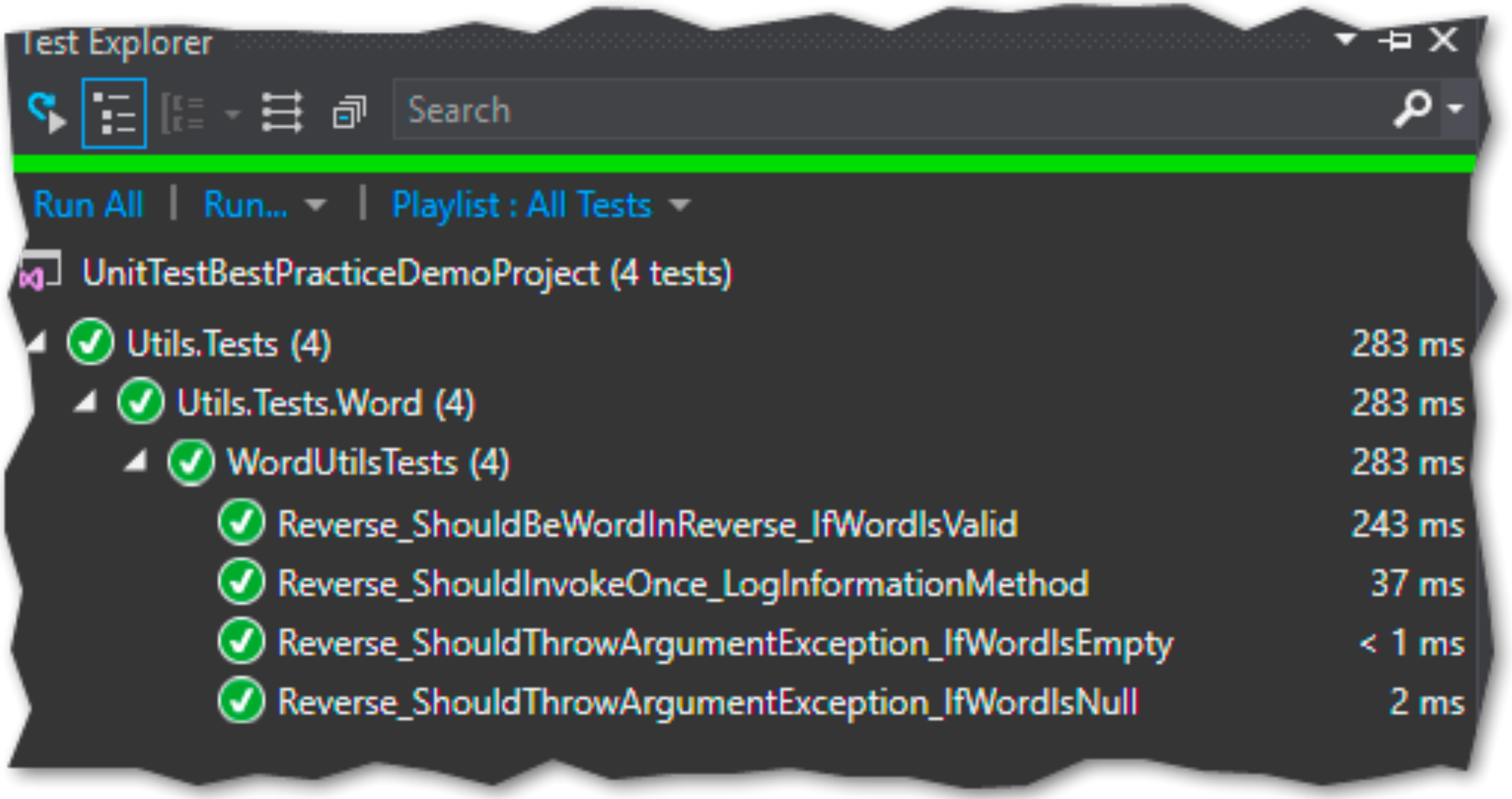
