

Innovative Approaches to Test Automation: Beyond the Ordinary

João Coelho

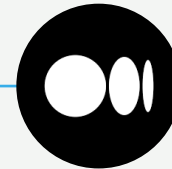
QA Automation Engineer @
Talkdesk



About me



joãocoelho10



@joaovitorcoelho10



joaovitorcoelho10@gmail.com



author-
3778566753

João Coelho
QA Automation Engineer @ Talkdesk

Summary



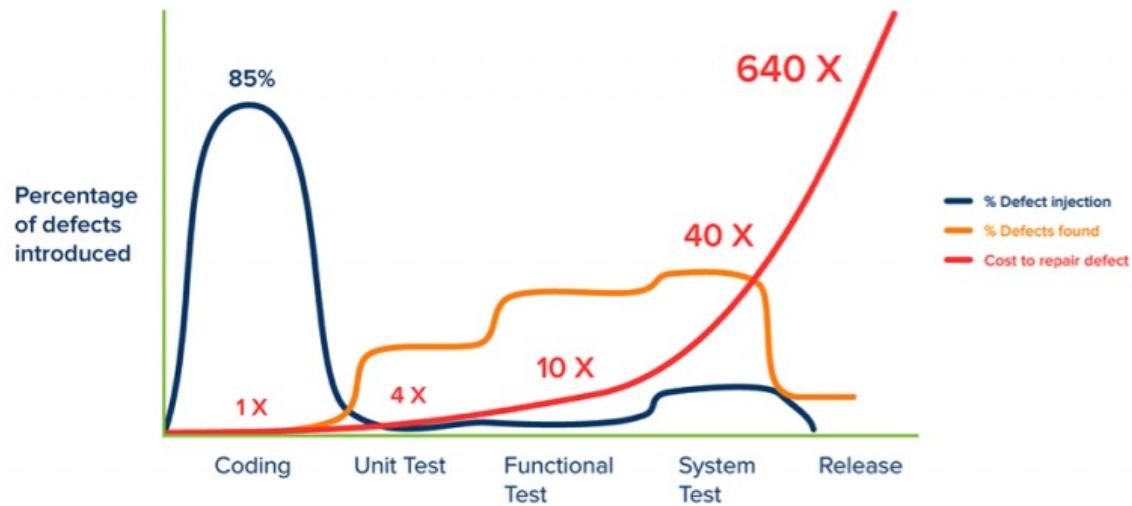
Property-based Testing



Performance Testing



UI Monkey Testing



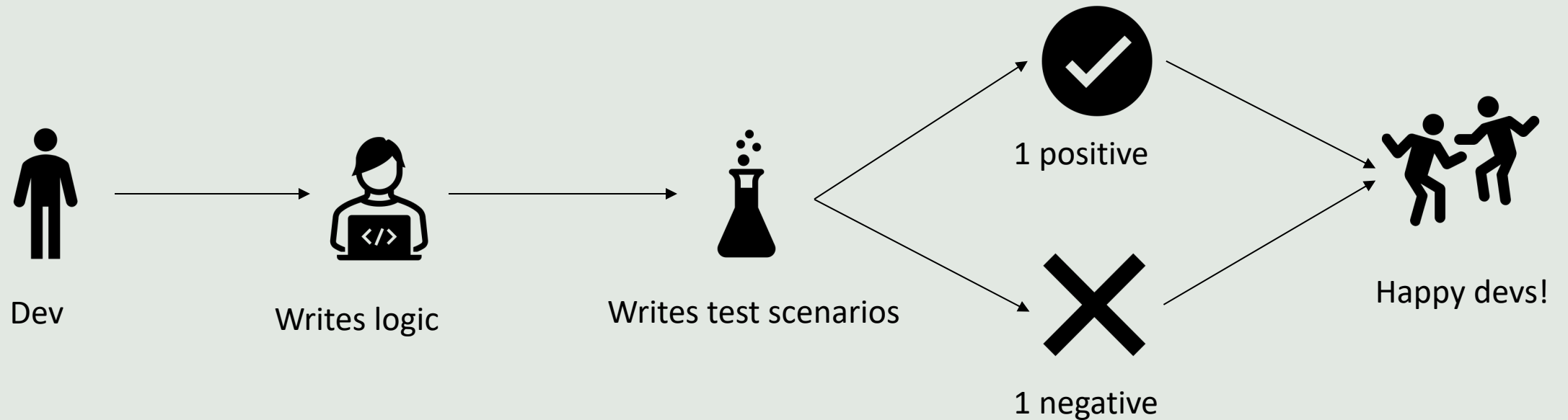
Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality*.

