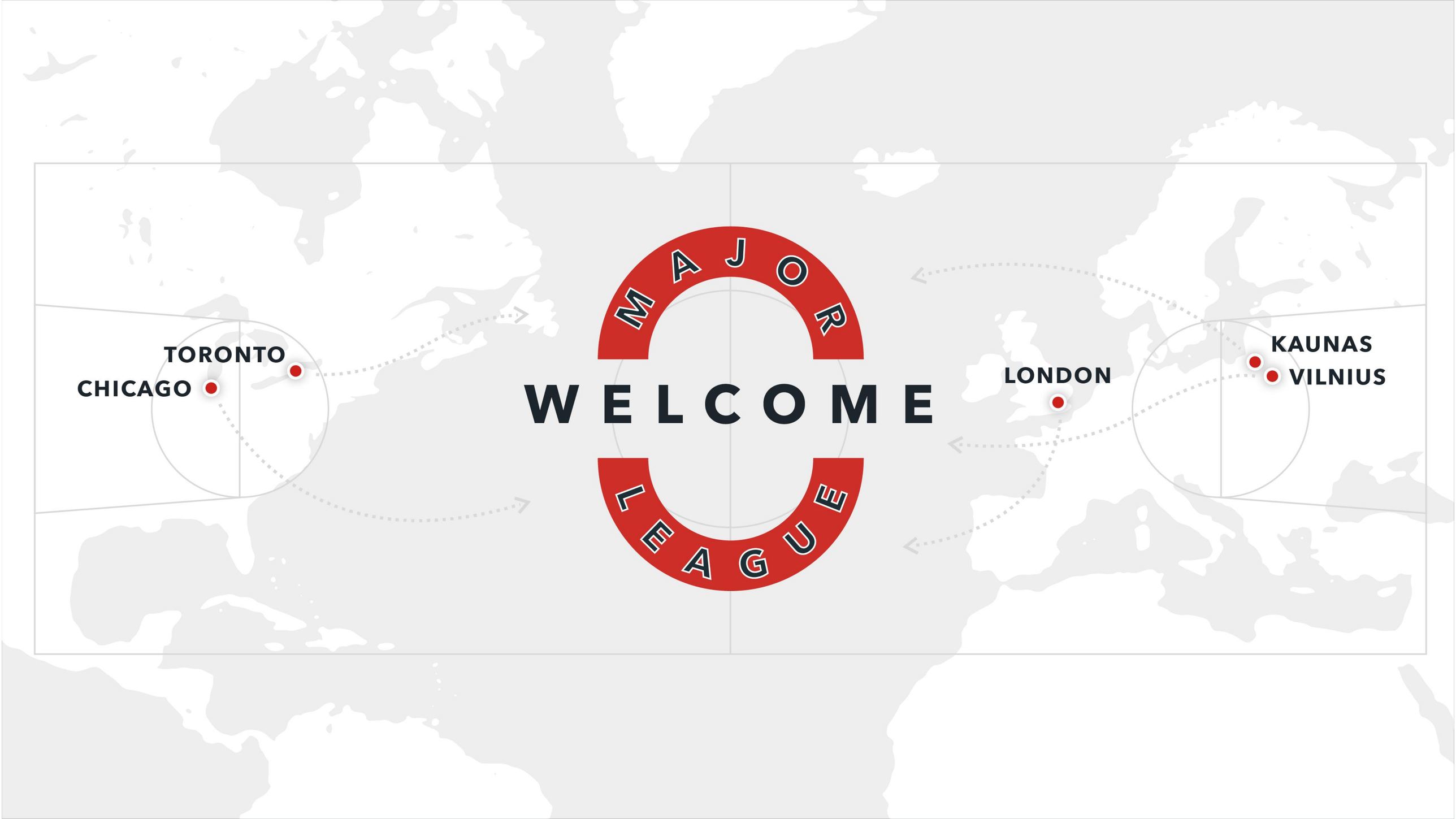
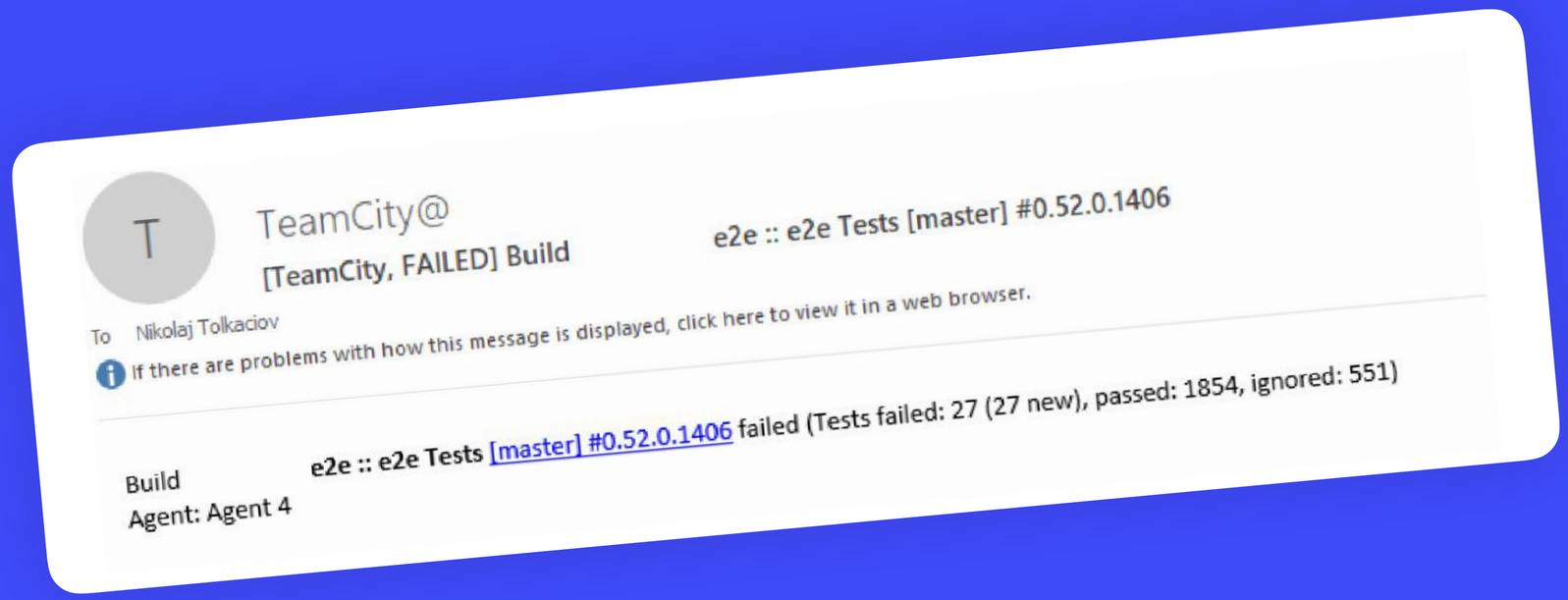

