

ТЕХНОАТОМ

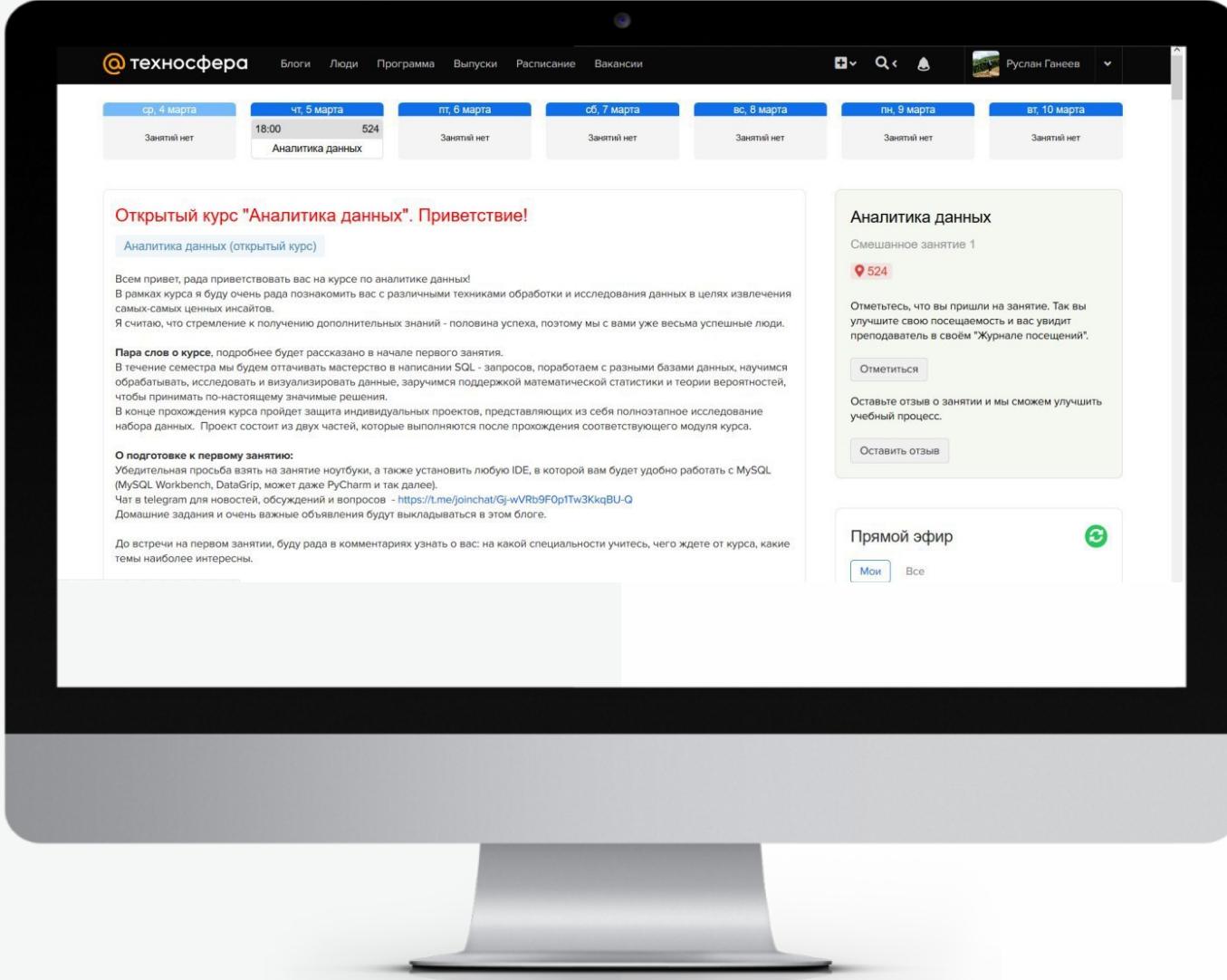
Автоматизация тестирования на Python

Илья Кириллов

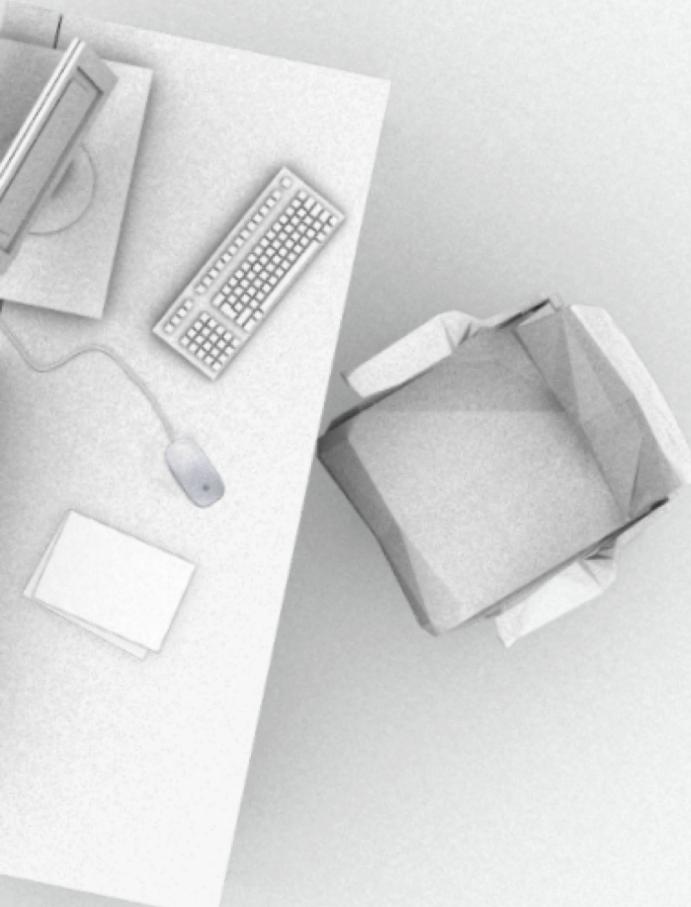


Вводная лекция

1. Основы тестирования ПО
2. Виды и цели автоматизированного тестирования
3. Инструменты и инфраструктура
4. Фреймворк py.test