(Fun) fact

- Did you know that the **annual** cost incurred by companies for writing unit tests averages **£14,287 per developer?**

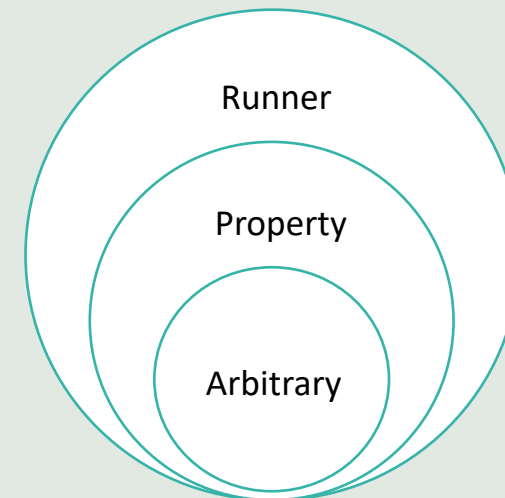
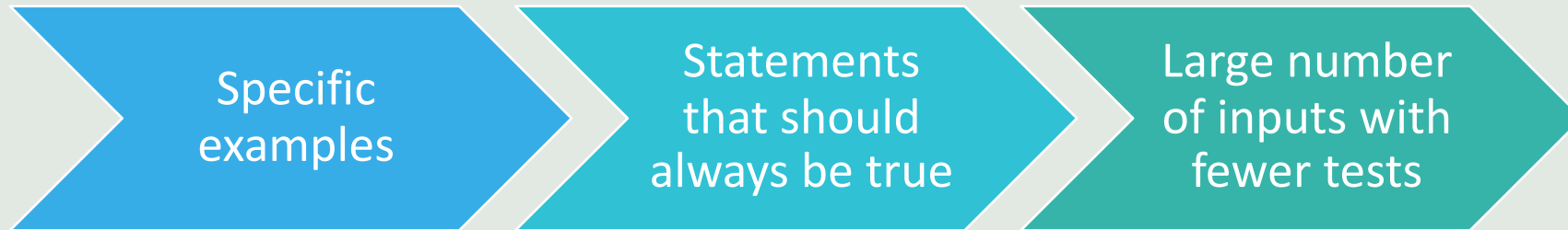
Source: 2019 Diffblue Developer Survey

The common workflow – Unit Tests



- What about *edge cases* and *boundary conditions*?

Property-Based Testing



Advantages of fast-check



Generative Testing



Randomized Testing



Reproducibility



Shrinking
Capabilities

What is an arbitrary?

01

Primitives

- String
- Float
- Booleans

02

Composites

- Array
- Object
- Tuples

03

Combiners

- Option
- Oneof
- Constant

04

Fake Data

- File
- JSON
- UUID

05

Others

- IPv4
- Email
- Falsy

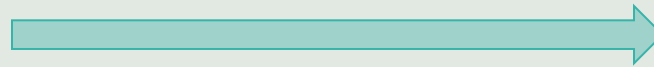
What is a property?

```
fc.property( fc.integer(),
```



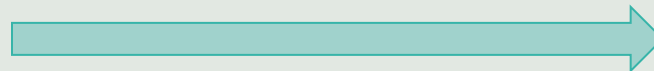
Define the **arbitraries**

```
(num) => {
```



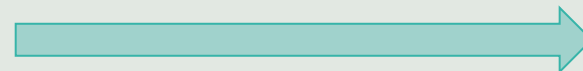
Assign the **arguments**

```
fc.pre(num % 2 === 0);
```