MAJOR
WELCOME
LEAGUE
TORONTO
CHICAGO
LONDON
KAUNAS
VILNIUS

DEVBRIDGE GROUP

5 ways to boost your automation effectiveness


T
TeamCity@
[TeamCity, FAILED] Build
e2e :: e2e Tests [master] #0.52.0.1406
To Nikolaj Tolkaciov
If there are problems with how this message is displayed, click here to view it in a web browser.
Build
Agent: Agent 4
e2e :: e2e Tests [master] #0.52.0.1406 failed (Tests failed: 27 (27 new), passed: 1854, ignored: 551)


T
TeamCity@
[TeamCity, FAILED] Build
e2e :: e2e Tests [master] #0.52.0.1406
To Nikolaj Tolkaciov
If there are problems with how this message is displayed, click here to view it in a web browser.
Build
Agent: Agent 4
e2e :: e2e Tests [master] #0.52.0.1406 failed (Tests failed: 27 (27 new), passed: 1854, ignored: 551)


T
TeamCity@
[TeamCity, FAILED] Build
Nikolaj Tolkaciov
problems with how this message is dis

Issues

1. Delusive test results
2. Difficulty locating elements
3. Test data maintenance
4. Demanding test infrastructure
5. Overcomplicated code

Developer
Home-brew
Automation

5+ years

Who am I?