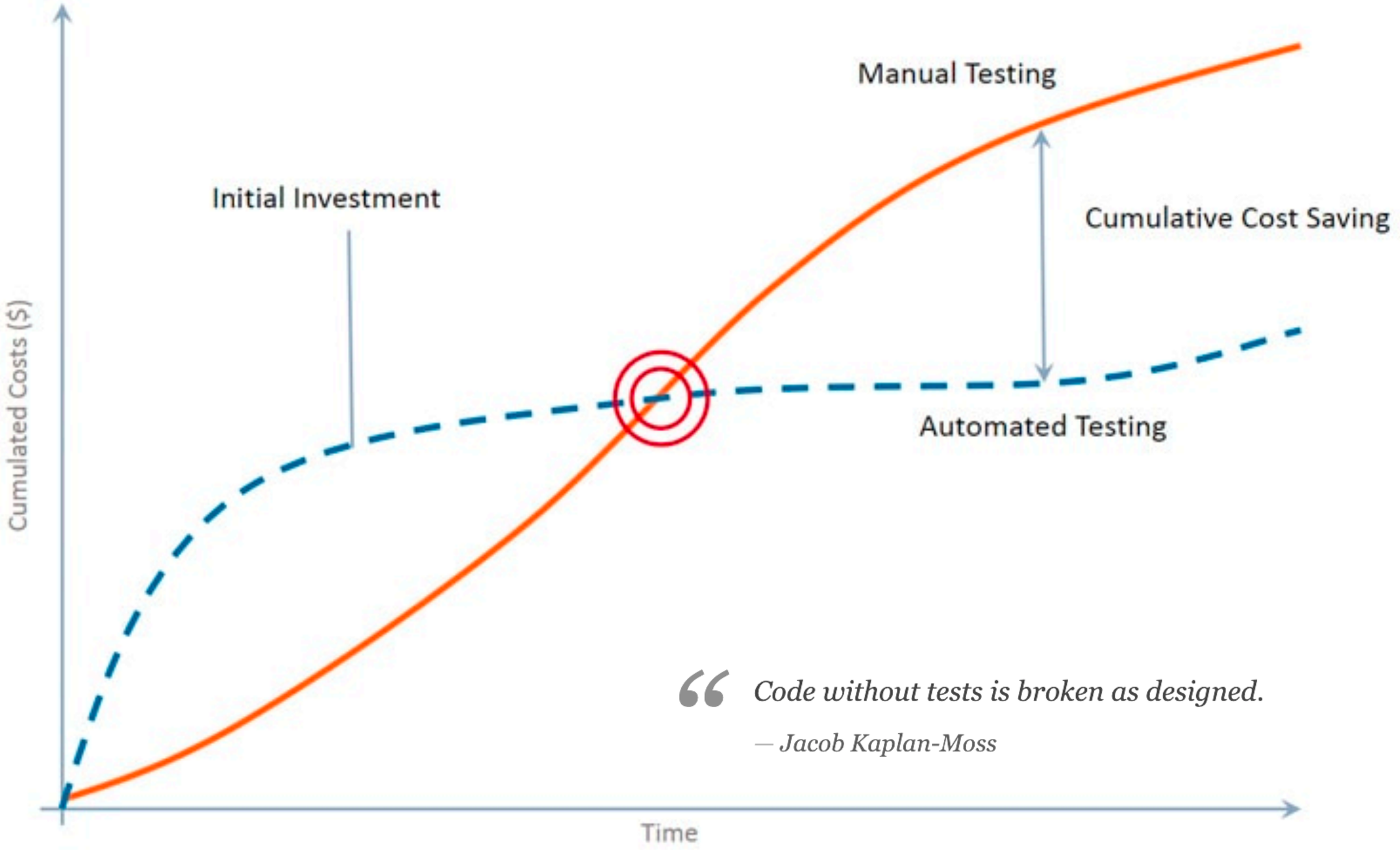
Unit & Integration Tests