Define **preconditions**

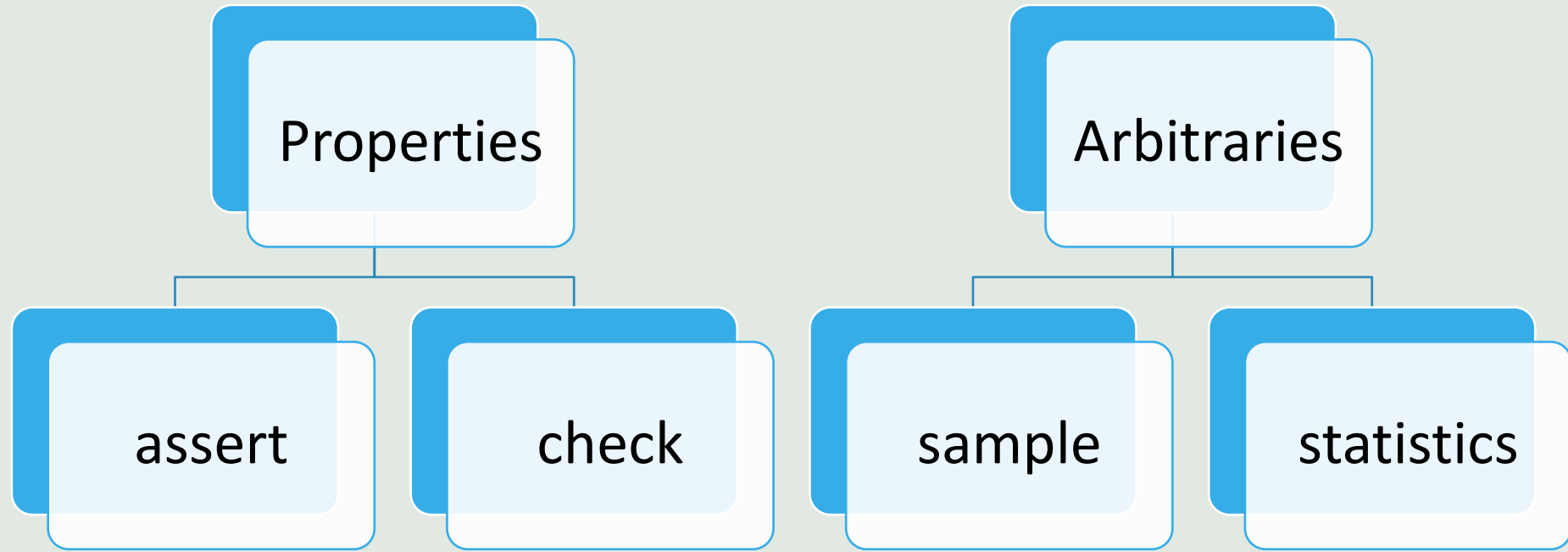
```
expect(isEven(num)).toBe(true);
```



Condition to validate

```
}  
)
```

What about runners?



Shrinking capability - Output

1) should always contain its substrings

Error: Property failed after 2 tests (seed: 1527422598337, path: 0:0):

["", "", ""]

Shrunk 1 time(s)

Got error: Property failed by returning false

Advanced usages

Fake Data

Fuzzing

Model based
testing

Race
conditions

Performance Testing - (Fun) facts



Google is the **most visited** website with **85.1 billion** visitors



88% of online users won't revisit to a **website** where they've had a **poor experience**



Slow loading speeds cost retailers a staggering **\$2.6 billion in lost sales annually**

Source: <https://www.forbes.com/advisor/business/software/website-statistics/>

What is Performance Testing?



What is its goal?

Evaluates how efficiently a system operates in different scenarios.



What does it evaluate?

- Responsiveness
- Speed
- Stability
- Ability to handle various levels of usage.



What does it find?