International
Fortune 500
70+ test engineers

5+ years
Devbridge

Who am I?

30+ engineers
4 scrum teams
Bi-weekly releases

Who am I?

5+ years
Devbridge
30+ (at peak)

Issue #1

Delusive test results

Issue #1 **Delusive test results. False - Positive results.**

NEGATIVE

- Demotivate team members
- Waste time
- Decrease effectiveness
- Might hide actual bugs

POSITIVE

+ You get emails *

*Only positive if you're in need of communication

Possible solutions

→ Make automated test stable

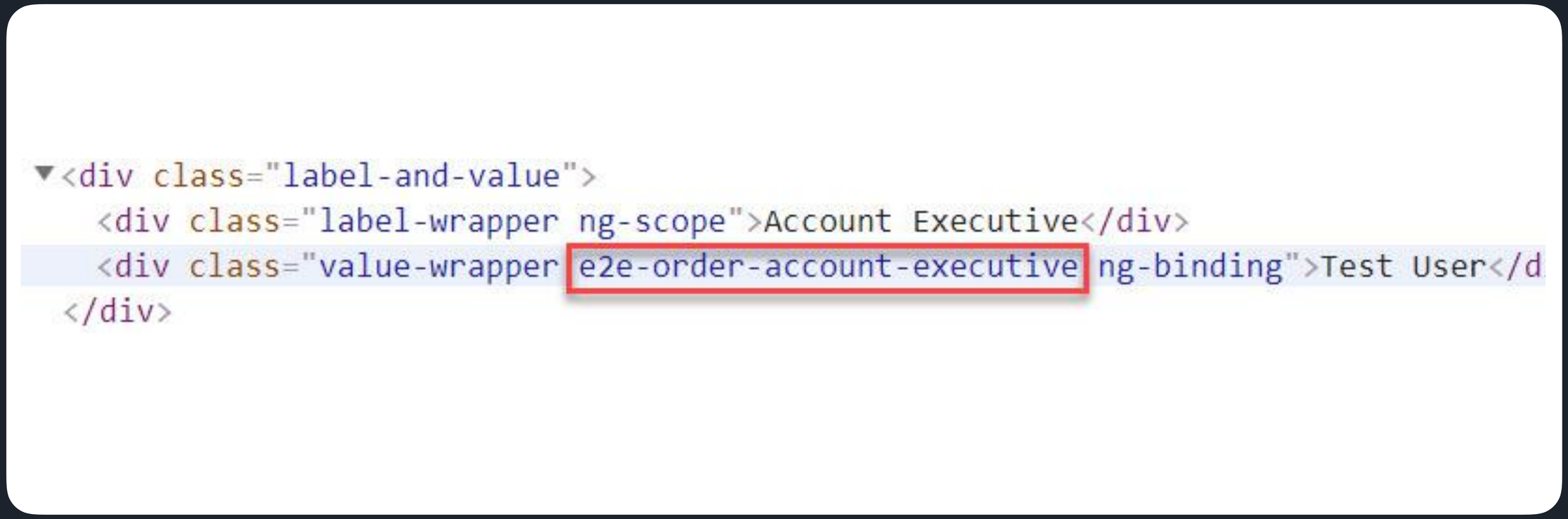
→ Rewrite test cases

→ Disable randomly failing ones and do them manually

Issue #2

Difficulty locating elements

Issue #2 **Difficulty locating elements. Injecting custom locators.**



Issue #2 **Difficulty locating elements. Injecting custom locators.**

obj/tasks.js

```
1  + module.exports = {
2  +     //DOM elements
3  +     createTaskButton : '[class="btn"]',
4  +     nameFilter : '/html/body/div[1]/div/div[2]/div/div/section/div/div[1]/di
```

Issue #2 **Difficulty locating elements. Injecting custom locators.**

POSITIVE

- More independency from developers
- Cleaner test code
- More flexibility in selecting required elements
- With developer participation this might D.O.R.