Django Development II

Azimjon Pulatov | PDP Online

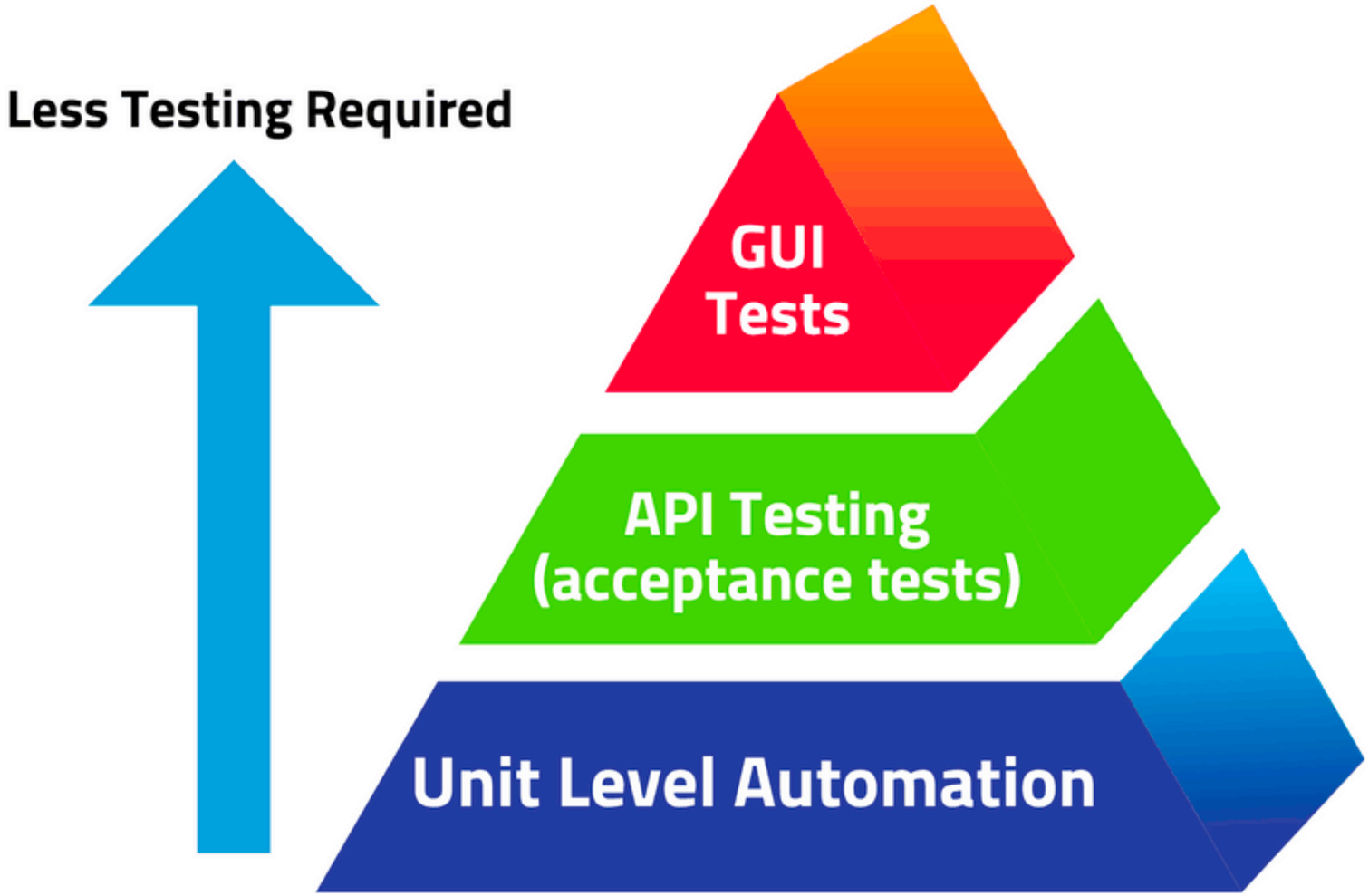
What is Automated Tests?


