



Санкт-Петербургский государственный университет
Кафедра системного программирования

Тестирование / Rust

Дмитриевцев Алексей Сергеевич

* Некоторые моменты вдохновлены
и/или позаимствованы из презентации
В.А Карасева

Санкт-Петербург
2024

Тестирование?

Какие проекты бывают?

- Без покрытия тестами
- Есть какие-то тесты, которые никто не поддерживает
- Хорошее покрытие тестами



ISO/IEC TR 19759:2005

Тестирование ПО — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

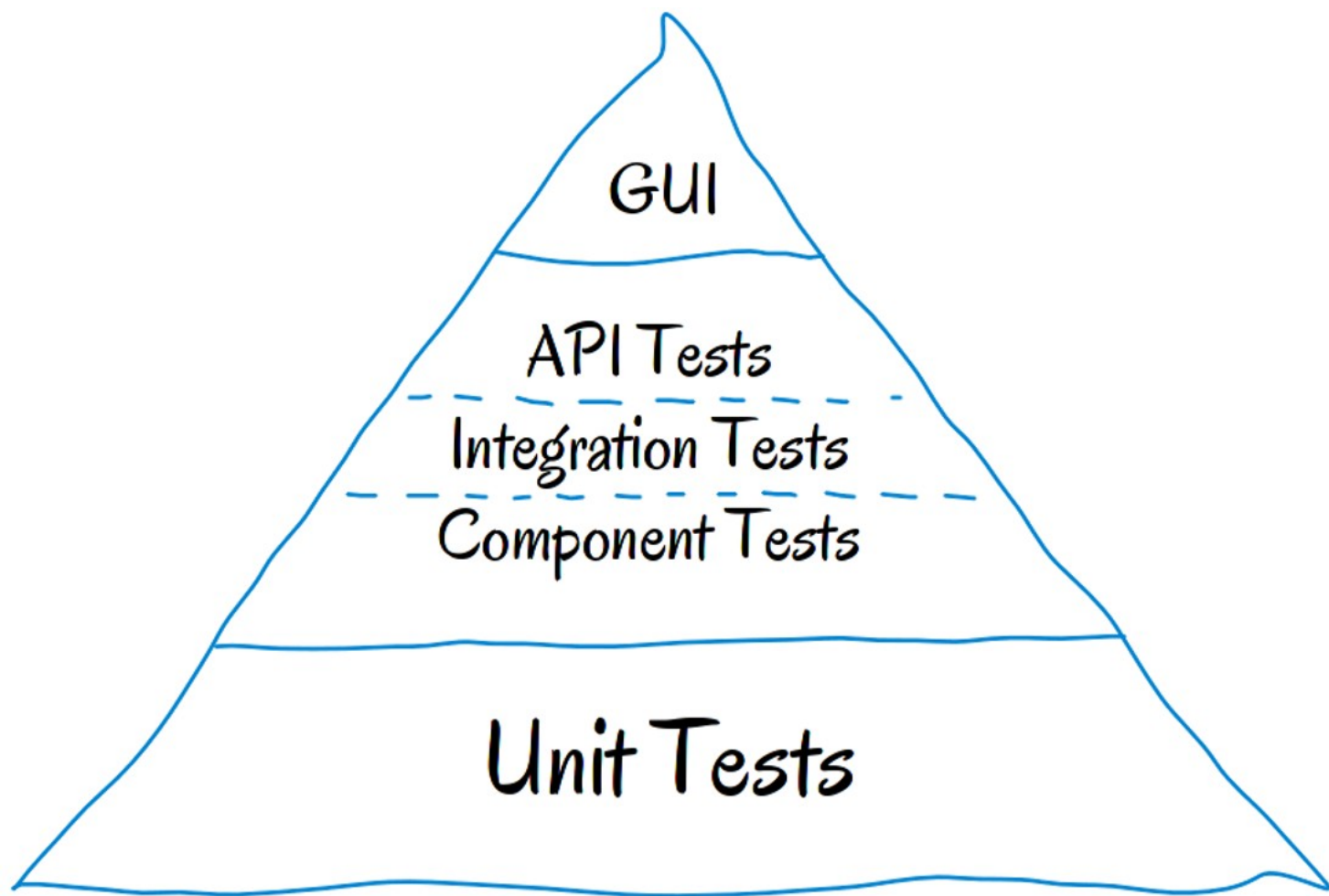


Какое тестирование бывает?

- Тестирование функциональности
- Тестирование производительности



Пирамида тестирования



Картинка с
<https://habr.com/ru/articles/568360/>

Unit-тесты

- Проверяем отдельные unit'ы
- Уверенность
- Описательность («Good tests act as documentation for the code that they are testing» — Gerard Meszaros)

Как стоит называть тесты?

```
fn is_leap_year(year: i32) -> bool {  
    return  
        year % 4 == 0 &&  
        year % 100 != 0 ||  
        year % 400 == 0;  
}
```

- `fn test()`
- `fn test_is_leap()`
- `fn test_is_leap_year_is_correct()`

Как стоит называть тесты?

```
#[test]
fn test_leap_years() {
    assert!(is_leap_year(2040));
    assert!(is_leap_year(2000));
}
```

```
#[test]
fn test_not_leap_years() {
    assert!(is_leap_year(2421));
    assert!(is_leap_year(1900));
}
```


Как стоит называть тесты?

```
#[test]
fn test_leap_years() {
    assert!(is_leap_year(2040));
    assert!(is_leap_year(2000));
}
```

```
#[test]
fn test_not_leap_years() {
    assert!(is_leap_year(2421));
    assert!(is_leap_year(1900));
}
```

Как стоит называть тесты?

```
#[test]
fn test_is_2040_leap_year() {
    assert!(is_leap_year(2040));
}

#[test]
fn test_is_2000_leap_year() {
    assert!(is_leap_year(2000));
}

#[test]
fn test_is_1900_not_a_leap_year() {
    assert!(is_leap_year(1900));
}

#[test]
fn test_is_2421_not_a_leap_year() {
    assert!(is_leap_year(2421));
}
```

Как стоит называть тесты?

```
#[test]
fn year_not_divisible_by_4_is_not_leap() {
    assert!(is_leap_year(1941));
}

#[test]
fn year_divisible_by_4_but_not_by_100_is_leap() {
    assert!(is_leap_year(2040));
}

#[test]
fn year_divisible_by_100_but_not_by_400_is_not_leap() {
    assert!(is_leap_year(1900));
}

#[test]
fn year_divisible_by_400_is_leap() {
    assert!(is_leap_year(2000));
}
```

Тесты в Rust

Флаги:

`--test-threads=4`

`--show-output`

`--ignored`

Папка `tests` (показать)

Параметризованные тесты

```
#[rtest]
```

```
#[case(0, 0)]
```

```
#[case(1, 1)]
```

```
#[case(2, 1)]
```

```
#[case(3, 2)]
```

```
#[case(4, 3)]
```

```
fn fibonacci_test(#[case] input: u32, #[case] expected: u32) {  
    assert_eq!(expected, fibonacci(input))  
}
```

Другие виды тестирования

- Fuzzing
- UI
- Интеграционные
- Системные
- Многое другое

Здесь несколько слов про CI на примерах если успеем