NEGATIVE

- Takes time to add
- Require basic HTML knowledge

Issue #3

Test data maintenance

Quiz

How you manage your test data?

A. I don’t use any test data, my tests are not requiring test data setup

B. I use same e2e steps to prepare data for tests

C. I use API calls to setup data

D. I use SQL scripts to setup data

E. I use DB backup and restore it before each run

Issue #3 **Test data maintenance. Database restore**

POSITIVE

- Fast and reliable test data preparation
- Easy test data management
- No garbage data
- Less maintenance in a long rung

NEGATIVE

- Takes time to implement

Issue #4

Demanding test infrastructure

Are you into DevOps?

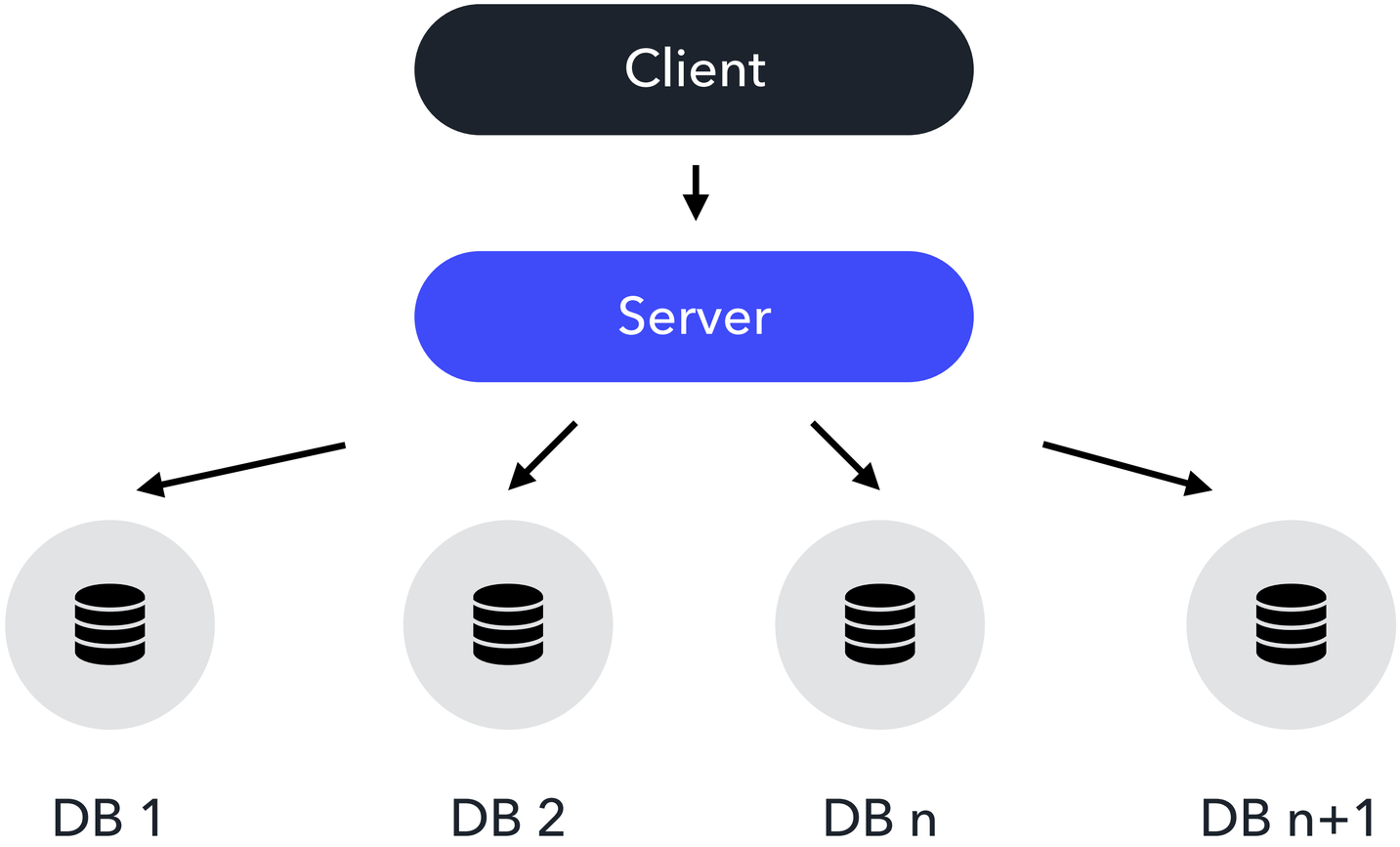
A. Automate boring and repetitive tasks.

