



Корпоративная платформа хранения
и обработки больших данных

Как внедрить автоматизацию тестирования, если ты рядовой QA инженер?

Dmitry Gusakov
Arenadata Software

The logo for SQA DAYS #29, featuring the letters "SQA" in a bold, dark blue font with a red checkmark inside the "Q". Below "SQA" is a horizontal line, and below that is the text "DAYS #29" in a dark blue font, with "#29" in red.

SQA
DAYS #29

Привет, меня зовут Дима



**“Полноценное
обеспечение качества
невозможно без
автоматизации.”**

Пришел в QA из DevOps

Более 5 лет занимаюсь различными видами автоматизированного тестирования

Около 3-х лет руковожу процессами обеспечения качества

Верю, что залог успеха в том, чтобы делится знаниями и опытом

Активно участвую и организую QA events

ARENADATA

Это кто такие?

ARENADATA — ведущий российский разработчик платформ сбора и хранения данных для компаний, которые хотят изменить бизнес с помощью цифровизации процессов и сложной аналитики, основанной на использовании Big Data.

Наши решения строятся
на технологиях
с открытым кодом
и помогают работать с



Большие
данные



Интернет
вещей



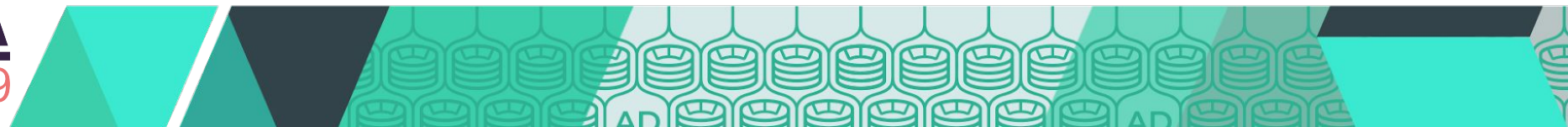
Машинное
обучение



Связанные
технологии

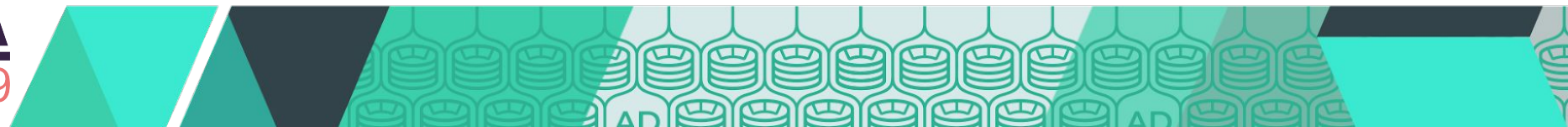


QA + Automation





QA + Automation, или для чего нам нужна автоматизация?

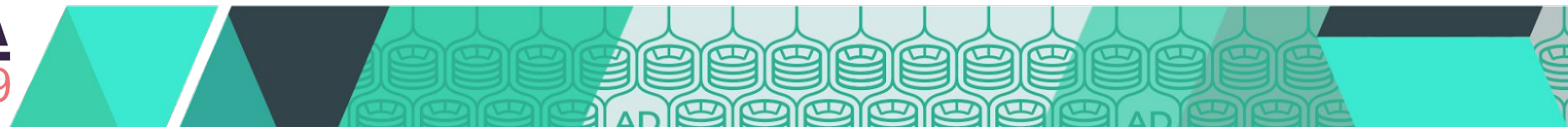


Для чего нужна автоматизация?





Первые шаги: Выбор инструментов



Какой язык выбрать?

Лучший язык для начинающего автоматизатора - язык разработки тестируемого приложения



- ✓ Будет у кого учиться
- ✓ Можно рассчитывать на ревью кода
- ✓ Разработка будет понимать ваши тесты



- ✗ Скромный выбор инструментов для тестирования
- ✗ Порог вхождения в язык может быть выше среднего

Какой язык выбрать?

Если рядом нет разработчиков, или других автоматизаторов то берем **Python + Pytest**, **Java + TestNG** или **JS + MochaJS**



Почему я за **Python + Pytest**?

