non-TDD approach by an order of at least **two times***

— *Evaluating the Efficacy of Test-Driven Development*

Microsoft Research (2006)

Test-first [...] did increase productivity probably because tests allow for better task understanding, better task focus, faster learning, and lower rework efforts

— *On the Effectiveness of Test-first Approach to Programming*

Carnegie Mellon University (2005)

WHEN?

WHO?

1957

INVENTED



JOHN VON NEUMAN



1989



REDISCOVERED

KENT BECK

30 YEARS OF TDD

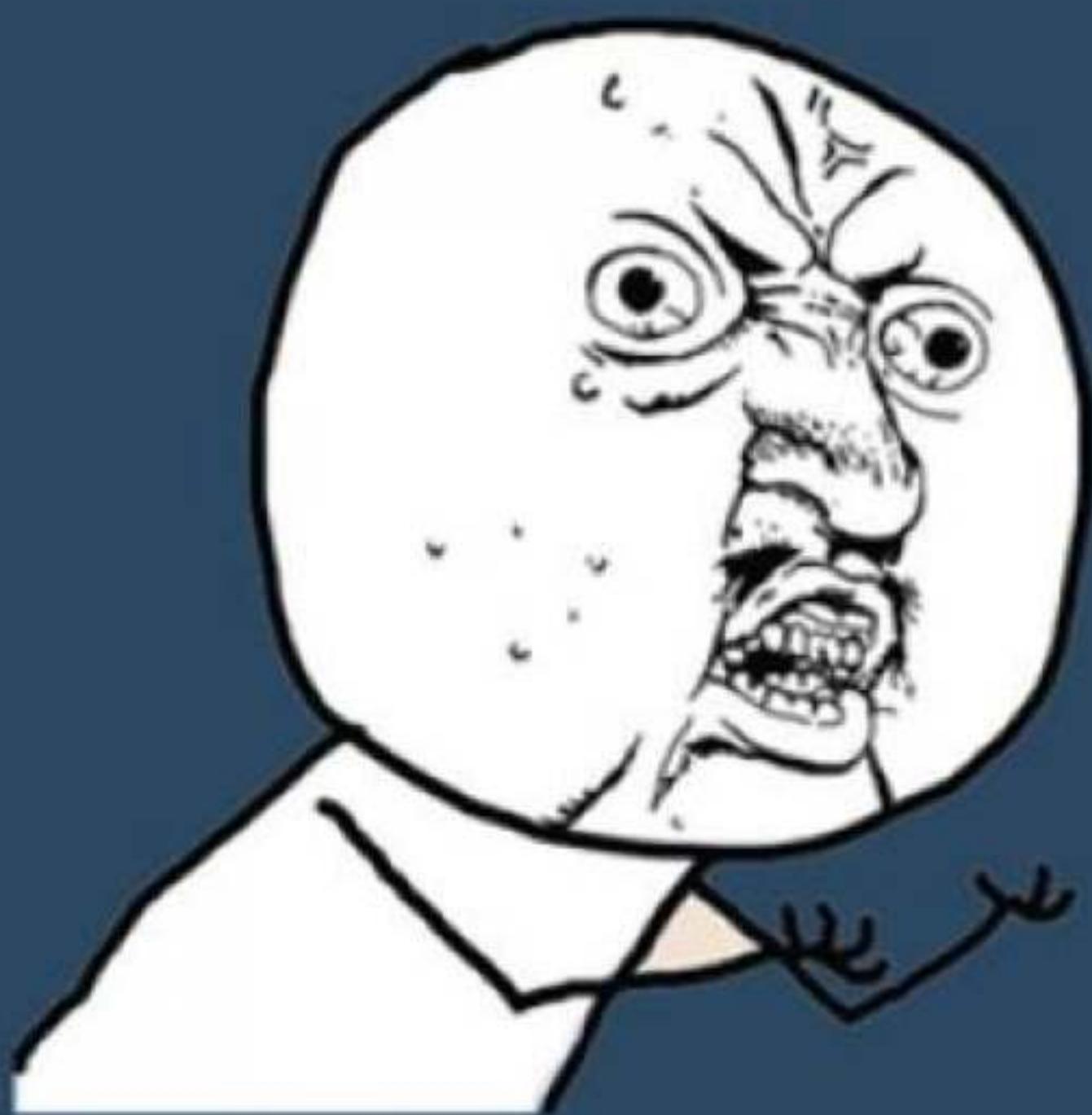
...OR 60?

FIRST JS TDD-
FRAMEWORK?