Не забудьте
отметиться
на портале

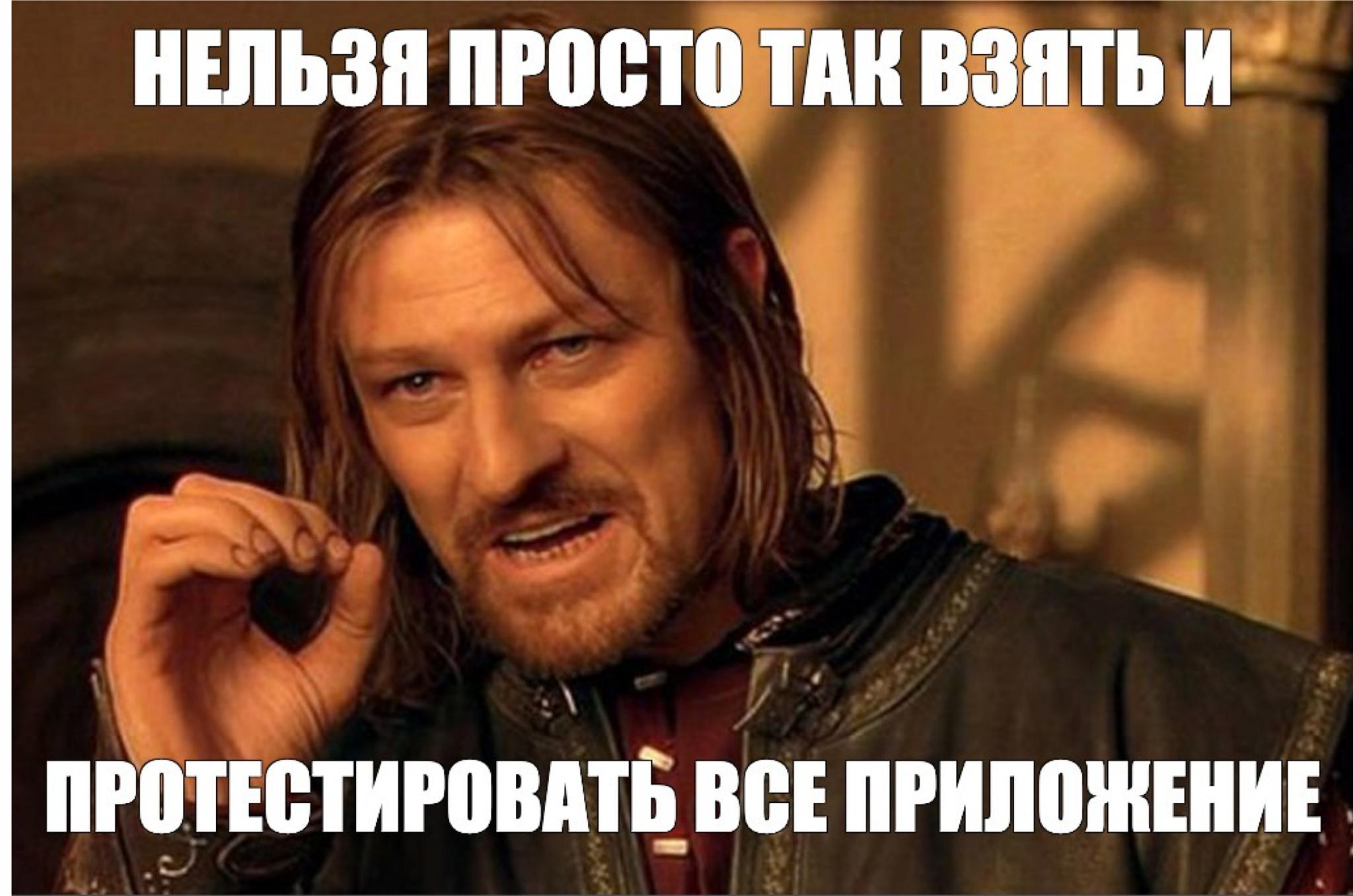


Основы тестирования ПО

Что такое тестирование?

Тестирование – процесс демонстрации отсутствия ошибок в приложении.

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ И



ПРОТЕСТИРОВАТЬ ВСЕ ПРИЛОЖЕНИЕ

Что такое тестирование?

Тестирование – процесс демонстрации того, что приложение работает правильно.

Что такое тестирование?

Тестирование – процесс выполнения программы с целью поиска ошибок.