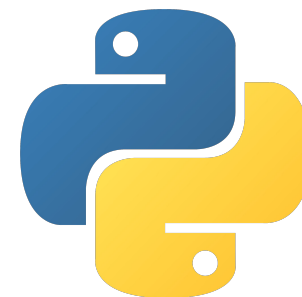
- ✓ Низкий порог вхождения
- ✓ Большой количество проверенных инструментов
- ✓ Активная поддержка community

Простейший тест на Python + Pytest

test_arithmetic.py

```
def test_simple():  
    result = sum([1,3])  
    assert result == 4
```

```
$ pip install pytest  
$ pytest test_arithmetic.py
```



API Test Automation

Java

Rest-assured



rest-assured.io

Python

Requests



docs.python-requests.org

JS

SuperTest



No image available

npmjs.com/package/supertest

Java + Rest-assured

```
    @Test
    public void testCreateTester() {
        given()
            .header("Content-type", "application/json")
            .body("{\"name\": \"GenericTester\"}");
        when()
            .post("/my/awesome/api/testers");
        then()
            .statusCode(201)
            .body("name", equalTo("GenericTester"));
    }
```

Python + Requests

```
def test_create_tester():  
    response = requests.post("/my/awesome/api/testers", json={"name": "GenericTester"})  
  
    assert response.status_code == 201  
    assert response.json["name"] == "GenericTester"
```

JS + SuperTest

```
describe('POST /my/awesome/api/testers', function() {  
  it('successful user creation', function(done) {  
    request(app)  
      .post('/my/awesome/api/testers')  
      .send({name: 'GenericTester'})  
      .expect(201)  
      .then(response => {  
        assert(response.name, 'GenericTester')  
        done();  
      })  
  });  
});
```

API Test Automation

Java

Rest-assured



rest-assured.io

Python

Requests



docs.python-requests.org

JS

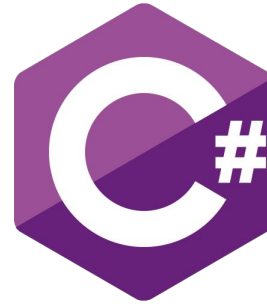
SuperTest



No image available

npmjs.com/package/supertest

API Test Automation



А где **C#**, **C++**, **Golang**,
PHP, **Ruby** и другие?