Test Explorer
Search
Run All | Run... | Playlist : All Tests
UnitTestBestPracticeDemoProject (4 tests)
Utils.Tests (4) 283 ms
Utils.Tests.Word (4) 283 ms
WordUtilsTests (4) 283 ms
Reverse_ShouldBeWordInReverse_IfWordIsValid 243 ms
Reverse_ShouldInvokeOnce_LogInformationMethod 37 ms
Reverse_ShouldThrowArgumentException_IfWordIsEmpty < 1 ms
Reverse_ShouldThrowArgumentException_IfWordIsNull 2 ms



> *Code without tests is broken as designed.*
>
> — *Jacob Kaplan-Moss*

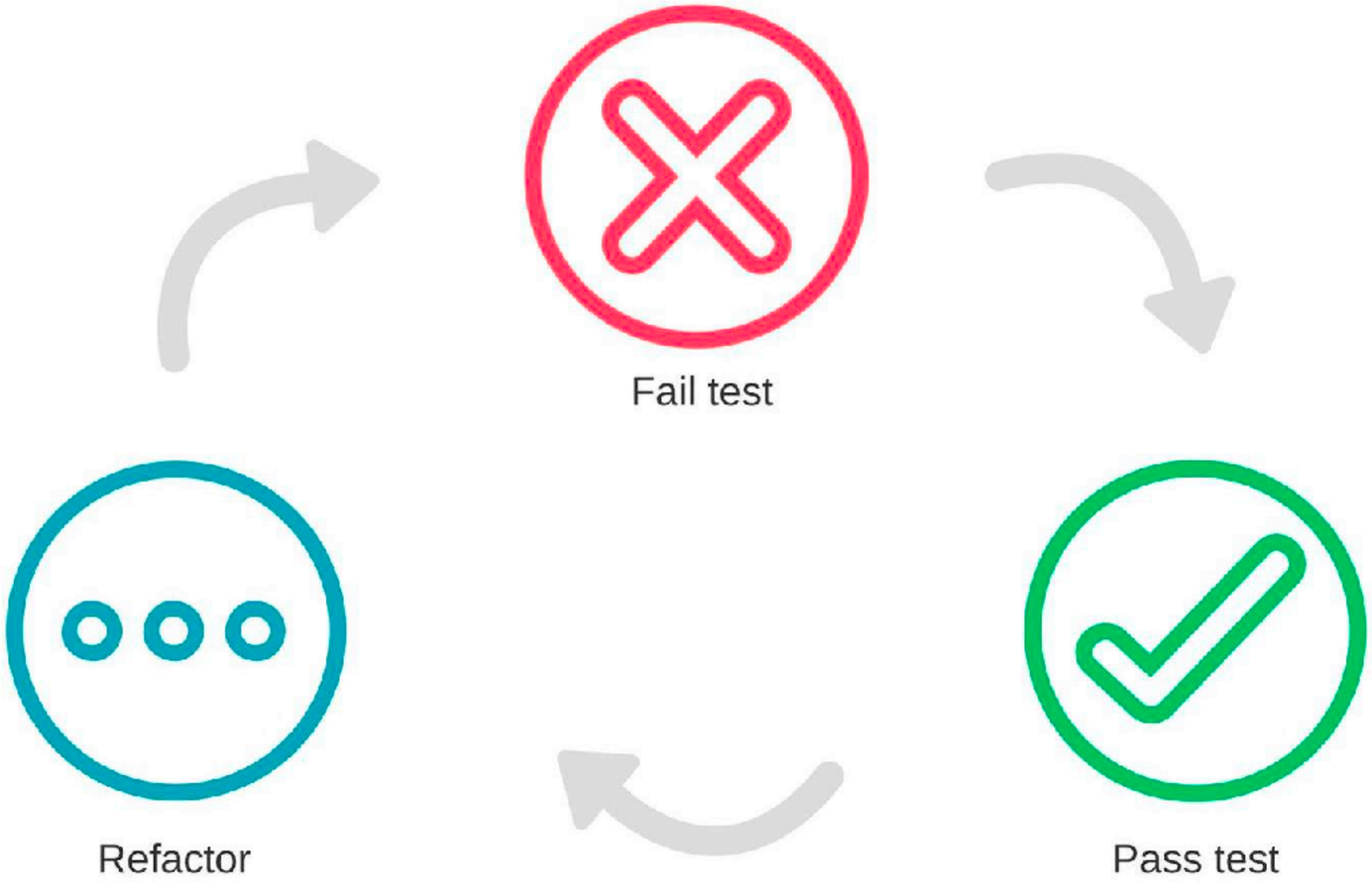
Test Automation Pyramid



Unit Testing

Good Unit Test Characteristics

Refactoring Code with Tests