Атрибуты качества ПО

- Функциональность
- Надежность
- Практичность/удобство
- Эффективность
- Сопровождаемость
- Мобильность

Методы обеспечения качества ПО

- Статический:
 - Статический анализ кода
 - Проверка спецификаций и требований
- Динамический:
 - Тестирование
 - Профилирование
 - Динамический анализ
- Формальный:
 - Верификация моделей

Немного терминов

- Тестировщик (тестер, тостер)
- Тест (test-case)
- Сценарий
- Баг (дефект, ошибка)
- Билд (сборка)
- Покрытие

Покрытие кода (code coverage)

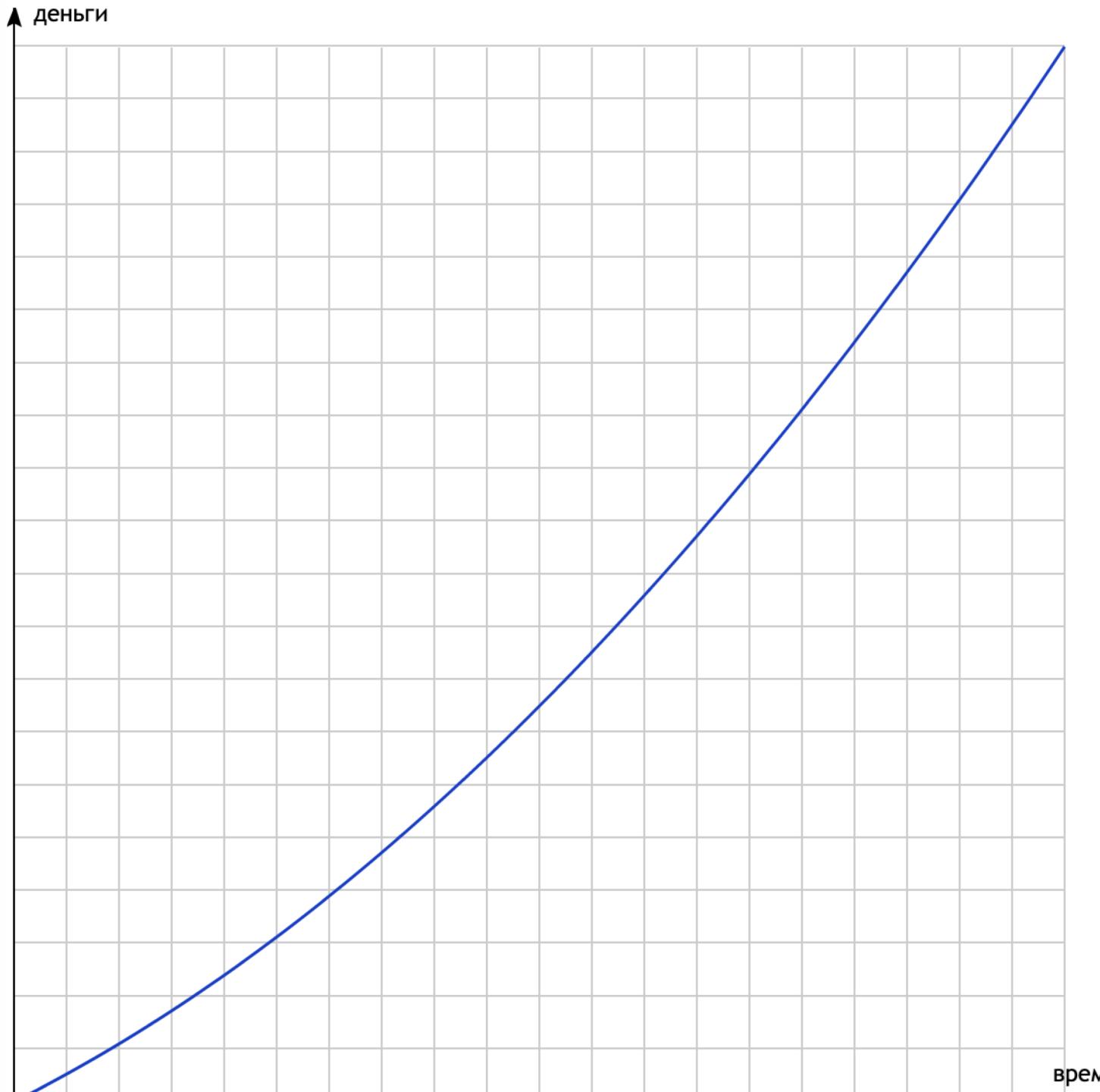
- Line coverage
- Branch coverage

```
a = input()  
b = input()  
  
if int(a) > 0 or int(b) < 10:  
    print('Success')  
else:  
    print('Failure')
```

Кто еще может заниматься тестированием?