UI Test Automation



Selenium IDE



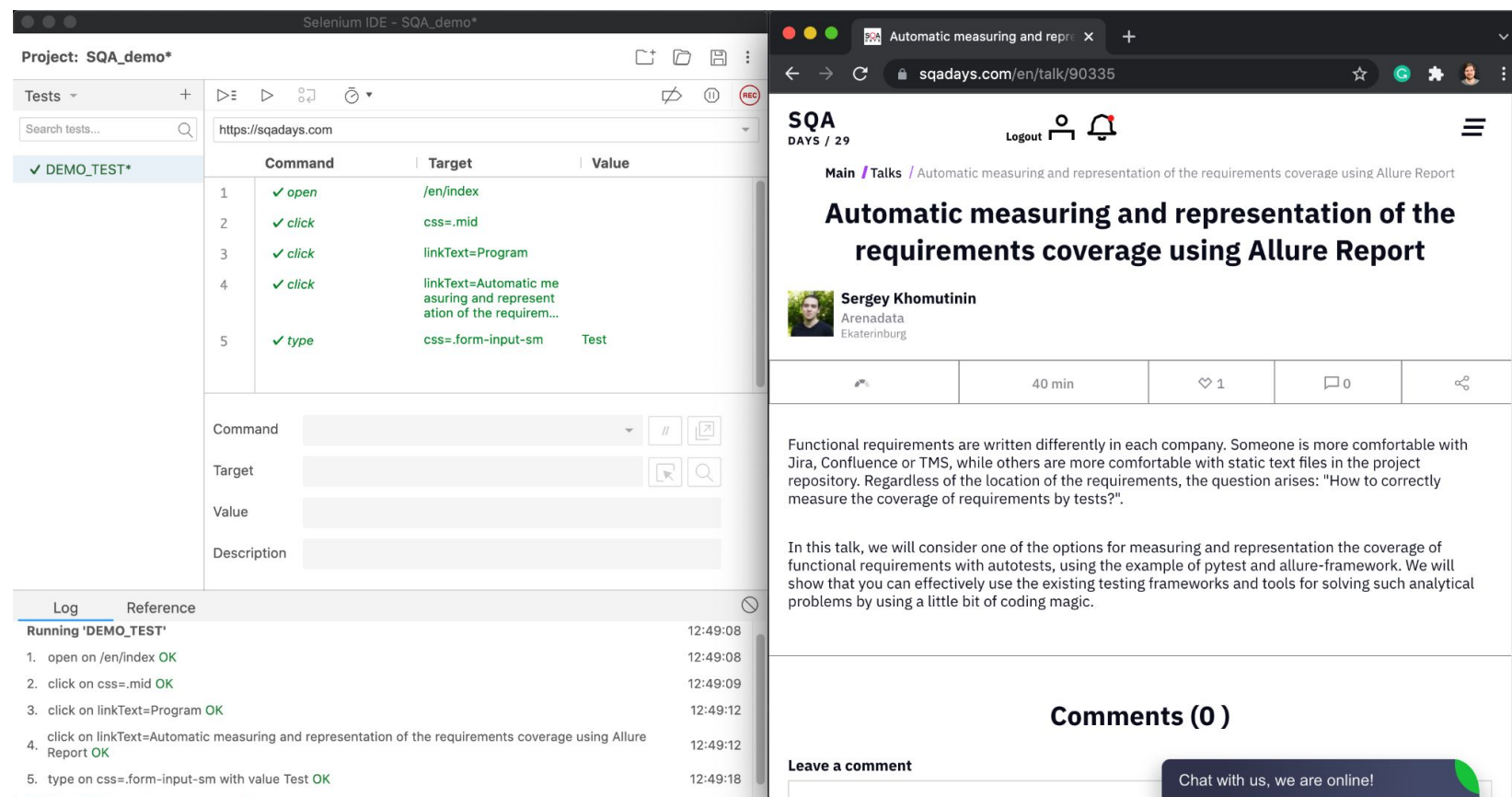
Beginner

Pro

UI Test Automation



Selenium IDE



The image shows two overlapping windows. The left window is Selenium IDE, displaying a test suite named 'DEMO_TEST*' with five steps:

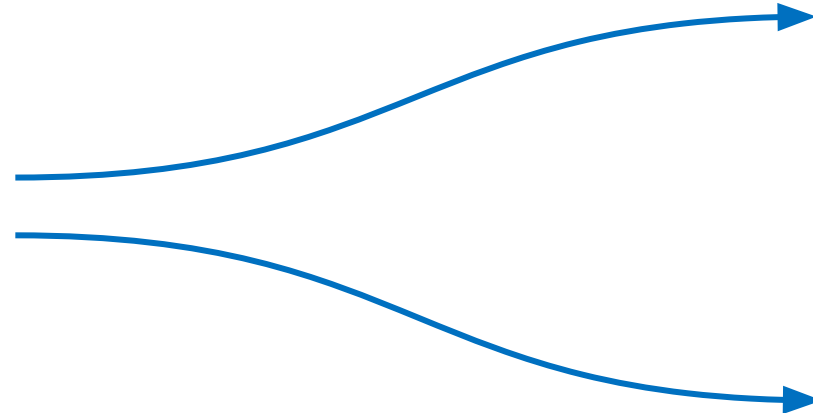
Step	Command	Target	Value
1	open	/en/index	
2	click	css=.mid	
3	click	linkText=Program	
4	click	linkText=Automatic measuring and representation of the requirements coverage using Allure Report	
5	type	css=.form-input-sm	Test

The right window is a browser showing a talk page on sqadays.com. The title is 'Automatic measuring and representation of the requirements coverage using Allure Report' by Sergey Khomutinin. The page content includes a paragraph about functional requirements and a section for comments.

UI Test Automation



Selenium IDE



Selenide
CONCISE UI TESTS IN JAVA



Selenium-Python
/ Selene

Beginner

Pro

UI Test Automation JS

cypress.io



Playwright

Beginner

Pro

Playwright

✓ Настоящая кросс браузерность

✓ Нативное мокирование

✓ Скорость работы

✓ Собственный раннер



✗ Высокий порог входа

✗ Собственный раннер

✗ Что точно НЕ стоит брать? ✗

Любые **BDD** или **ATDD** фреймворки

✗ Что точно НЕ стоит брать? ✗

Любые **BDD** или **ATDD** фреймворки

Comment

@tag

Feature: Eating too many cucumbers may not be good for you

Eating too much of anything may not be good for you.