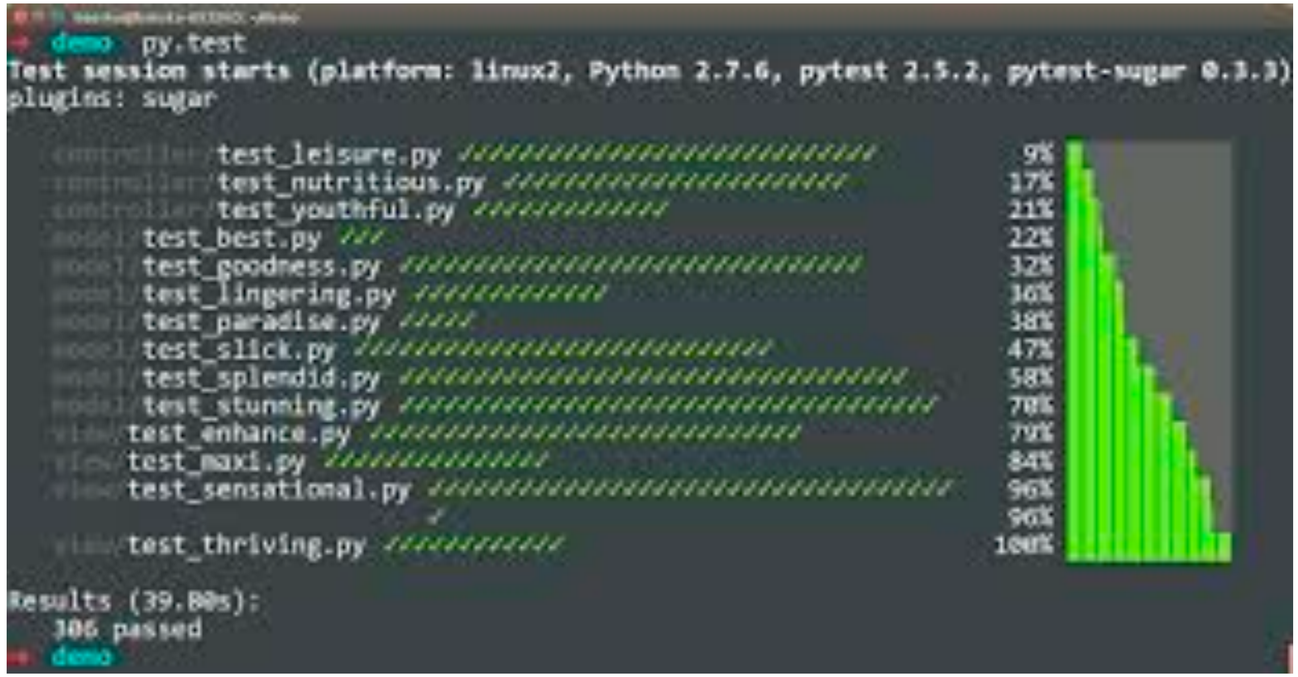
Testing with Django

```
from django.test import TestCase
from myapp.models import Animal

class AnimalTestCase(TestCase):
    def setUp(self):
        Animal.objects.create(name="lion", sound="roar")
        Animal.objects.create(name="cat", sound="meow")

    def test_animals_can_speak(self):
        """Animals that can speak are correctly identified"""
        lion = Animal.objects.get(name="lion")
        cat = Animal.objects.get(name="cat")
        self.assertEqual(lion.speak(), 'The lion says "roar"')
        self.assertEqual(cat.speak(), 'The cat says "meow"')
```

