

Python2018

compscicenter.ru

aleksey.kladov@gmail.com

Лекция 11: Тестирование

Терминологическая Путаница

unit-testing

тестирование изолированных компонент /
автоматизированное тестирование

integration-testing

тестирование взаимодействия компонент / не unit
тестирование



Свойства Тестов

- Простота написания (!)
- Скорость работы (unit vs. integration)
- Стабильность при изменении кода (code churn)
- Стабильность при изменении внешних компонент
- Надёжность (тесты проходят, но работает ли программа?)

ЛИЧНЫЙ ОПЫТ

- Тестирование это сложно и не понятно
- Лучшая метрика — покрытие user-visible фич
- Лучшие тесты — быстрые интеграционные тесты без внешних зависимостей
- Нужны медленные flaky тесты для зависимостей
- Data-driven тесты эффективны
- Не умею тестировать асинхронный код (в том числе UI)

unittest

```
import unittest

class TestStringMethods(unittest.TestCase):

    def test_upper(self):
        self.assertEqual('foo'.upper(), 'FOO')

if __name__ == '__main__':
    unittest.main()
```

- API из Java
- Отличный пример неудачного использования OOP