Scenario: Eating a few is no problem

Given Alice is hungry

When she eats 3 cucumbers

Then she will be full

✗ Что точно НЕ стоит брать? ✗

Любые **BDD** или **ATDD** фреймворки

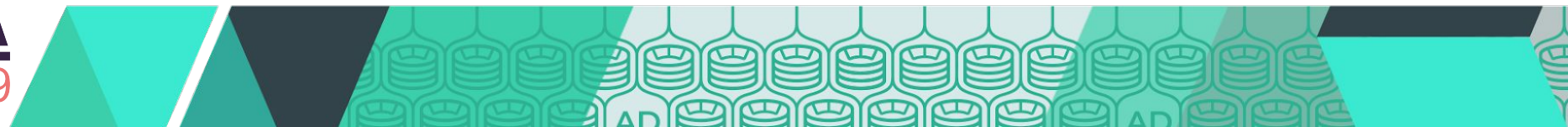


Почему:

- Большое количество кода для написания простого теста
- Низкая гибкость
- Сложности при масштабировании проекта



Идем в бой! Автоматизируем для себя



С каких тестов начать автоматизацию?

Выбираем из **регресс** или наиболее **часто падающих** тестов

Ищем наиболее **монотонные** для ручного тестирования

Берем самые **простые** для автоматизации

Выбираем 1, **самый понятный**

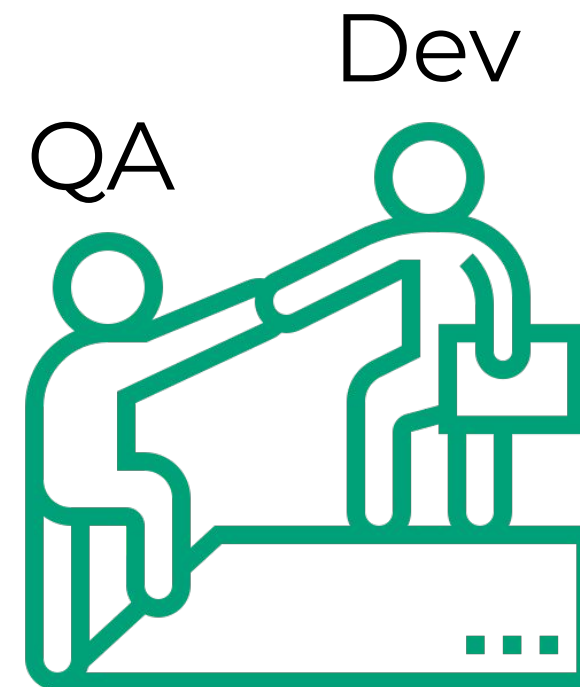


Profit!

Разработчики в помощь!

Чем разработчики могут помочь QA:

- Написать скелет тестового фреймворка
- Помочь в понимании синтаксиса языка
- Подсказать полезные библиотеки
- Провести ревью кода
- Предоставить SDK для работы с API
- Морально поддержать



Я не программист! Как мне писать этот ваш код?

Курсы и ресурсы

Бесплатные

Платные

 **sololearn**  **stepik**

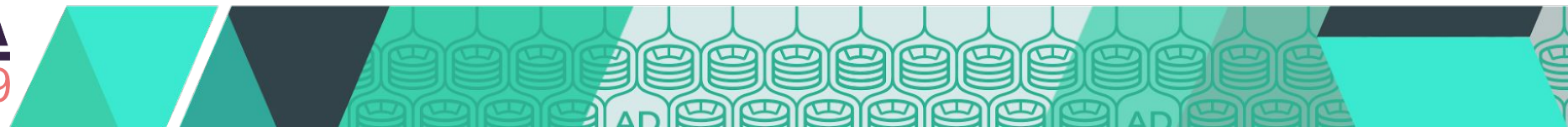
Яндекс Практикум

coursera **You Tube**

Skillbox



Типичные ошибки Как их избежать?



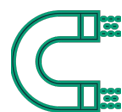
Топ 5 ошибок новичка автоматизатора



Давайте сразу сделаем все!