2001

JS UNIT

YU NO



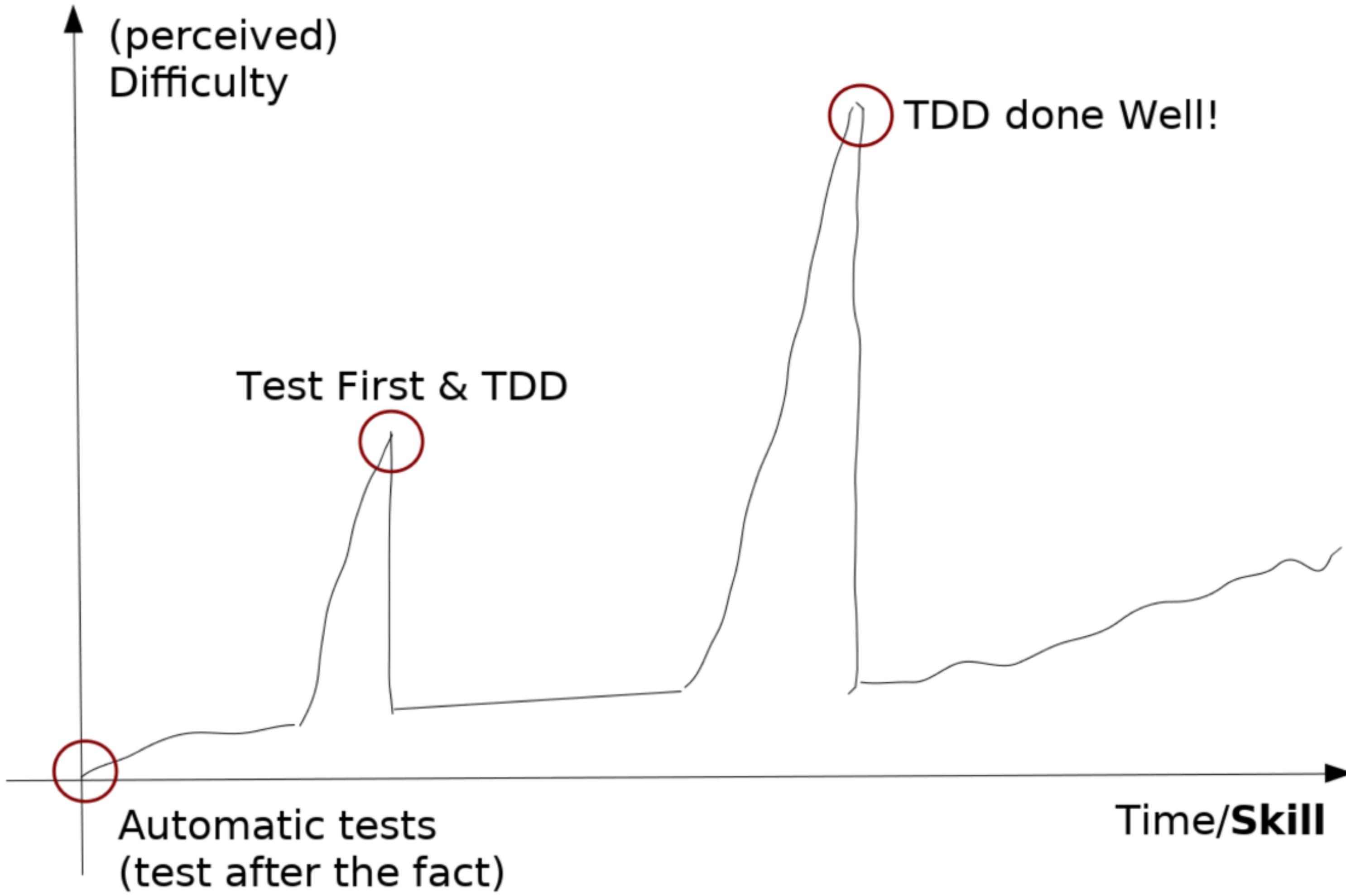
TDD













A photograph of a woman lying face down on a red running track. She is wearing a black leotard and has her hair pulled back in a bun. A high jump bar with a blue and white padded mat is positioned directly above her head. The background shows a white fence and trees under a clear blue sky.

SLOW
HARD TO MAINTAIN

BUT WHY?



TDD is not about Testing
— everywhere on the Internet

TDD === DESIGN

TESTS == SPECS

The background features a stack of several antique books with gold-tooled spines and covers. A small, dark glass bottle with a stopper sits atop the books. The lighting is dramatic, casting deep shadows and highlighting the texture of the book spines.

LIVING DOCUMENTATION

EXAMPLE

LEAP YEAR

```
isLeapYear(2000); // => true
```

A BAD TEST

```
testIsLeapYearIsCorrect()
```

from [Structure and Interpretation of Test Cases](#) by Kevlin Henney

A SPEC DOCUMENT

- A year is a leap year if it is divisible by 4 but not by 100
- A year is a leap year if it is divisible by 400
- A year is NOT a leap year if it is not divisible by 4
- A year is NOT a leap year if it is divisible by 100 but not by 400

from [Structure and Interpretation of Test Cases](#) by Kevlin Henney

```
test("A year is a leap year if it is divisible by 4 but not by 100", {  
    ...  
});  
  
test("A year is a leap year if it is divisible by 400", {  
    ...  
});  
  
test("A year is NOT a leap year if it is not divisible by 4", {  
    ...  
});  
  
test("A year is NOT a leap year if it is divisible by 100 but not  
    by 400", {  
    ...  
});
```

FLATTEN THE TDD
LEARNING CURVE?

A close-up portrait of a man with a beard and mustache, wearing a maroon flat cap and a striped shirt. He is smiling and looking slightly to his left. The background is a solid dark grey.

Dan
North



BDD BEHAVIOUR DRIVEN DEVELOPMENT

2006 <https://dannorth.net/introducing-bdd>

Dan

North

VOCABULARY

Specs describing behavior

QUESTIONS

- Where to start?
- What should I (not) test?
- How to name the test?
- How can tests help me to locate bugs?