- Разработчик
- Менеджер
- Заказчик

#013

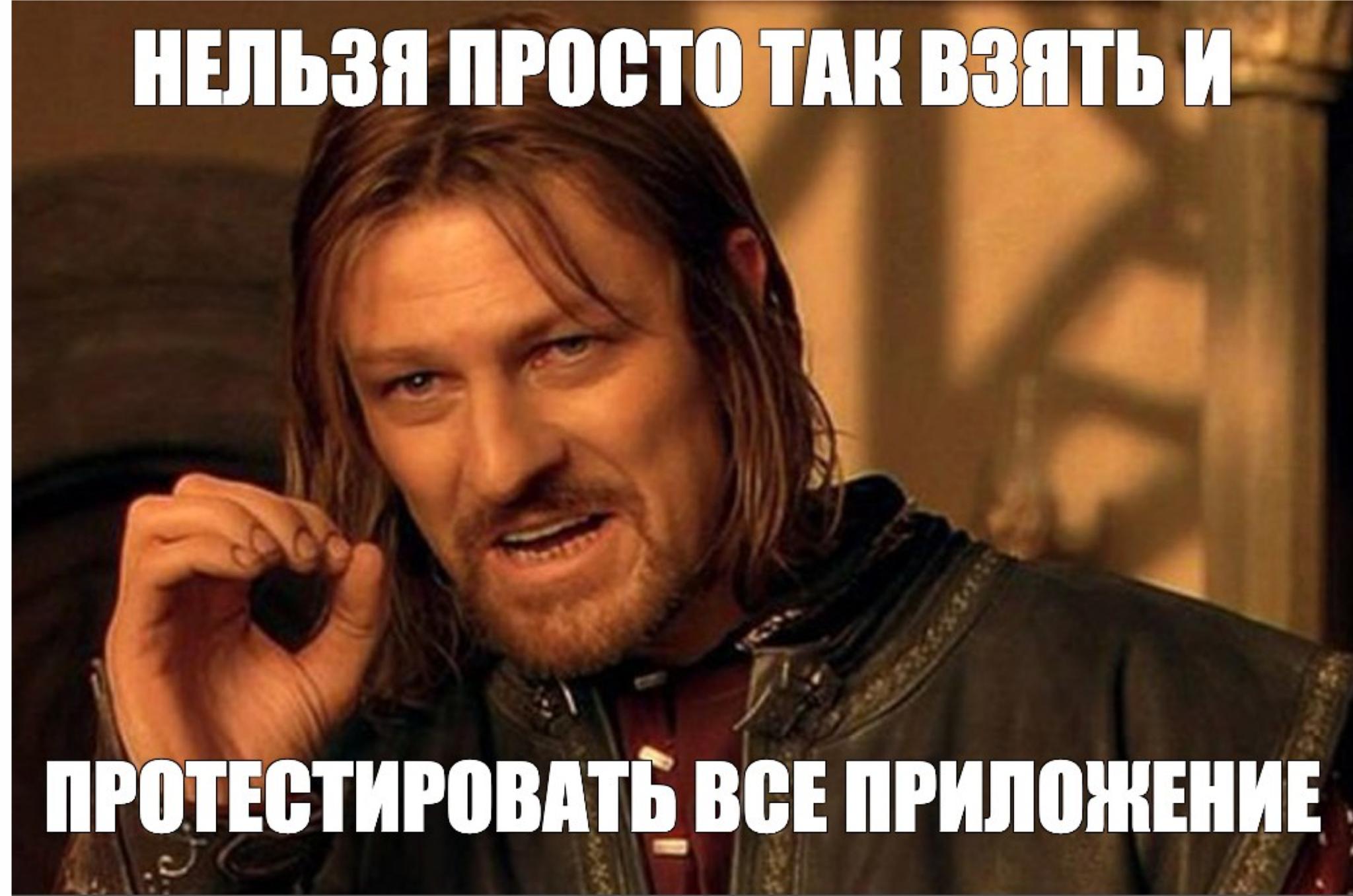


Сложности тестирования

- Тестов - много, времени – мало

Решение: необходимо создать ограниченный набор тестов, который позволит протестировать приложение, не утратив при этом качество тестирования

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ И



ПРОТЕСТИРОВАТЬ ВСЕ ПРИЛОЖЕНИЕ

Техники тест-дизайна

- Классы эквивалентности
- Граничные условия
- Таблица принятия решений
- Граф потока управления
- Граф потока данных
- Pair-wise

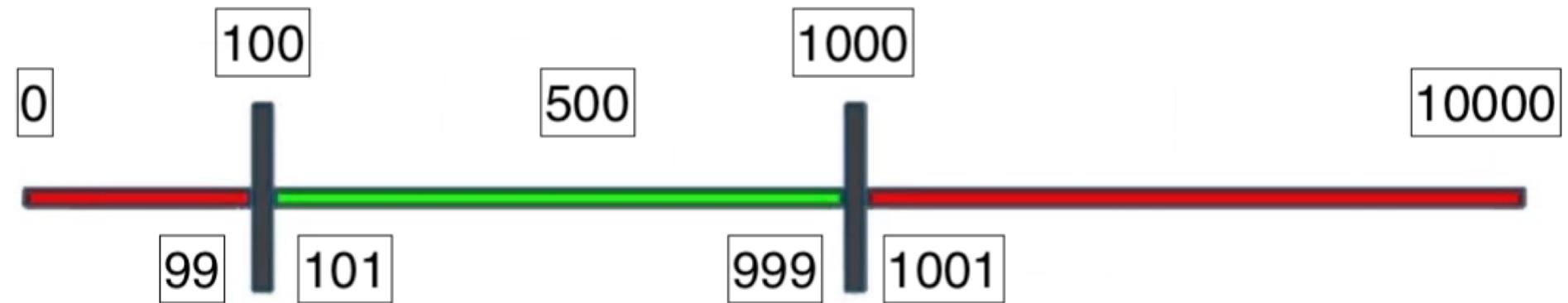
Классы эквивалентности

- Классы эквивалентности – наборы входных параметров, данные которых система считает эквивалентными.
- Тесты эквивалентны тогда, когда:
 - они тестируют одно и то же
 - если один из них выявляет ошибку, то и остальные ее выявят
 - если один из них не выявляет ошибку, то и остальные ее не выявят



Границные условия

- Границные условия (значения) – это ситуации перед границей, на границе и за границей входных условий.



Pair-wise

- Pair-wise - формирование таких наборов данных, в которых каждое тестируемое значение каждого из проверяемых параметров хотя бы единожды сочетается с каждым тестируемым значением всех остальных проверяемых параметров

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	1	1	1	1	1
3	1	1	0	1	0	1	0	1	0	1
4	1	0	1	0	1	0	1	0	1	0

Тестовые наборы

- Smoke
- Regression
- Acceptance

[Edit](#)[Comment](#)[Assign](#)[To Do](#)[In Progress](#)[Done](#)

...