Остатки тестовых данных после теста



Зависимые тесты



Невозможность параллельного выполнения

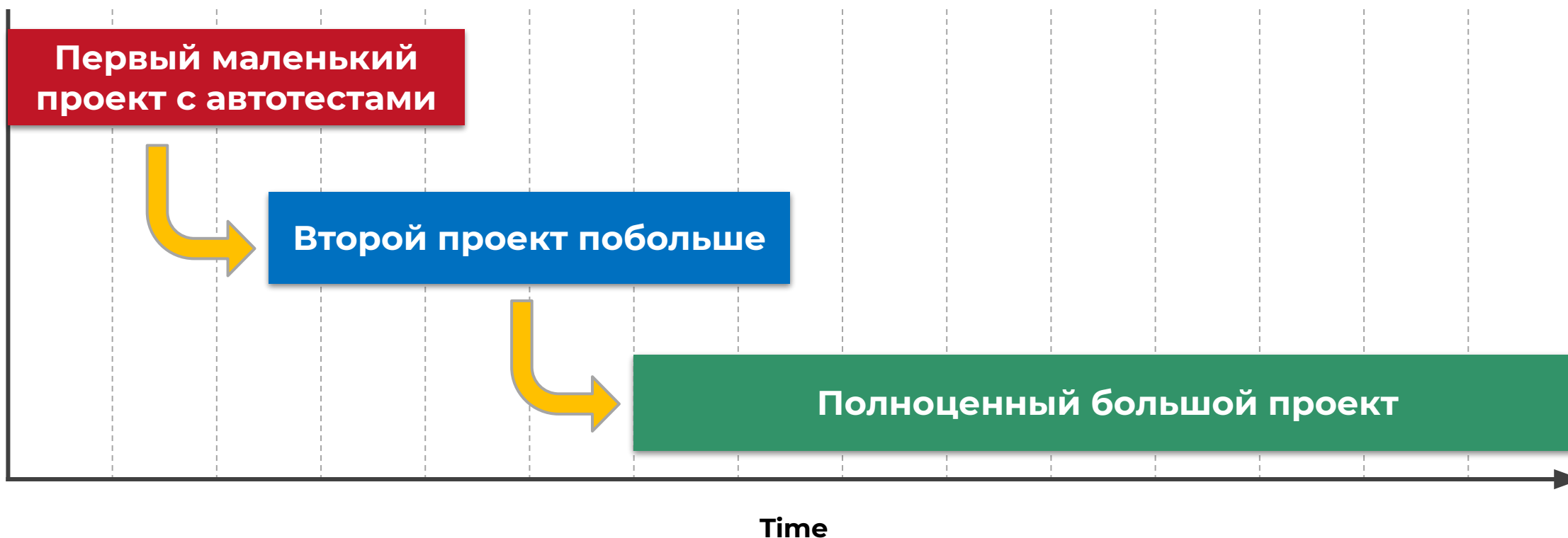


Привязка тестов к конкретному окружению

Как избежать типичных ошибок?



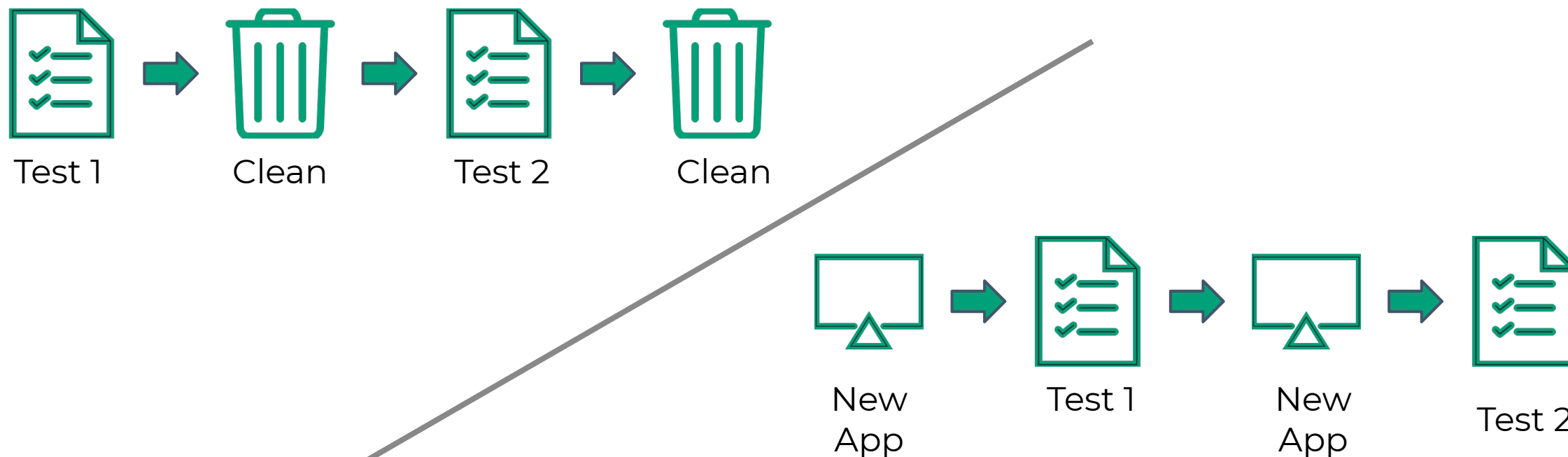
Давайте ~~сразу сделаем все!~~ двигаться постепенно!