Ideas from

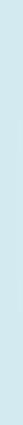
BDD



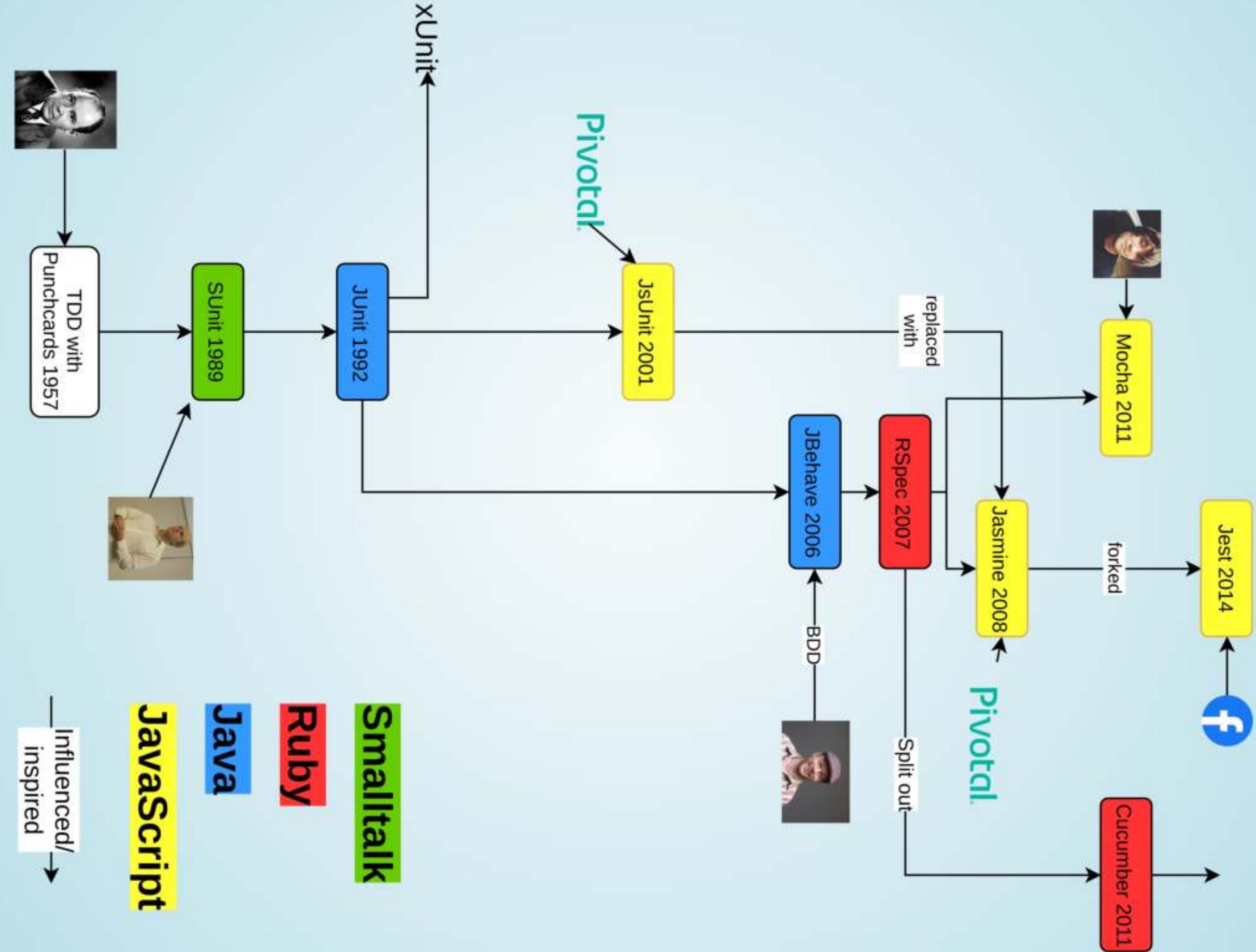
BDD is TDD done right

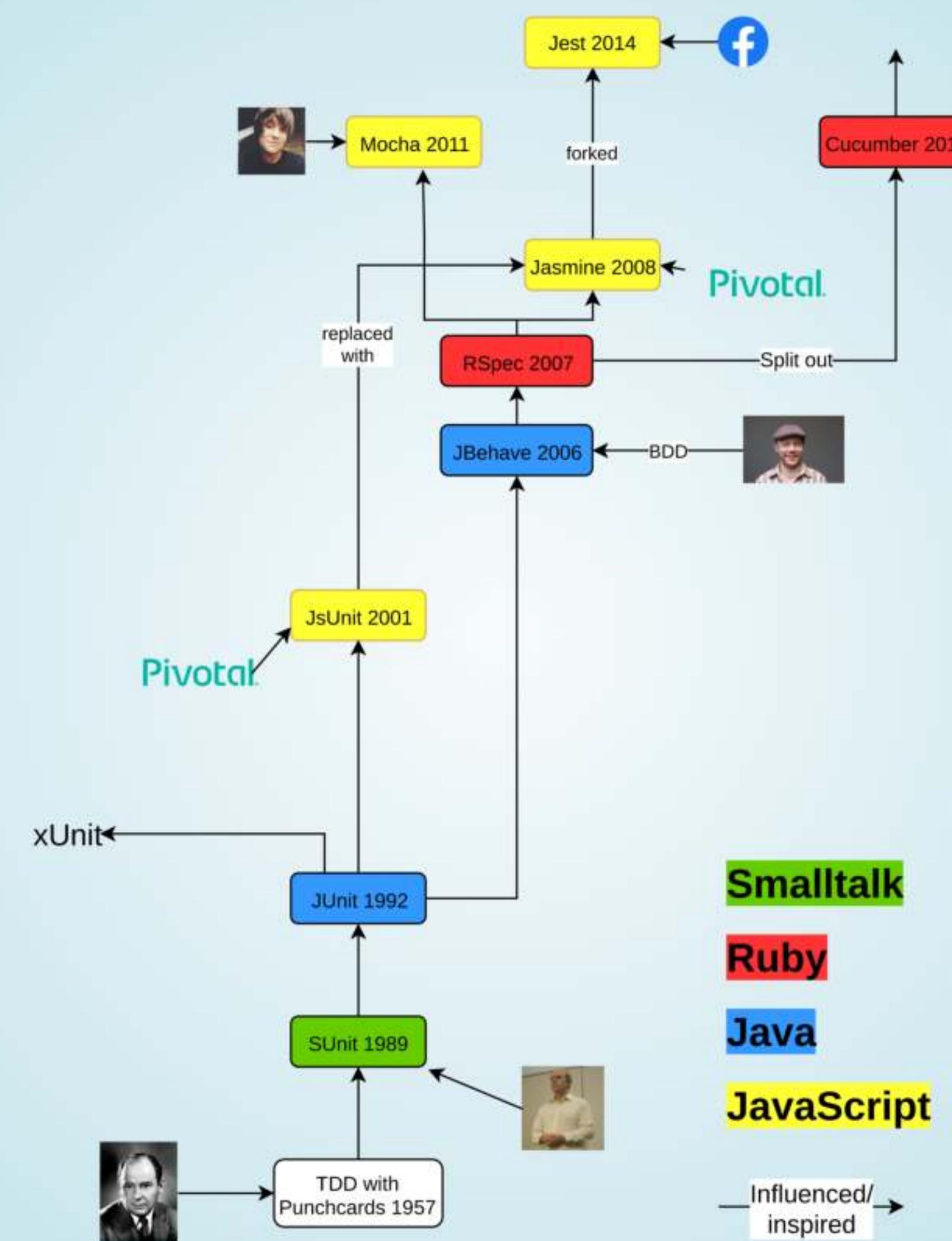
— J.B. Rainsberger

BDD-STYLE TDD
TDD DONE RIGHT

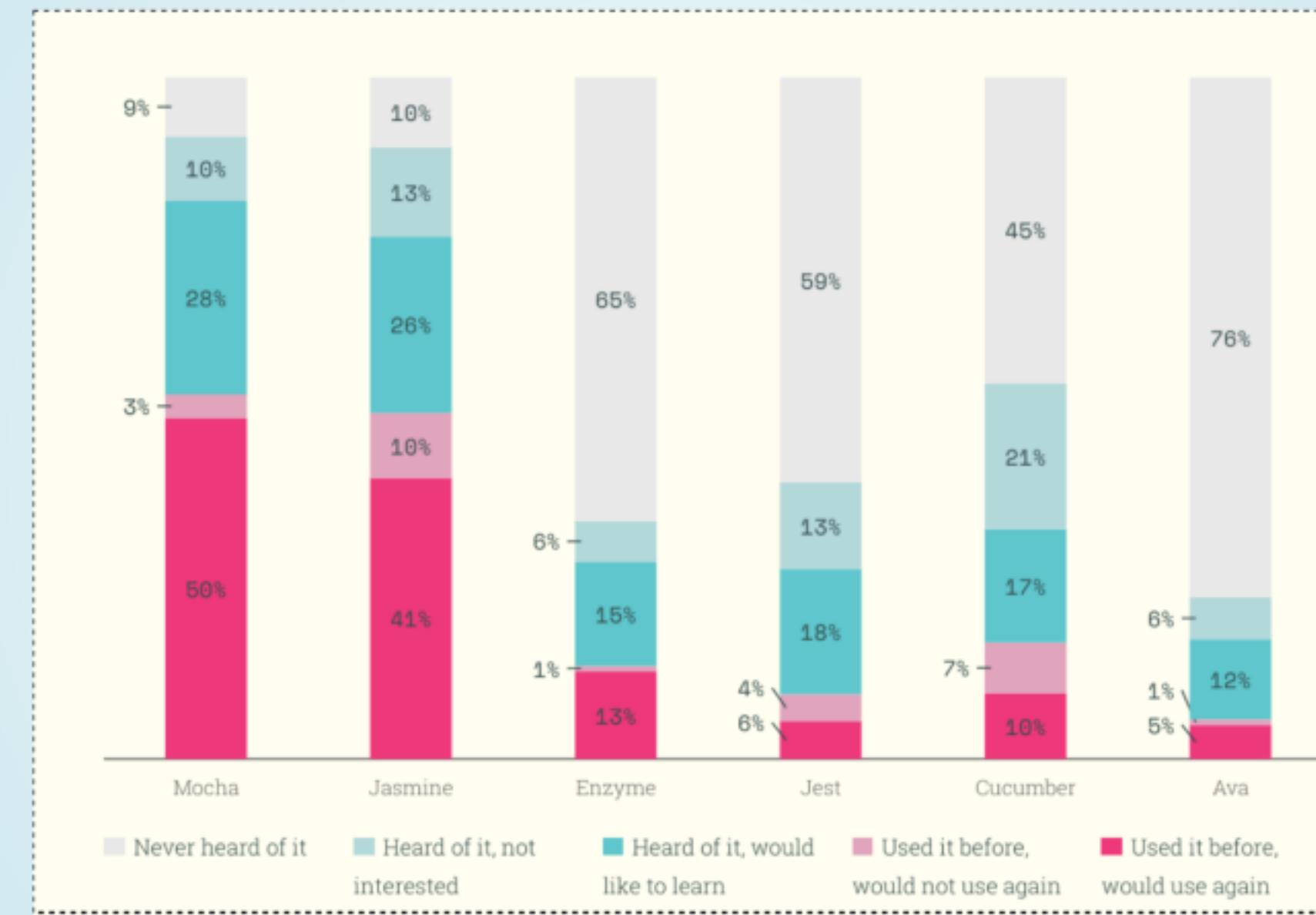


*modern JavaScript-Unit-Test-Frameworks
are BDD-Frameworks*