Отчеты

Type:

[Bug](#)

Status:

[TO DO](#) (View workflow)

Assignee:

Unassigned

[Assign to me](#)

Priority:

[Medium](#)

Resolution:

Unresolved

Labels:

None

Reporter:

Usersnap Developer

Environment:

Chrome Version 70.0.3538.77 (Official Build) (64-bit) -

Votes:

Screen Resolution: 1920 x 947

Sprint:

Watchers:

[1 Stop watching this issue](#)**Description**

Created:

14 minutes ago

Components:

Updated:

Just now

on homepage

Description:

Development

Outdated header image on www.usersnap.com

Steps to reproduce:

Create branch

- 1) Go to usersnap.com
- 2) check the header image

Agile

Javascript console errors:

View on Board

no error messages

Result:*expected Result:*

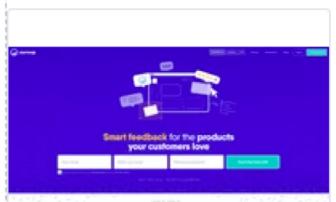
header_image_2.jpeg

actual Result:

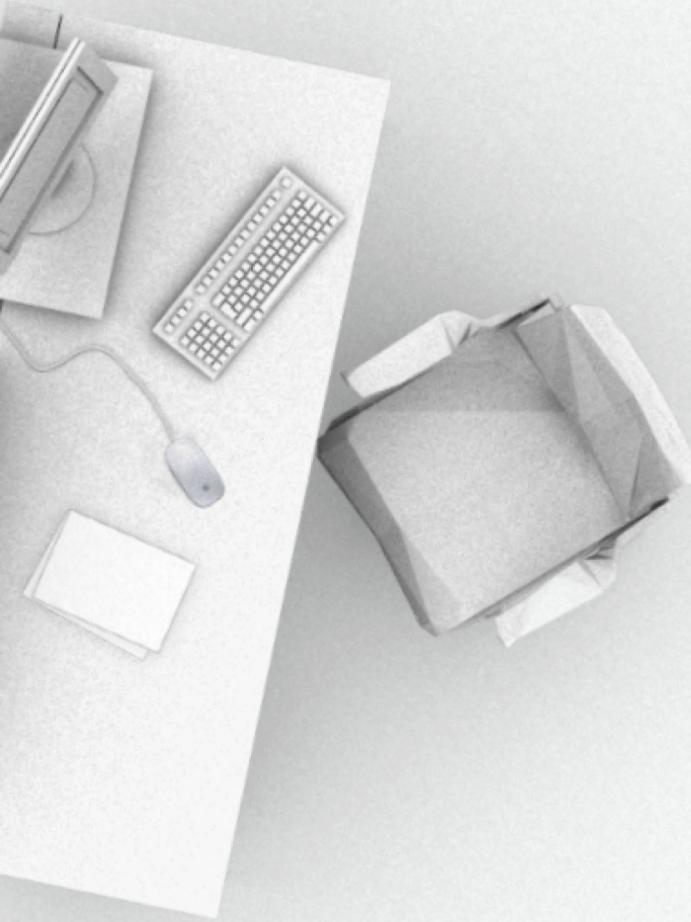
header_image_1.jpeg

Attachments

...

 Drop files to attach, or [browse](#).