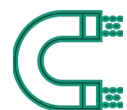
Как избежать типичных ошибок?



Остатки тестовых данных после теста



Как избежать типичных ошибок?



Зависимые тесты



Невозможность
параллельного выполнения



Test 1

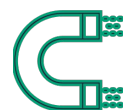


Test 2



Test 3

Как избежать типичных ошибок?



Зависимые тесты

&



Невозможность
параллельного выполнения



Test 3

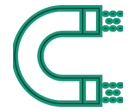


Test 1



Test 2

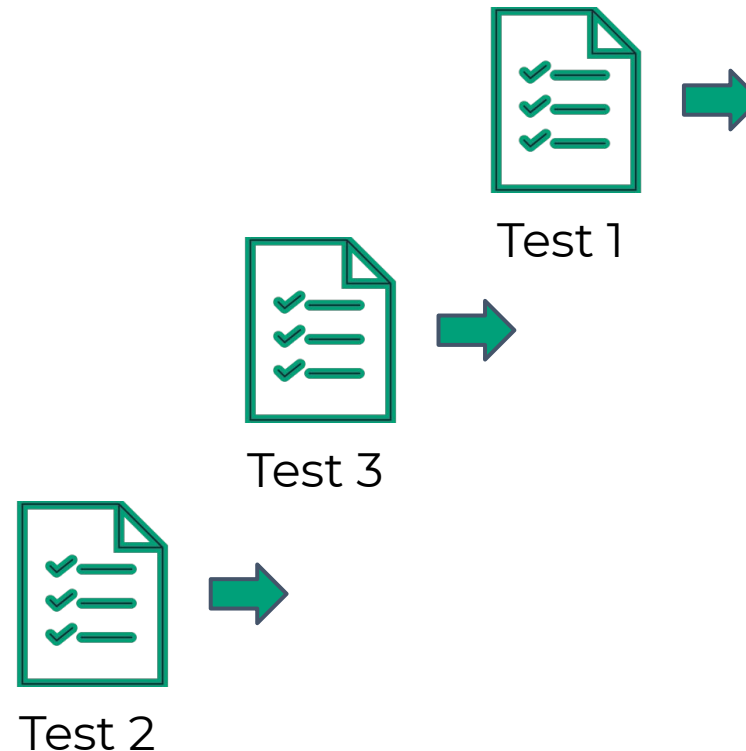
Как избежать типичных ошибок?