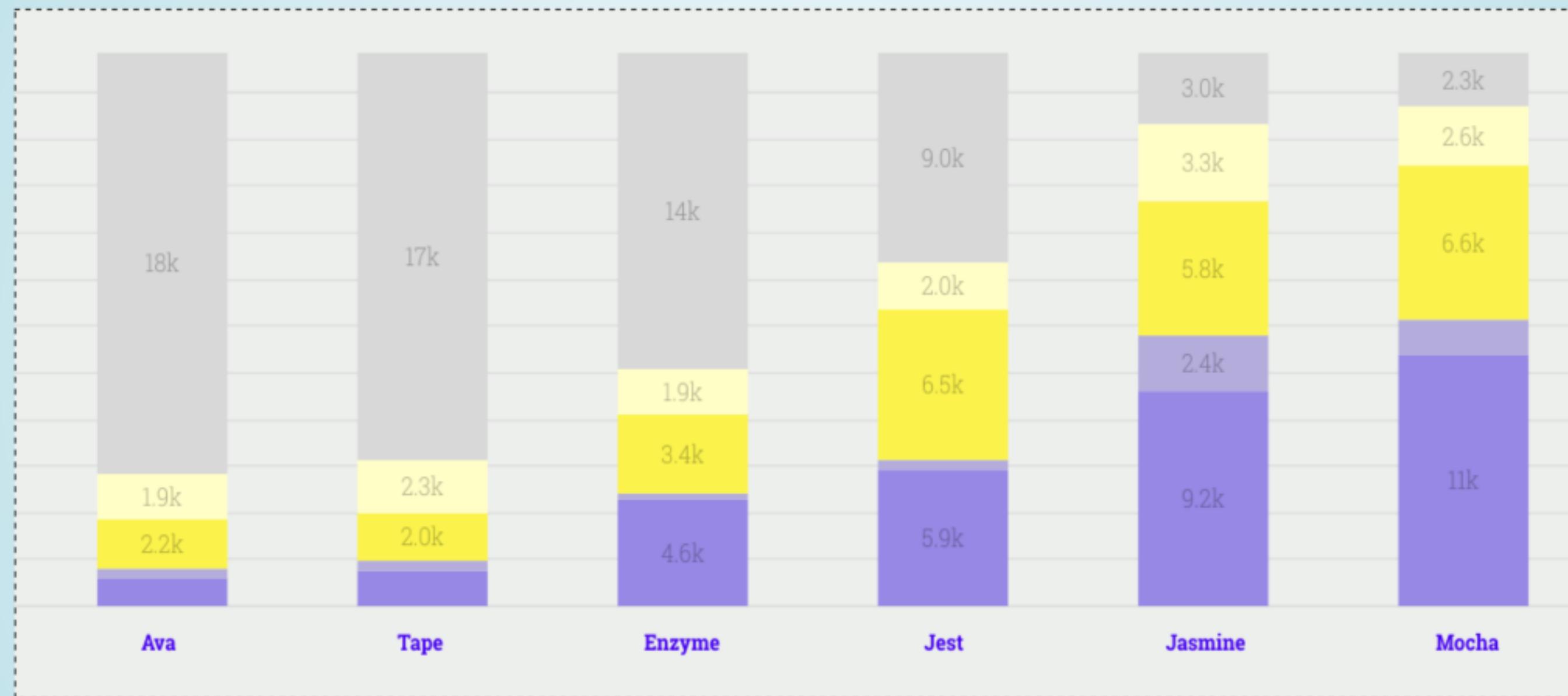


STATE OF JS 2016



Jasmine: 41% - Mocha: 50% - Jest: 6%

STATE OF JS 2017



Jasmine: 41% - Mocha: 49% - Jest: 25%

STATE OF JS 2018



Jasmine: 29% - Mocha: 40% - Jest: 40%

JS UNIT-TESTING 2019

- Mocha & Jest
- (Jasmine, Ava, Tape, ...)
- BDD family: describelt
- Autowatcher