B. Reduce human error probability.

C. Speed up workflows.

D. Work towards self-sufficient test automation.

We use Database Routing



Issue #4 **Demanding test infrastructure. Database Routing.**

POSITIVE

- Automated and manual testing same builds
- Less environments
- Scale development slots if needed

NEGATIVE

- Takes time to implement
- Not all builds are tested

Issue #5

Overcomplicated code

Can automated test's code be inferior quality compared to application under test?

Key principles

→ Reading your code should be pleasant.

- Variables and functions - self explanatory.
- Functions - focused, short and reusable.
- Global patterns, coding style and conventions.
- Perform code reviews.
- DRY rule.
- KISS principle.

Is there any benefit from keeping code clean?

Clean code is easier to collaborate on

Clean code is easier to refactor

Clean code is easier to augment

Clean code is easier to maintain

Clean code is easier to port

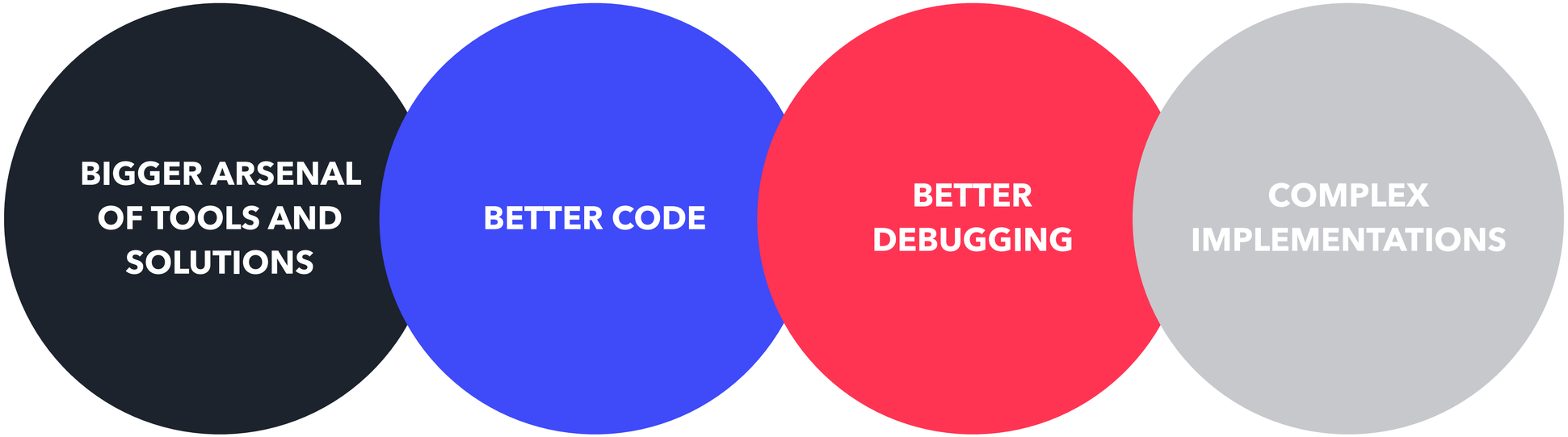
Clean code is easier to scale

Collaborate
Refactor
Augment
Maintain
Port
Scale

How to do it?


Make some research
Participate in workshops
Ask developers

Collaboration with developers



Is it all only positive experience when collaborating with developers?