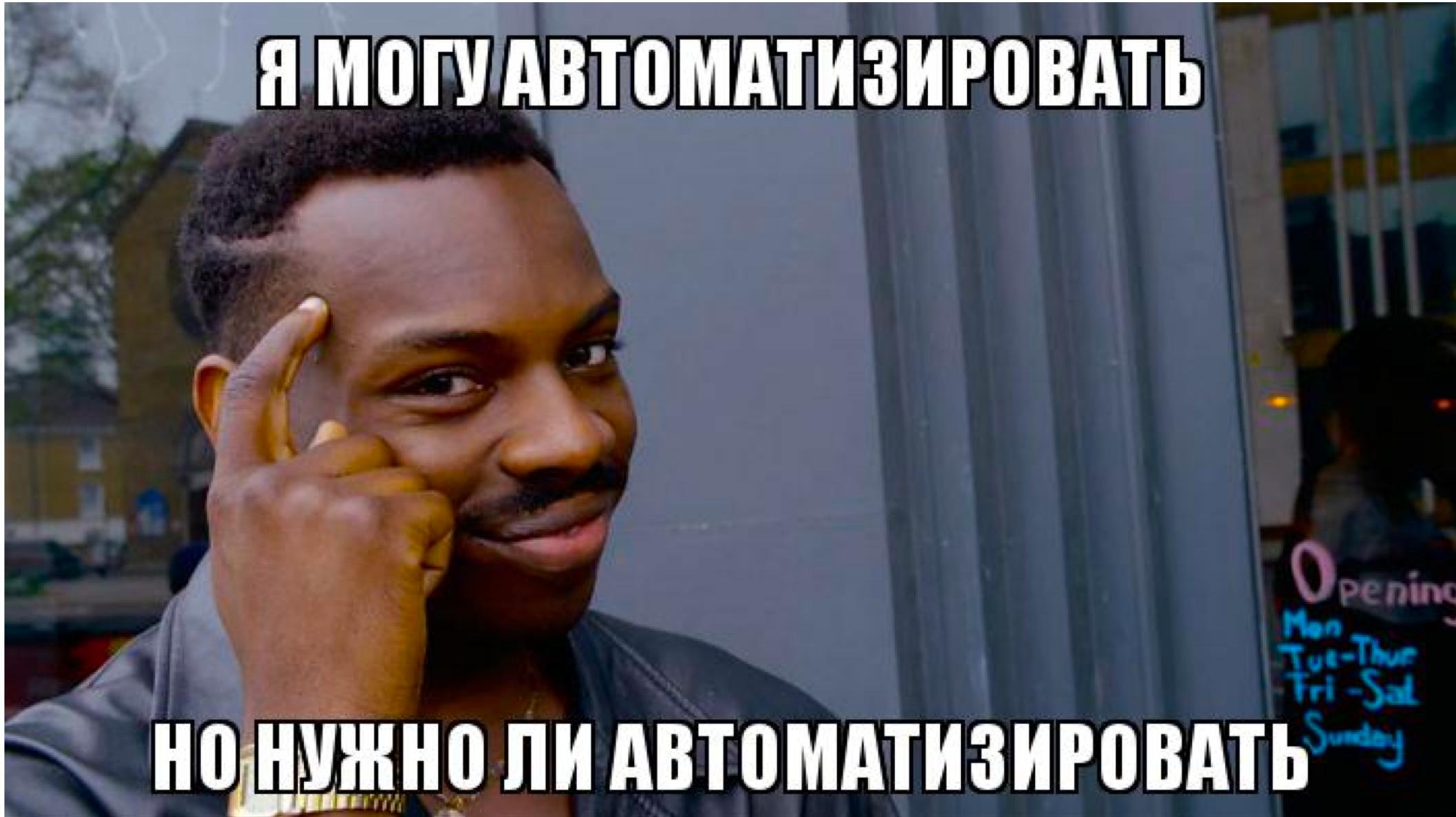
#022



Автоматизированное тестирование

Что такое автоматизированное тестирование?

Это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически



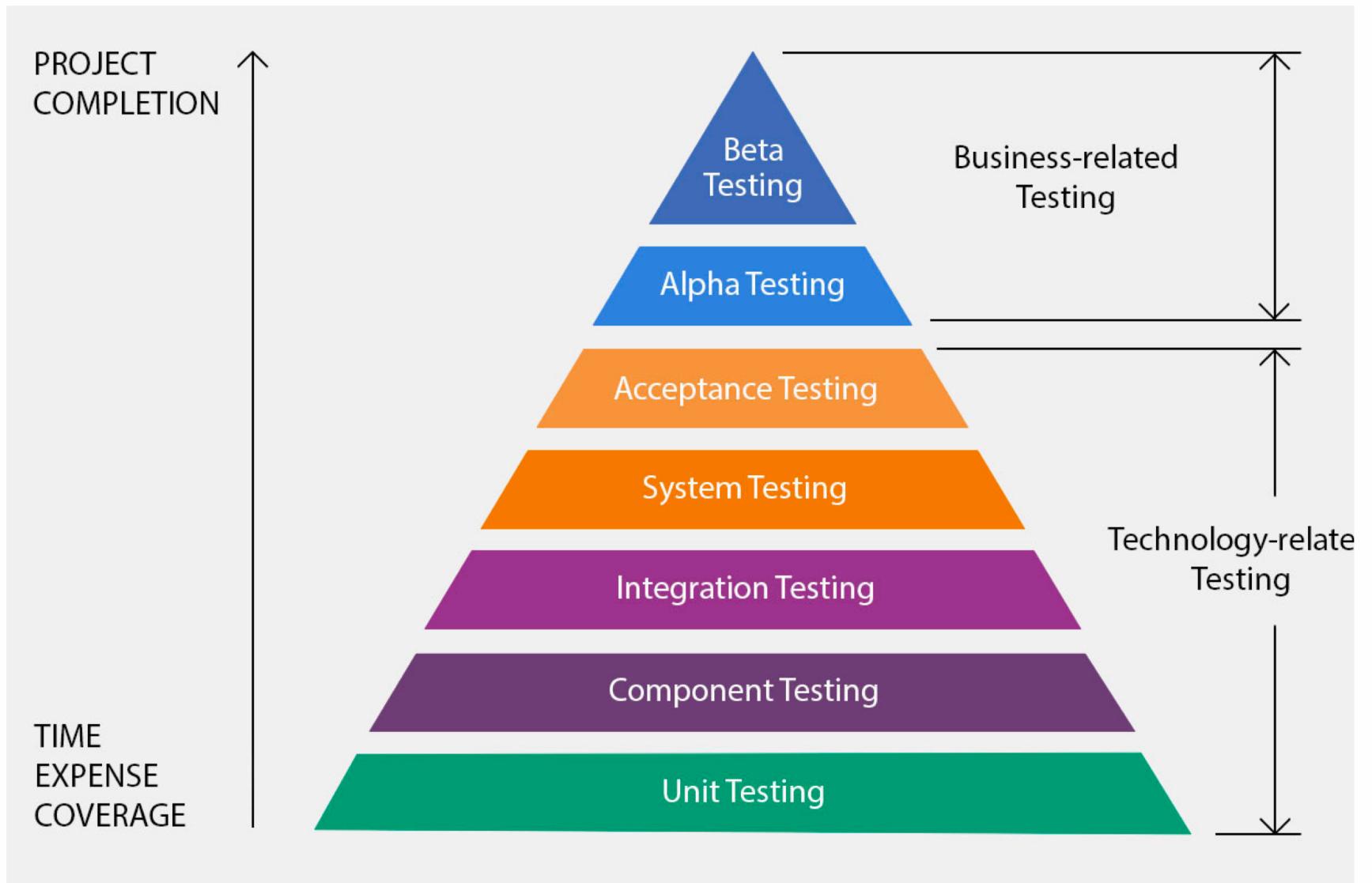
Преимущества автоматизированного тестирования