MOCHA VS JEST
LIGHTWEIGHT VS BATTERIES-INCLUDED



JEST

STRING CALCULATOR

KATA

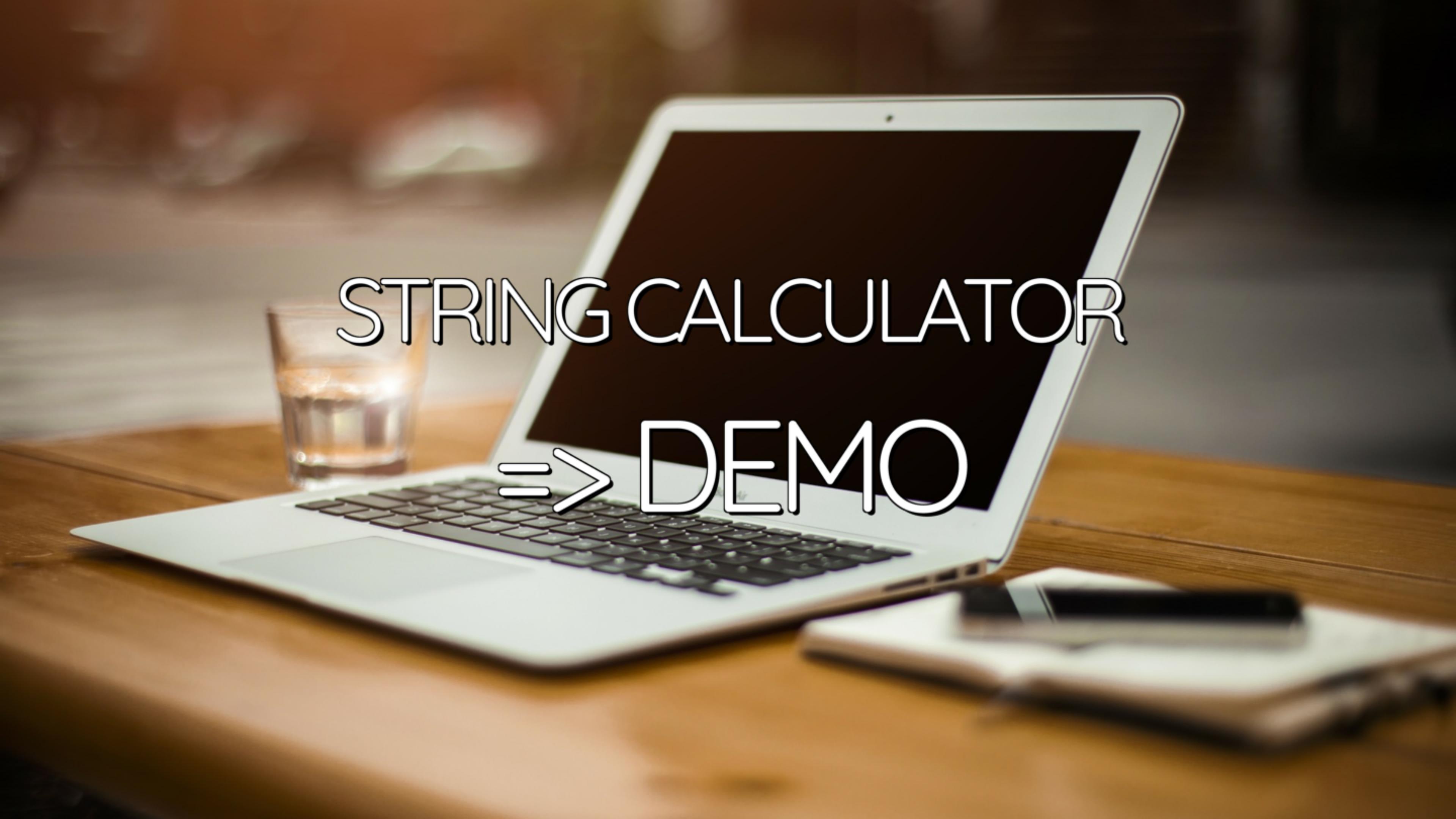


Roy Osherove

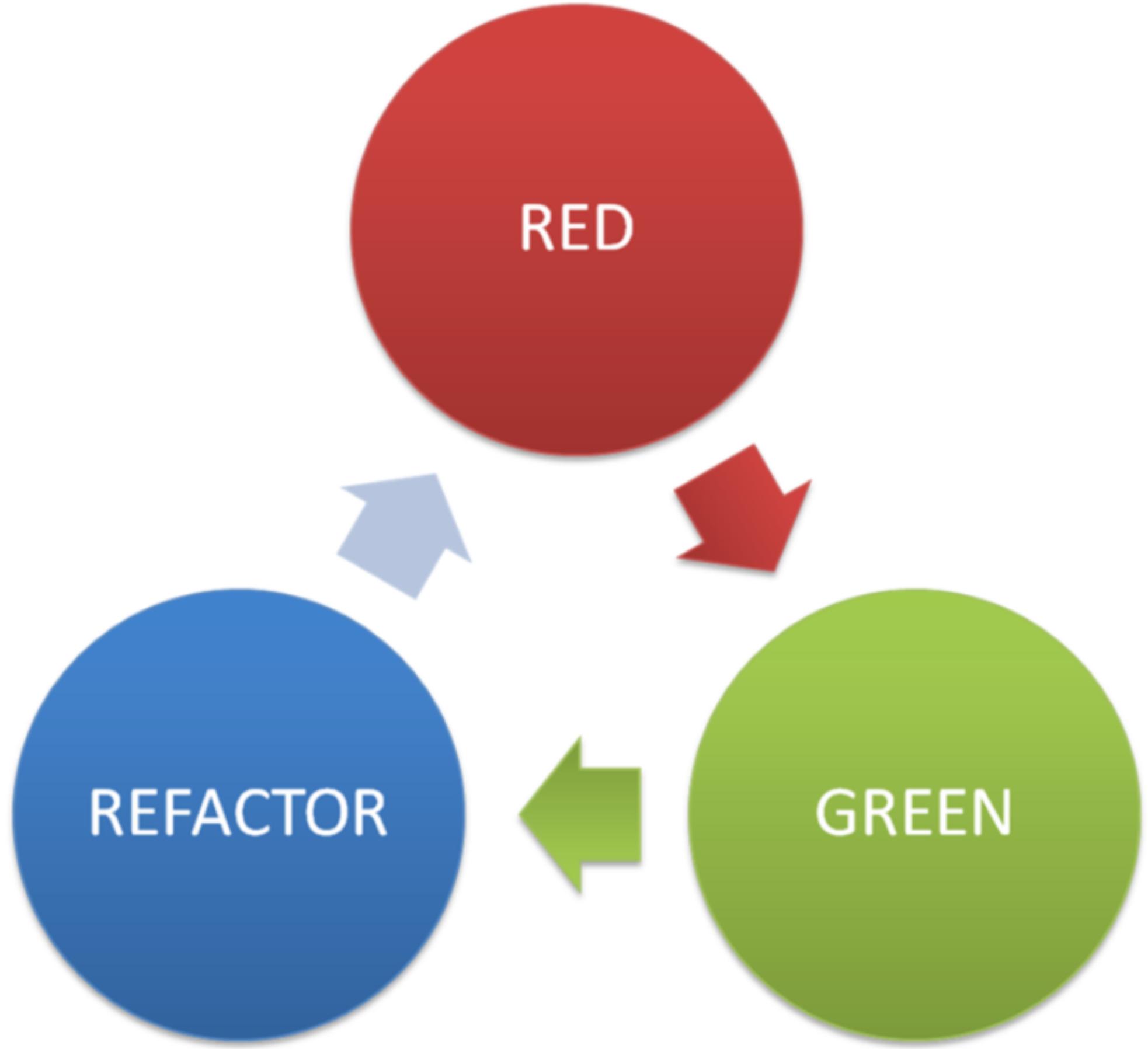
STRING CALCULATOR

KATA

"1,2,3" => 6

A white laptop is open on a light-colored wooden desk. On the screen, the words "STRING CALCULATOR" are written in a large, bold, white sans-serif font. Below them, a large white arrow points to the right, followed by the word "DEMO" in a large, bold, white sans-serif font.

STRING CALCULATOR
=> DEMO



ARRANGE

ACT

ASSERT

SETUP

EXECUTE

EXPECT

GIVEN

WHEN

THEN

The background features a stack of several antique books with gold-tooled spines and covers. A small, dark glass bottle with a stopper sits atop the books. The lighting is dramatic, casting deep shadows and highlighting the texture of the book spines.

LIVING DOCUMENTATION



**EXPLORE THE
SPACES**

"AYearIsALeapYearIfItIsDivisibleBy4ButNotBy100"

"A year is a leap year if it is divisible by 4 but not by 100"