- Performance bottlenecks
- Vulnerabilities
- Areas requiring optimization

Some Performance Test Types



Comparison

Type	VUs/Throughput	Duration	Scenario
Smoke	+	+	Check baseline metrics
Load	++	++	Does it maintain performance?
Stress	+++	++	Above-average loads
Soak	++	+++	Prolonged use
Spike	++++	+	Seasonal events
Breakpoint	++++ (?)	++++ (?)	Find upper limits

Meet k6



Lightweight, developer-oriented and modern tool



Written in Go

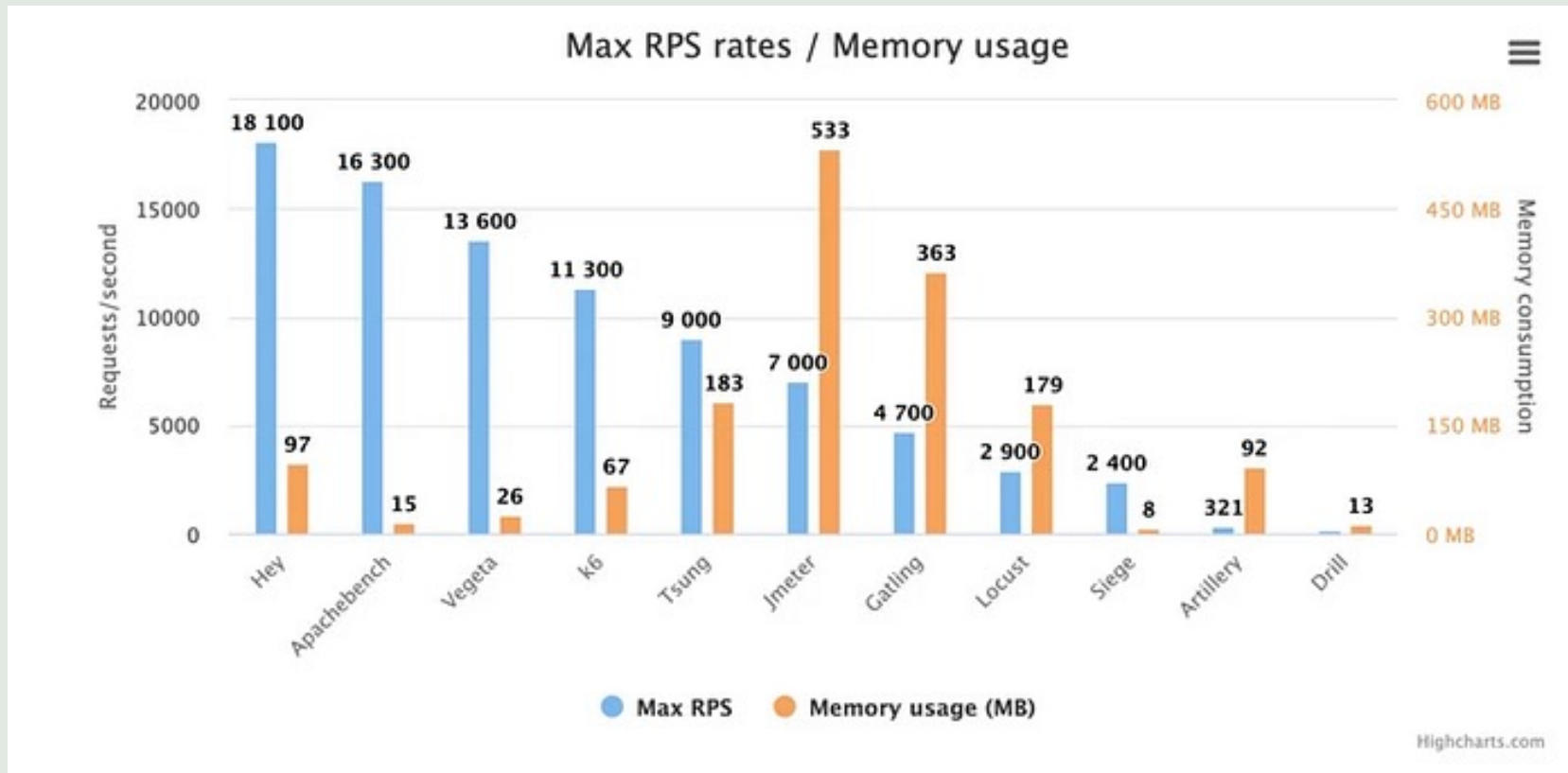


Scripts are developed using JavaScript



Supports HTTP 1 & 2, WebSockets and gRPC natively

What do I mean by lightweight?