Зависимые тесты &

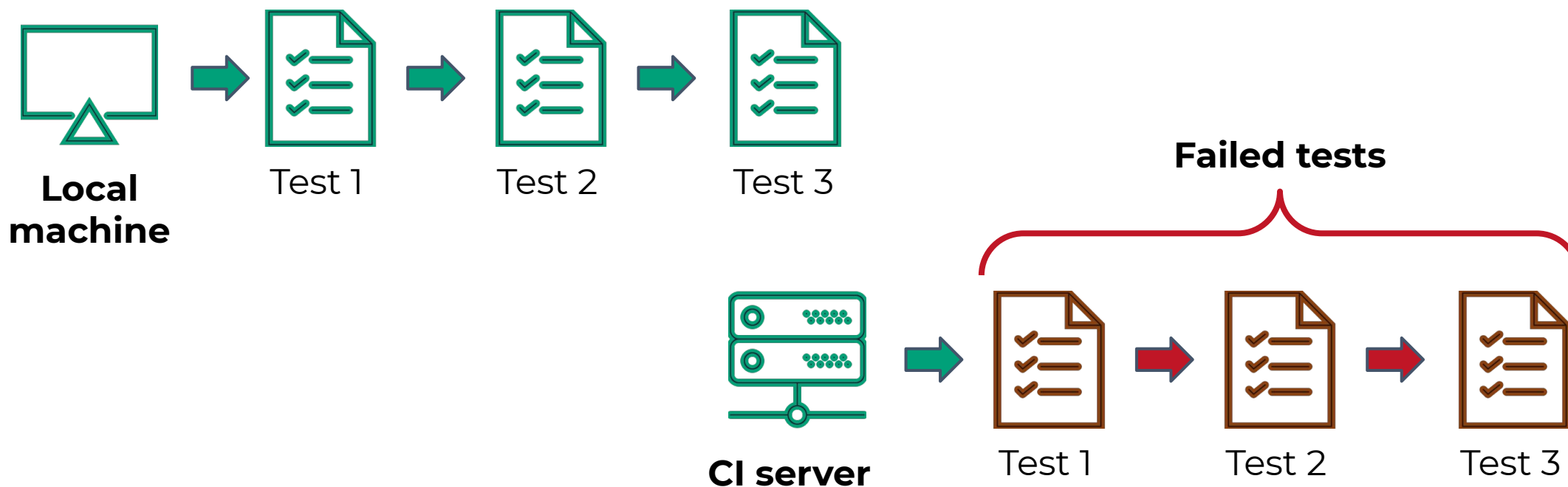


Невозможность
параллельного выполнения



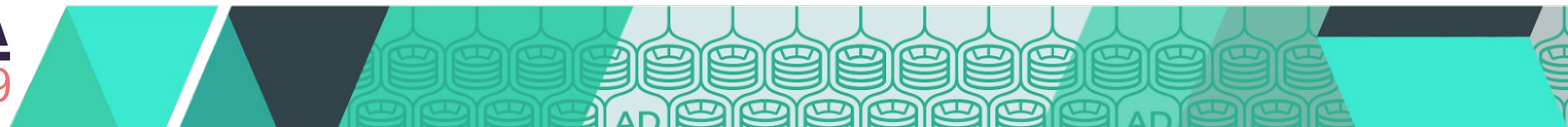
Как избежать типичных ошибок?

 Привязка тестов к конкретному окружению





Набираемся опыта Собираем метрики



Метрики эффективности

$$P = (A * N - D - M) * C$$

M - время ручного теста (часов)

A - время выполнения автотеста (часов)

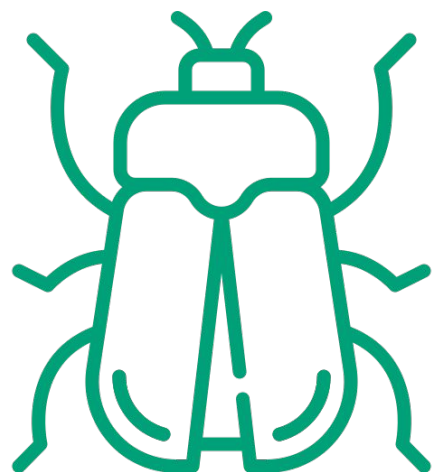
D - время разработки автотеста

N - количество запусков автотеста

P - **выгода от автоматизации**

C - стоимость часа работы QA

Метрики эффективности



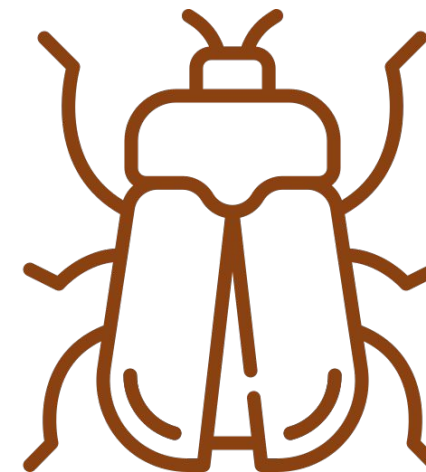
Bugs

From Manual
testing

VS

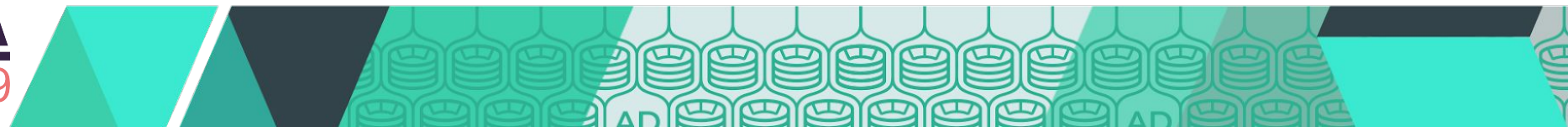
Bugs

From Automated
testing





Продаем автоматизацию руководству



Стратегии “продажи” автоматизации

Что не сработает?

Вы:

Давайте наймем больше людей для автоматизации. Те что есть не справятся!