FOCUS

Ai-A

Di

8

A

ASA 125

2.8 4 5.6 8

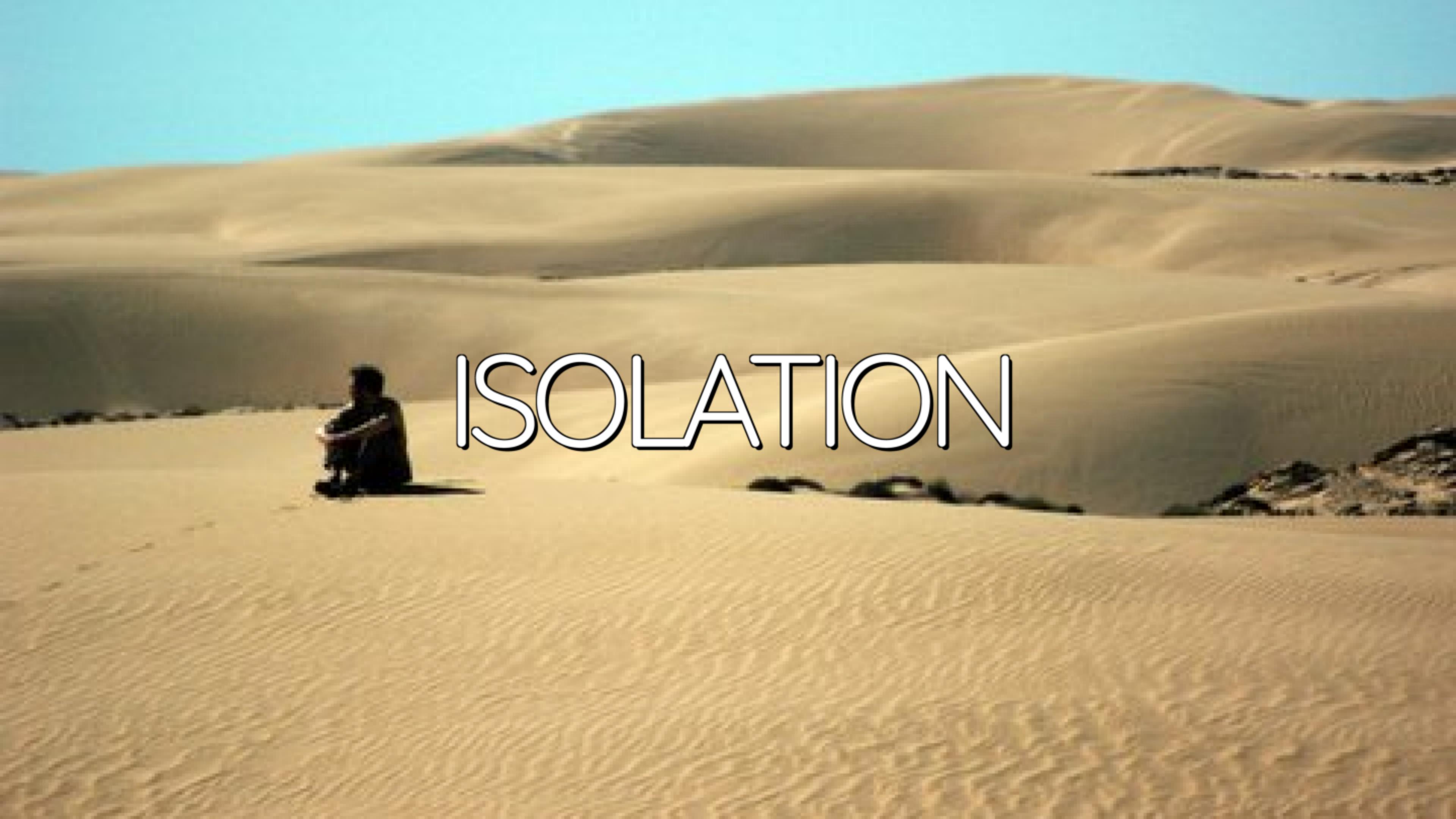
11 16 22

for Flash

SE



SUT
SUBJECT UNDER TEST

A photograph of a vast, rolling landscape, likely a field or prairie, under a clear blue sky. In the lower-left foreground, a person is sitting cross-legged on the ground, their back to the viewer, appearing small against the vastness of the scene.

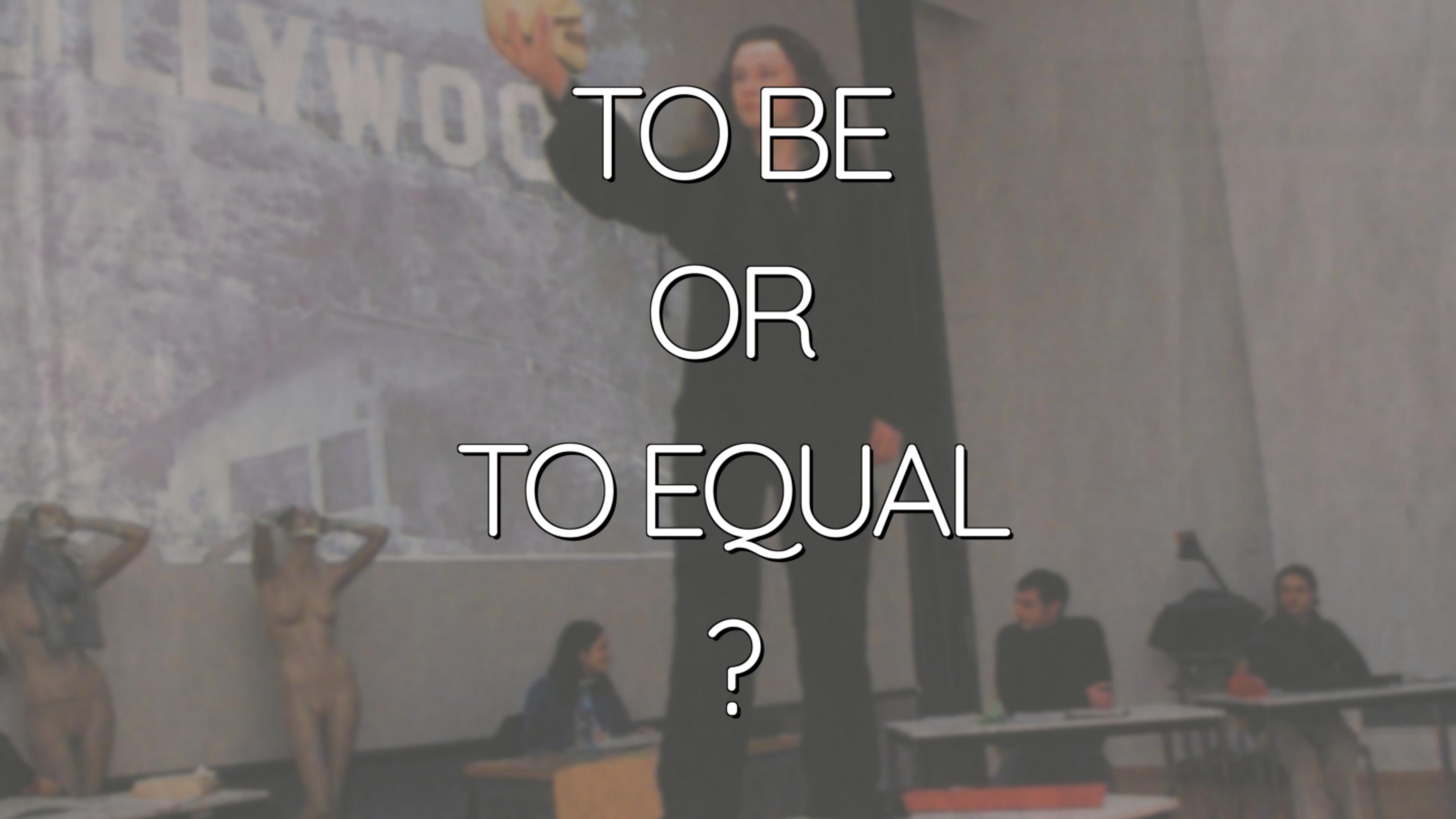
ISOLATION



BABY STEPS



EXERCISE
FIZZBUZZ

A classroom setting with several students seated at their desks, looking towards the front. A teacher stands near the chalkboard on the left side of the frame. The background is slightly blurred.

TO BE
OR
TO EQUAL
?

ASSERTIONS /
EXPECTATIONS

```
const person1 = {  
  name: "Schmidt",  
  vorname: ["Stefanie", "Adelheit"]  
}  
const person2 = {  
  name: "Schmidt",  
  vorname: ["Stefanie", "Adelheit"]  
}  
  
expect( person1 ).toEqual( person2 ); // => true  
expect( person1 ).toBe( person2 );    // => false
```

```
.toBe(value) / .toEqual(value) / .toStrictEqual(value)
.toBeCloseTo(number, numDigits?)
.toBeDefined() / .toBeUndefined()
.toBeInstanceOf(Class)
.toBeNull() / .toBeNaN()
.toBeFalsy() / .toBeTruthy()
.toBeGreaterThan(number) / .toBeLessThan(number)

.toContain(item)
.toHaveLength(number)
.toHaveProperty(keyPath, value?)

.toMatch(regexpOrString)
.toMatchObject(object)
.toThrow(error?)
```

```
.not  
.resolves  
.rejects
```

ASSERTION FRAMEWORKS

- Chai, Must, Should,...
- Jest, jest-extended
- [Unexpected](#)
- [PowerAssert](#)
- [HamJest](#)