- Быстрое выполнение
- Повторяемость
- Меньше затрат на поддержку
- Отчеты

Цели автоматизации тестирования

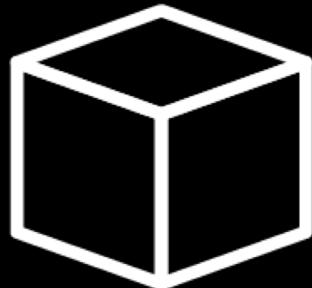
- Повышение производительности труда
- Повышение надежности
- Повышение скорости разработки

Уровни тестирования



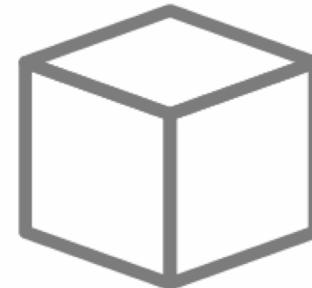
Методы тестирования

BLACK BOX



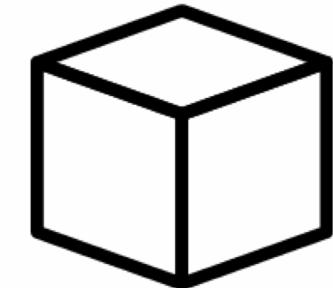
ZERO KNOWLEDGE

GRAY BOX



SOME KNOWLEDGE

WHITE BOX



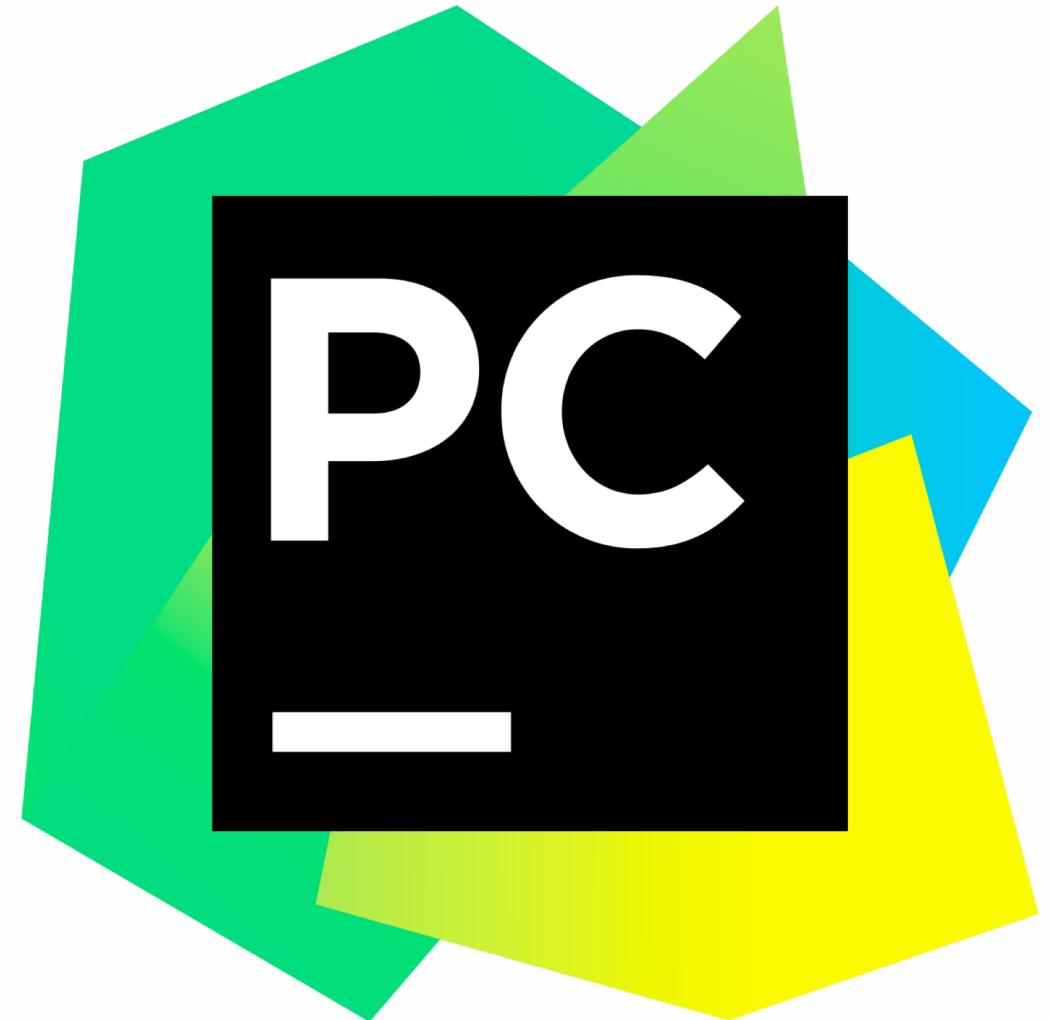
FULL KNOWLEDGE

Инструменты



Инструменты

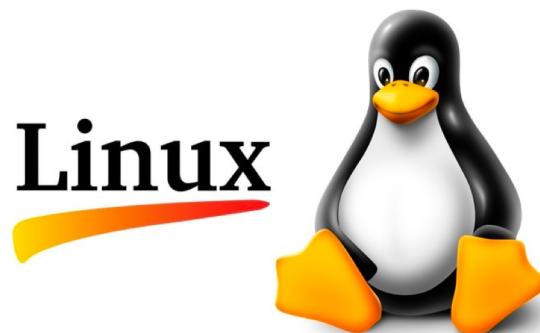
#031



Виртуальное окружение

Виртуальное окружение - это изолированное окружение среды (в нашем случае это окружение Python), которое позволяет нам использовать определенные версии приложений.