Руководитель:

У нас и так бюджет по швам трещит! Нанимать людей нет возможности. Вы же как-то справляетесь.

Вы:

Давайте поручим разработке написание автотестов, они умеют писать код.

Руководитель:

А кто будет разработкой заниматься? У нас сроки релизов и без ваших автотестов плотные!

Стратегии “продажи” автоматизации

Менеджменту в первую очередь важны деньги.

Что не работает?

Вы:

Давайте наймем больше людей для автоматизации. Те что есть не справятся!

Руководитель:

У нас и так бюджет по швам трещит! Нанимать людей нет возможности. Вы же как-то справляетесь.

Вы:

Давайте поручим разработке написание автотестов, они умеют писать код.

Руководитель:

А кто будет разработкой заниматься? У нас сроки релизов и без ваших автотестов плотные!

Стратегии “продажи” автоматизации

Быстрее тестируем => Экономим время => **Экономим деньги**

Детальнее тестируем => Находим больше багов =>
=> Экономим время на их исправление => **Экономим деньги**

Тестируем без привлечения рук => Чаще тестируем =>
=> Быстрее находим и правим баги => **Экономим деньги**

Стратегии “продажи” автоматизации

Что может сработать?

Вы:

Уже 3 релиза мы используем автотесты для части регресса. Время на тестирование сократилось почти на день.

Руководитель:

Интересно. А мы можем сократить его еще?

Вы:

Конечно! Но если мы хотим и дальше внедрять автотесты нам могут понадобится еще несколько ребят.

Вы:

На последних релизах благодаря автотестам мы отловили несколько критичных багов.

Руководитель:

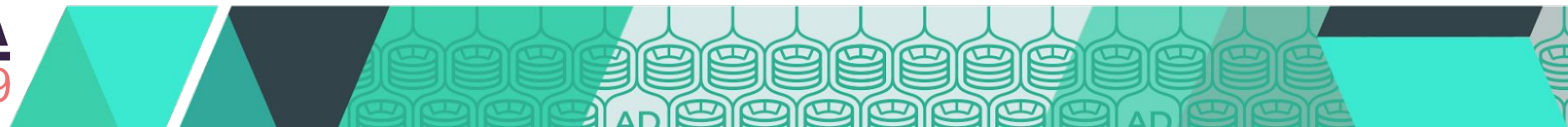
А почему при ручном тестировании их не нашли? Вы что плохо тестируете?

Вы:

Дело в том, что благодаря автотестам мы расширили количество вариаций тестовых данных. Руками такой объем тестирования едва ли был возможен



Что дальше?



Идем в светлое будущее с автотестами

После того как получен опыт в написании автотестов, собраны метрики и идея официально одобрена руководством путей может быть несколько:

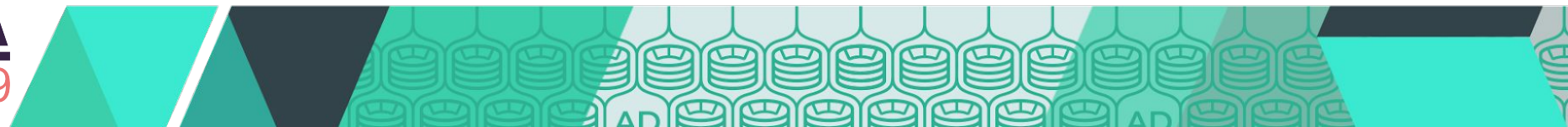
- **Продолжаем развивать автоматизацию на текущем проекте как поддержку ручным тестам**

ИЛИ

- **Переходим к внедрению полного автоматизированного регресса на проекте**



Подведем итоги



Выводы

- ✓ **Начинаем** автоматизацию **с простых**, но важных **тестов**
- ✓ Вдумчиво подходим к выбору **инструментов**
- ✓ Внедряем автоматизацию для **экономии своего времени**, а не для красивого достижения
- ✓ **Не стесняемся учиться** на проверенных курсах и материалах
- ✓ Лучшее **обоснование** пользы автоматизации - **практический результат**
- ✓ Автоматизация - это не сложно.
Разумная **автоматизация - это интересно** и выгодно!



Корпоративная платформа хранения
и обработки больших данных

Спасибо за внимание!

Гусаков Дмитрий
Head of QA

 gdv@arenadata.io

 [@Gusakov_dv](https://t.me/@Gusakov_dv)

SQA
DAYS#29