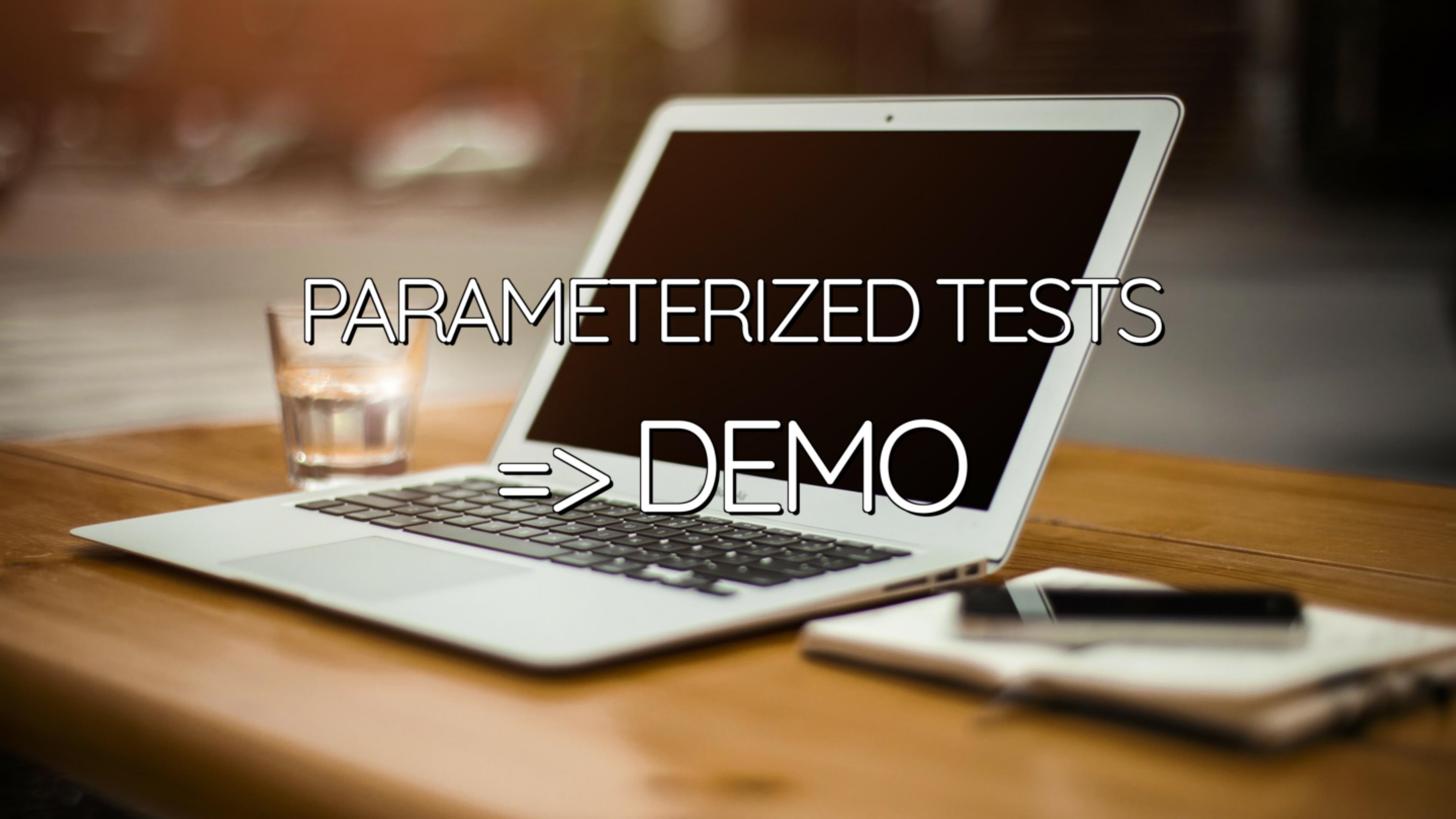
A photograph of a white laptop resting on a light-colored wooden desk. To the left of the laptop is a clear glass containing a light-colored liquid, possibly water or juice. To the right of the laptop, partially visible, is a white smartphone. The background is slightly blurred, showing a warm-toned wall.

ASSERTION FRAMEWORKS

=> DEMO

PARAMETERIZED TESTS

```
test.each(  
  [  
    [1, 1, 2],  
    [1, 2, 3],  
    [2, 1, 3]  
  ]  
)(  
  'add(%i, %i) => %i',  
  (a, b, expected) => {  
    expect(a + b).toBe(expected);  
  },  
);
```

A white laptop is open on a light-colored wooden desk. To its left is a small glass containing a clear liquid with ice cubes. In the background, a stack of books is visible. The laptop screen displays the text "PARAMETERIZED TESTS" in large, bold, white capital letters.

PARAMETERIZED TESTS

=> DEMO



TESTDOUBLES

www.denthighplastics.co.uk
Denthigh Building Practice
Denthigh Mobile Party Barge
SUPPLIERS & INSTALLERS

TEST DOUBLES

- Dummies
- Stubs
- Fakes
- Mocks & Spies

TEST DOUBLES IN JEST

- Same Interface: Spy
- jest.fn
- spyOn
- mockReturnValue
- (native JS)

STUB

```
class Person {  
    read() /* some reading from API */  
  
    fullName() {  
        const personData = this.read();  
        return personData.firstName + " " + personData.name;  
    }  
}  
  
it('should return full name of Person from API', () => {  
    const person = new Person();  
    const spy = jest.spyOn(person, "read");  
    spy.mockReturnValue({name: "Wonder", firstName: "Alice"});  
    expect(person.fullName()).toEqual("Alice Wonder");  
});
```

MOCK/SPY

```
const greet = (print, name) => print("Hello " + name);

describe('Mock Example: greet', () => {
  it('should print a greeting with Name', () => {
    let print = jest.fn()
    greet(print, "Alice");
    expect(print).toBeCalledWith("Hello Alice");
  });
});

greet(console.log.bind(console), "Steve");
```

The background image is a wide-angle aerial photograph of the London skyline during sunset. The sky is filled with warm orange and yellow hues. In the foreground, the River Thames flows from the bottom left towards the center, with the Tower Bridge spanning across it. To the right, the dense urban landscape of the City of London is visible, with numerous skyscrapers and office buildings. In the far distance, the Canary Wharf business district is prominent. The overall atmosphere is vibrant and captures the essence of a bustling city at dusk.

...LONDON SCHOOL OF TDD



EXERCISE
MARS ROVER

TEST-COVERAGE

LINE-COVERAGE
CO-COVERAGE

```
<?php
declare(strict_types=1);

class coverageIncreaser
{
    public function increaseOverallCoverage()
    {
        $a = 1;
        $a++;
        $a++;
        // ...
        // add some lines here if your
        // coverage dropped below the allowed level
        // ...
        $a++;
        $a++;
        return $a;
    }
}
```

ASSERTS?

MUTATION BASED TESTING

Stryker

Enhance your tests with mutation testing.