pytest testing framework



pytest tests

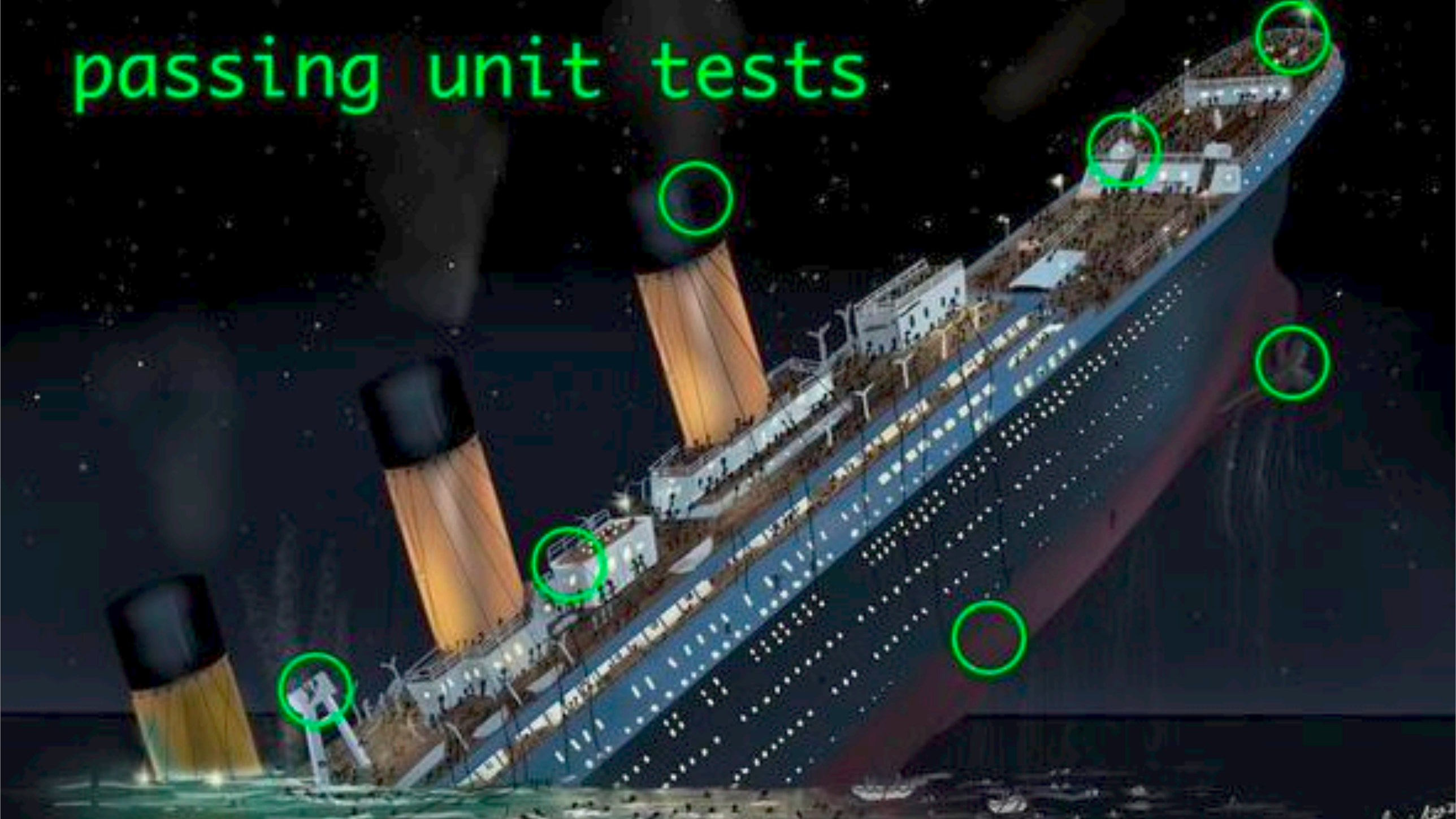
test_port2_pytest.py

```
def test_empty():
    p = Portfolio()
    assert p.cost() == 0.0

def test_buy_one_stock():
    p = Portfolio()
    p.buy("IBM", 100, 176.48)
    assert p.cost() == 17648.0

def test_buy_two_stocks():
    p = Portfolio()
    p.buy("IBM", 100, 176.48)
    p.buy("HPQ", 100, 36.15)
    assert p.cost() == 21263.0
```

```
$ pytest -q test_port2_pytest.py
...                                                                  [100%]
3 passed in 0.01s
```


passing unit tests