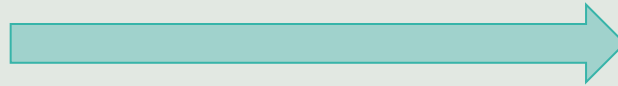
Source: Grafana Blog “Comparing k6 and JMeter for load testing”

The lifecycle of a k6 test



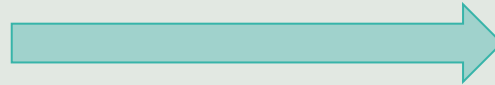
How easy it is to write a k6 script?

```
import { check } from 'k6';  
import http from 'k6/http';
```



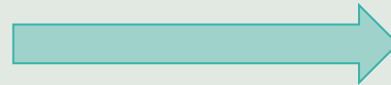
Start by the **imports**

```
export const options = {  
  thresholds: {  
    http_req_failed: ['rate<0.01']  
  }  
};
```



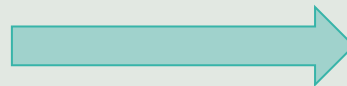
Define the **test options**

```
export default function () {  
  const res = http.get('http://test.k6.io/');
```



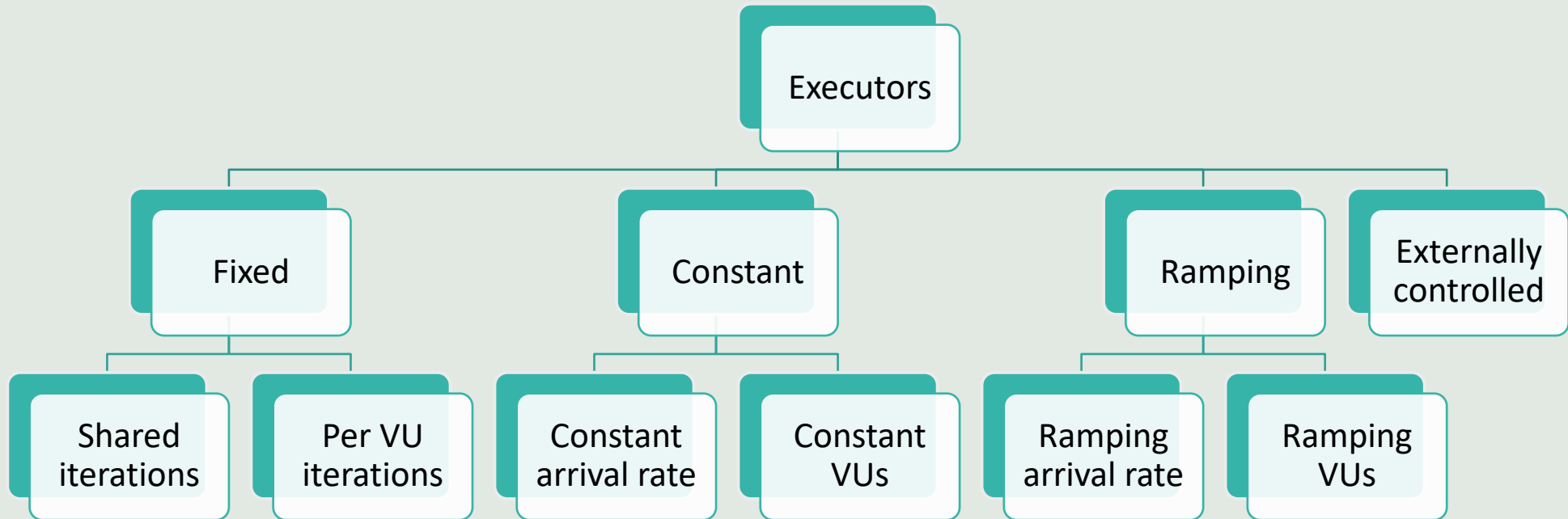
Perform your **request**

```
  check(res, {  
    'is status 200': (r) => r.status === 200;  
  })  
}
```