Использование различных ОС при автоматизации тестирования



MacTM OS

НЕХОЧУ НИЧЕГО НАСТРАИВАТЬ



ВЫБИРАЮ ПИСАТЬ АВТОТЕСТЫ

#034



py.test



Структура проекта

```
ROOT
├── fixtures
│   └── fixtures.py
├── tests
│   ├── test_1.py
│   └── test_2.py
├── tools
│   └── helpers.py
└── conftest.py
pytest.ini
```

Основы pytest

1

Полезная литература в помощь

<https://docs.pytest.org/en/stable/contents.html>

<https://docs.python.org/3/>

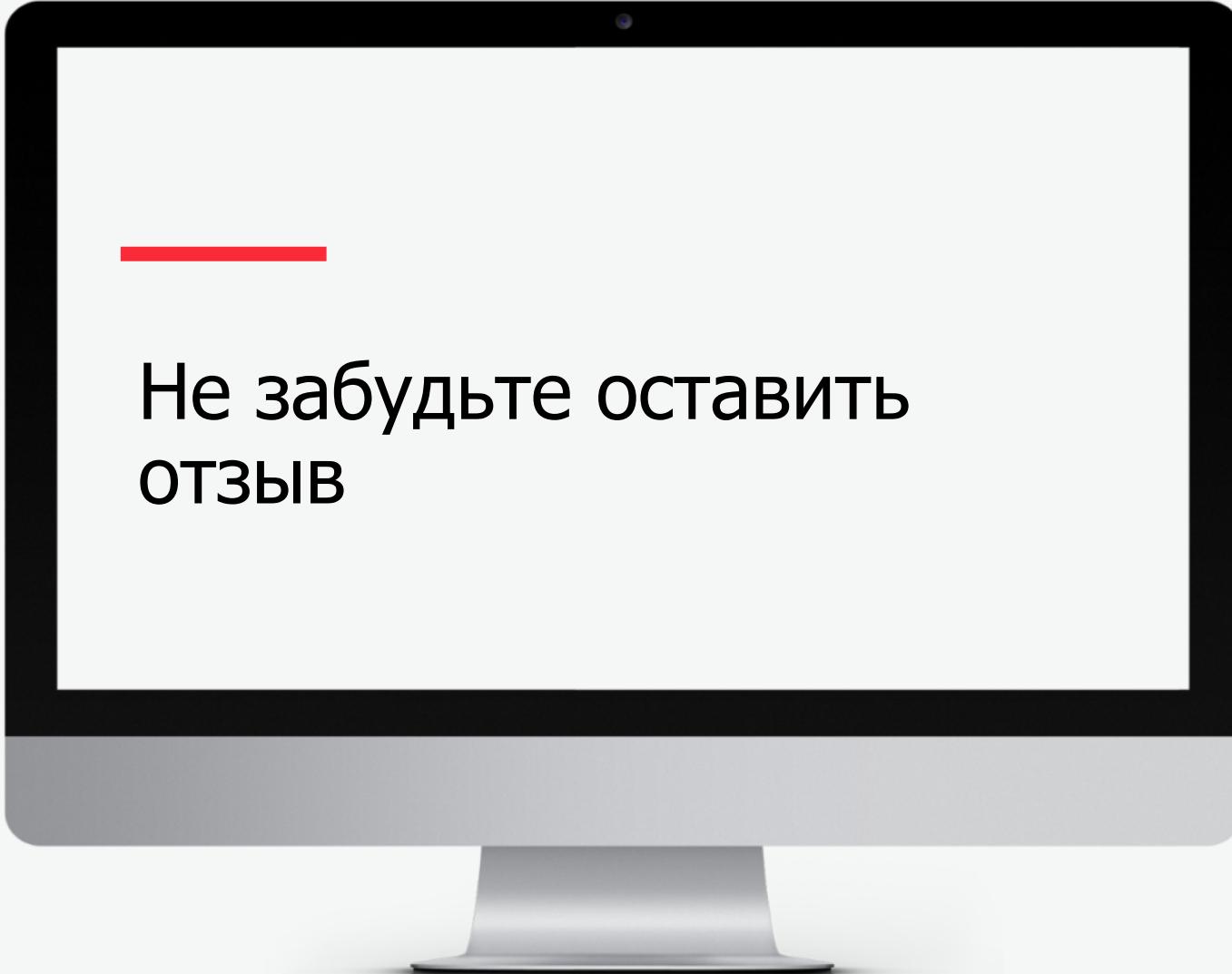
4

30.09.2020

Балла
за задание

Срок
сдачи

- Написать по 5 автотестов с учетом техник тест-дизайна на каждую из структур данных в Python:
 1. List
 2. Set
 3. Dictionary
 4. String
 5. Int
- Как минимум один тест на каждую структуру должен быть параметризован.
- Как минимум один тест на каждую структуру должен быть в классе
- Проверить код линтером (pylint, flake8, pycodestyle)



—

Не забудьте оставить
отзыв



**СПАСИБО
ЗА ВНИМАНИЕ**