Powered by
InfoSupport



EXERCISE MOB OF MUTANTS

UI-UNIT-TESTING

JSDOM

VS

BROWSER DOM

UI-UNIT-TESTING TOOLS

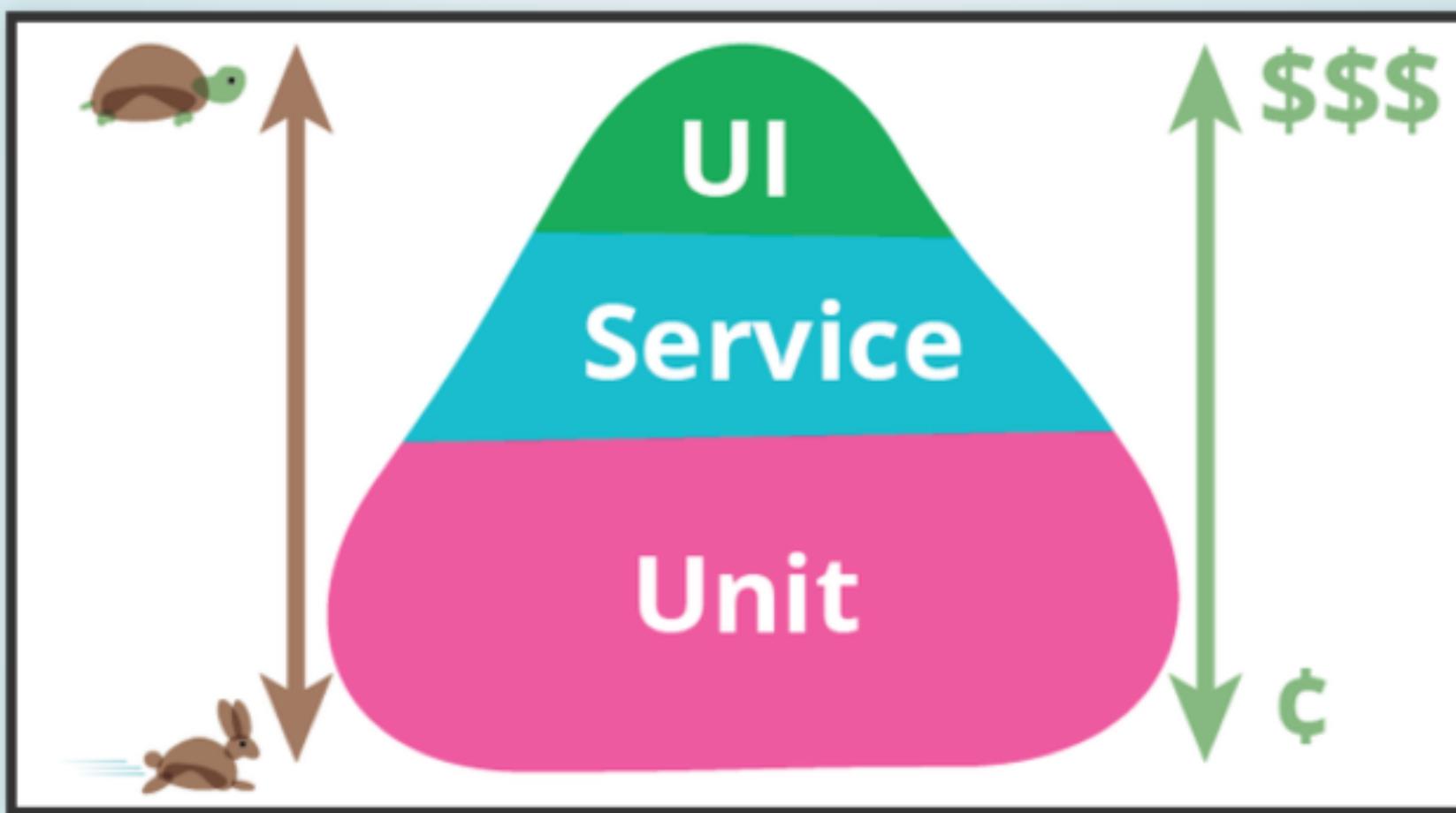
- Karma
- unexpected-dom / unexpected-react
- Enzyme
- Testing-Library.com



EXERCISE
TICTAC TESTING

END TO END

TEST-PYRAMID



by Martin Fowler

<https://martinfowler.com/bliki/TestPyramid.html>

END TO END

- Hard to Maintaincdcd
- Smoke-Tests
- 2nd Line Of Test Defence
- NO MORE Selenium

MODERN E2E-UI-TESTING

- Puppeteer
- Cypress
- Test-Coffee

A white laptop is open on a light-colored wooden desk. On the screen, the word "CYPRESS" is written in large, bold, white capital letters. Below it, the word "DEMO" is also in large, bold, white capital letters, preceded by a white arrow pointing to the right. To the left of the laptop, there's a small, clear glass containing a yellowish liquid, possibly whisky. In the background, a stack of books is visible.

CYPRESS
=> DEMO



EXERCISE

TICTAC TESTING: CYPRESS

RECAP

- Unit-Testing Domain: **Jest**
- Assertions
- Parametrized Testing
- TestDoubles: Stubs & Spys
- UI-Unit-Testing: **Enzyme**
- Test-your-Tests: **Stryker**
- E2E: **Cypress**

IMAGE CREDITS

- Typewriter by rawpixel on pixabay, Licence CC0
- Dan North <http://dannorth.net/bio/>
- Space Image by Gerd Altmann from Pixabay, Licence CC0
- Animal Mouse by Tiburi on Pixabay, CC0
- Punchcard Wikimages on Pixabay, CC0
- London Photo by Luca Micheli on Unsplash, CC0
- Fire Motorcycle Stunt by digihanger on Pixabay, CC0
- Weights by Arcaion on Pixabay, CC0
- Home Office Workstation by Free-Photos on Pixabay, CC0

- <http://www.flickr.com/photos/janodecesare/9069301176/sizes/k/> CC BY-NC-ND 2.0
- http://www.flickr.com/photos/mike_nelson/4723888594/sizes/o/in/photostream/ CC BY 2.0
- <http://www.flickr.com/photos/kaptainkobold/3800229848/sizes/o/in/photostream/> CC BY-NC-ND 2.0
- http://www.flickr.com/photos/a_ninjamonkey/3565672226/sizes/o/in/photostream/ CC BY-NC-SA 2.0
- <http://www.flickr.com/photos/gisellenw/7683661450/sizes/l/in/photostream/> CC BY 2.0
- <http://www.flickr.com/photos/kaptainkobold/3273951718/sizes/o/in/photostream/> CC BY-NC-ND 2.0
- <http://www.flickr.com/photos/rogersg/3814863064/sizes/l/in/photostream/> CC BY-SA 2.0
- <https://www.flickr.com/photos/loufi/3500076/> CC BY 2.0
- <http://www.flickr.com/photos/seandreilinger/133305683/sizes/o/in/photostream/> CC BY-NC-SA 2.0

OTHER LINKS AND REFS

- TDD Origin: <https://arialdomartini.wordpress.com/2012/07/20/you-wont-believe-how-old-tdd-is>
- String Calculator by Roy Osherove: <http://osherove.com/tdd-kata-1/>
- Studies of TDD:
<http://biblio.gdinwiddie.com/biblio/StudiesOfTestDrivenDevelopment>
- State of JS: <https://2018.stateofjs.com/testing/overview/>
- TJ. Holowaychuck: <https://medium.com/@kelas/how-is-tj-holowaychuk-so-insanely-productive-604818b4e9eb>

Slides on Github

@ codecentric

Twitter: @MarcoEmrich

Brown Paper Bag Lunches

@ codecentric

<https://www.codecentric.de/leistungen/loesungen/brown-paper-bag-lunches>

WORKSHOP

@ codecentric

<https://www.codecentric.de/schulung/javascript-testing-2019>