- `"uninitialized": "always"` requires one variable declaration for uninitialized va
- `"uninitialized": "never"` requires multiple variable declarations for uninitialize
- `"uninitialized": "consecutive"` requires consecutive variable declarations for

always

Examples of **incorrect** code for this rule with the default `"always"` option:

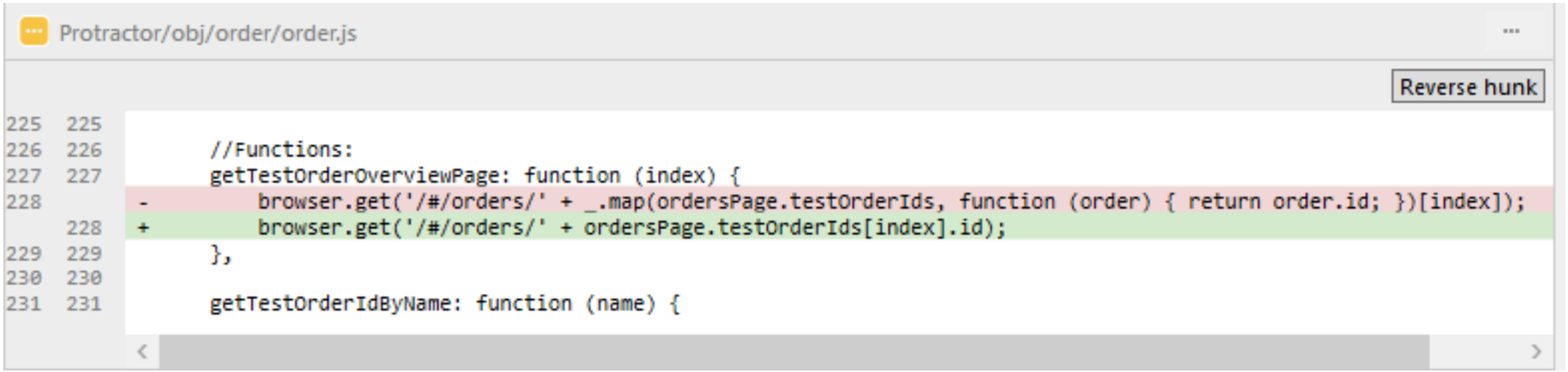
```
/*eslint one-var: ["error", "always"]*/
/*eslint-env es6*/

function foo() {
    var bar;
    var baz;
    let qux;
    let norf;
}
```

Examples of **correct** code for this rule with the default `"always"` option:

```
/*eslint one-var: ["error", "always"]*/
/*eslint-env es6*/

function foo() {
    var bar,
        baz;
    let qux,
        norf;
}
```


Protractor/obj/order/order.js
Reverse hunk
225 225 //Functions:
226 226 getTestOrderOverviewPage: function (index) {
228 - browser.get('/#/orders/' + _.map(ordersPage.testOrderIds, function (order) { return order.id; })[index]);
228 + browser.get('/#/orders/' + ordersPage.testOrderIds[index].id);
229 229 },
230 230
231 231 getTestOrderIdByName: function (name) {

```
const _ = require('lodash');

/**
 * @class
 * @param {string[]} headers
 * @param {Array.<Array.<string>>} data
 */
const DataTableWithHeaders = function (headers, data) {
    /**
     * @returns {Array.<Array.<{value: string, header: string}>>}
     */
    this.getDataRows = function() {
        return _.map(data, (row) => _.map(row, (column, index) => ({ header: headers[index], value: column })));
    }

    /**
     * @returns {Array.<string>}
     */
    this.getHeaders = function() {
        return headers;
    }

    this.get = function() {
        return _.concat(headers, data);
    }

    /**
     * @param {string} header
     * @returns {Array.<string>}
     */
    this.getColumnValues = function(header) {
        const index = _.indexOf(headers, header);
        return _.map(data, (row) => row[index]);
    }
};
```

Summary

1. Reduce False-Positive results to zero
2. Injecting custom locators to the application
3. Database restoring for test setup and tear-down
4. Invest into infrastructure for E2E tests
5. Promote clean code practices

nikolaj.tolkaciov@gmail.com

Questions?

DEVBRIDGE GROUP

Thank You.

Nikolaj Tolkačiov
Lead Test Engineer