**Тестирование кода**

Если код не тестировать, то стоимость разработки сильно возрастает (а не падает!)

Потому что:

* обнаруживают проблему пользователи. Они ее **кое-как** описывают. (У меня ничего не работает). По их описанию понять как повторить проблему трудно - затраты на службу поддержки и время программиста, пока он разберется по этому описанию. Проблема может быть, но программист ее может не найти по *такому*описанию и не исправить.
* пока программист видит свежий код, он его помнит. Через некоторое время нужны дополнительные ресурсы, чтобы понять свой же код. Если фиксит баг другой программист, то затраты еще больше.
* При внесении изменений (новые фичи, фиксы других багов) могут поломать существующий рабочий код. Если нет проверки кода, то вместо 1 старого бага получим несколько новых.   
  \*Очень важно быстро перепроверить старый работающий код\*.

**Термины тестирования**

unittest ([тестирование компонент](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%3Cbr%20/%3E%E2%80%A2D1%8C%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) - тестирование отдельных функций или модулей.

Проблемы:

* Тестового кода часто больше, чем основного кода;
* необходимы объекты-заглушки;
* не отлавливаются проблемы интеграции модулей.

**Test-driven Development (TDD)**

* Идея - сначала напишем тесты (как должно работать), а потом допишем под них код (и сразу будем убеждаться, что работает).

**Библиотеки unit-тестирования для питона**

* [unittest](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonUnittest) - встроена в питон
  + [tutorial](https://www.python-course.eu/python3_tests.php) (Eng)
* doctest - встроена в питон
* noise - может запускать написанные unittest и doctest, есть свои методы
* [pytest](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonPytest) - его будем изучать после изучения декораторов и понимания что нужно для развернутого тестирования
  + <https://habrahabr.ru/post/269759/>

**Что читать**

* <https://www.python-course.eu/python3_tests.php> - doctest, unittest (Eng)
* <https://habrahabr.ru/post/121162/> - обзор пакетов тестирования питона для начинающих
* <http://devpractice.ru/python-unittest-lessons/> - учебник по unittest на русском
* <https://wiki.python.org/moin/PythonTestingToolsTaxonomy> - сводная таблица всего,
* что есть в питоне для разного тестирования.
* <https://habrahabr.ru/post/269759/>
* <http://pythontesting.net/framework/pytest/pytest-introduction>

-- [TatyanaDerbysheva](http://acm.mipt.ru/twiki/bin/view/Main/TatyanaDerbysheva) - 12 Nov 2017