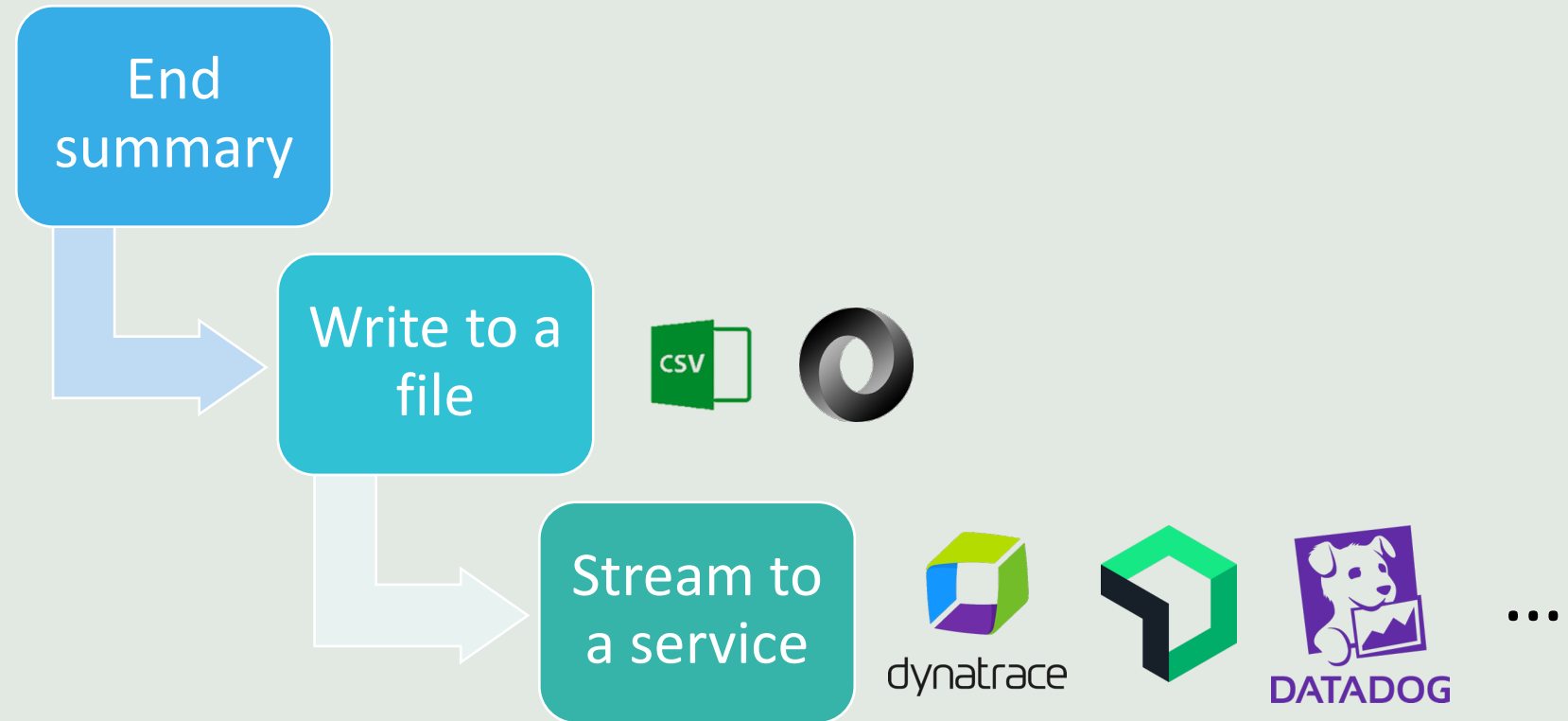


Make your **assertion**

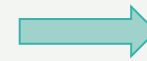
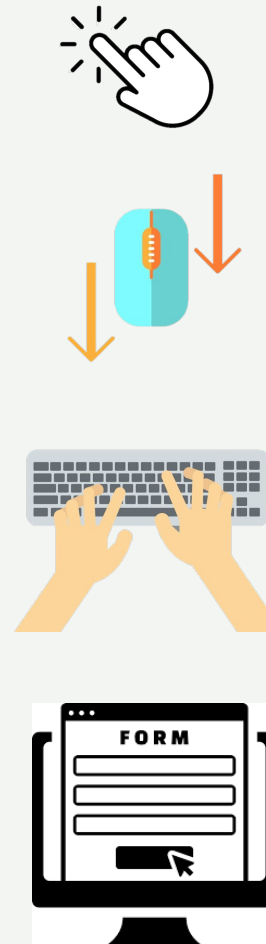
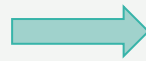
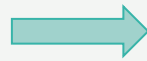
Scenarios



Output options



(UI) Monkey Testing



gremlins.js – Meet some terms



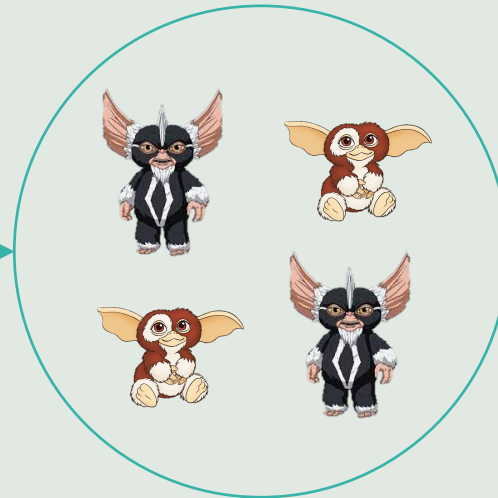
Gremlin

*clicker
toucher
formFiller
scroller
typer*



Mogwai

*alert
fps
gizmo*



Horde

unleash



Execution options



Playwright

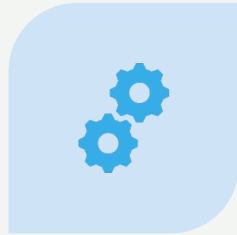


Cypress

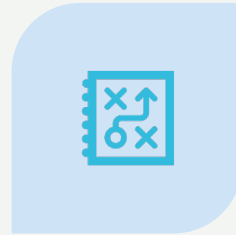


Bookmarklet code

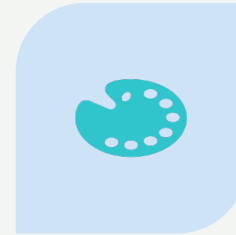
Execution customization



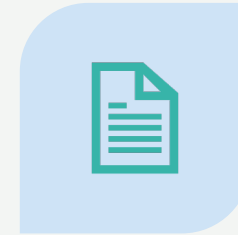
Control with which
elements will it
interact



Define the
distribution
strategy



Customizable
Gremlins



Customizable
logger

Execution Sample

```
tests > TS gremlins.spec.ts > test("run gremlins.js") callback
5 test("run gremlins.js", async ({ page }) => {
14
15   await page.goto("https://medium.com/@joaovitorcoelho");
16
17   const frame = page.mainFrame();
18   await frame.evaluate() => gremlins
19     .createHorde({
20       species: [
21         gremlins.species.toucher(),
22         gremlins.species.scroller()
23       ],
24       mogwai: [
25         ...gremlins.allMogwais,
26       ]
27     })
28   .unleash();
```

PROBLEMS DEBUG CONSOLE PORTS OUTPUT TERMINAL

zsh + v ... x

gremlin-tutorial

Ln 15, Col 60 Spaces: 2 UTF-8 LF () TypeScript

Checkout my articles


Effortless Performance Testing with k6: A Comprehensive Guide

João Coelho  · Follow
17 min read · Aug 10, 2023

 5    



Fast-Check: A Comprehensive Guide to Property-Based Testing

João Coelho  · Follow
8 min read · Feb 23, 2024

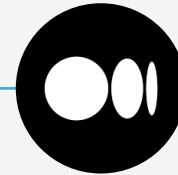
 70     



Thank you for watching!



joãocoelho10



@joaovitorcoelho10



joaovitorcoelho10@gmail.com



author-3778566753

João Coelho
QA Automation Engineer @ Talkdesk