Модульное тестирование

Модульное тестирование, или **юнит-тестирование** — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Модульное тестирование позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже протестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Этот вид тестирования выполняется самими программистами. В процессе написания модульных тестов разработчик:

- делает аудит логики основного кода
- автоматически проверяет модульность своего кода

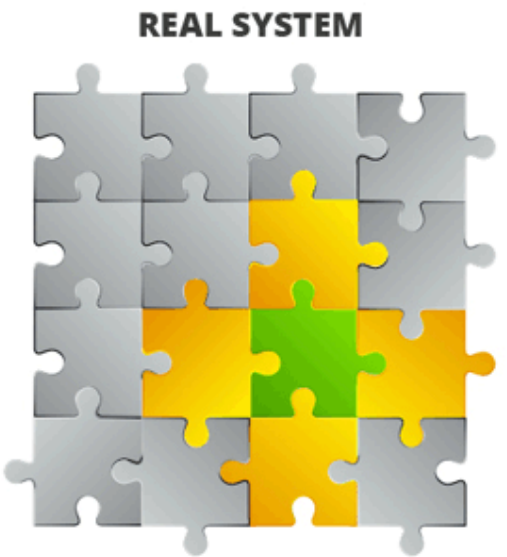
Модульное тестирование практически невозможно в условиях некачественного исходного кода. Таким образом, оно подталкивает программистов к использованию лучших практик программирования.

QA: Unit тесты – это тестирование отдельного модуля АС.

DEV: Я пишу код, а мой падаван пишет к нему юнит-тесты

DEV: Написание тестов отнимает время.

DEV: Это работает, только если писать проект с нуля.



Источники

"Goto Fail, Heartbleed, and Unit Testing Culture" by Martin Fowler.
http://ru.wikipedia.org/wiki/Модульное_тестирование
http://www.protesting.ru/testing/levels/component.html
http://citforum.ru/SE/testing/unit_testing/
http://blog.openquality.ru/unit-tests-why

Модульное тестирование. Ограничение

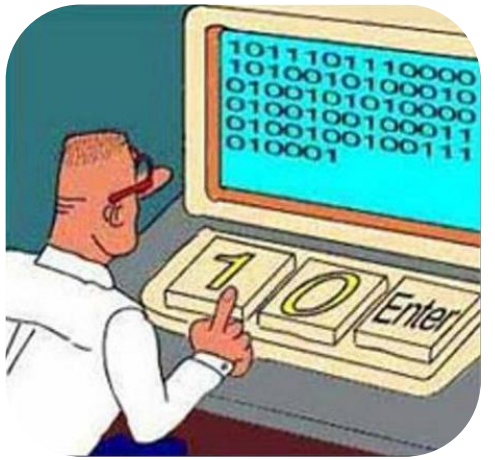
Не нужно писать тесты, если

- Вы пишите тесты на сложный и сильно связный код (об этом вам может подсказать инструмент статического анализа кода). Сначала нужно сделать рефакторинг кода
- Вы делаете прототип для демонстрации концепции. После демонстрации прототипа вы не планируете использовать созданный программный код
- Вы всегда пишете код без ошибок, обладаете идеальной памятью и даром предвидения. Ваш код настолько крут, что изменяет себя сам, вслед за требованиями заказчика. Иногда код объясняет заказчику, что его требования не нужно реализовывать

http://eax.me/unit-testing/

https://habrahabr.ru/post/169381/

Наличие юнит тестов дополняет, а не отменят другие виды тестирования.



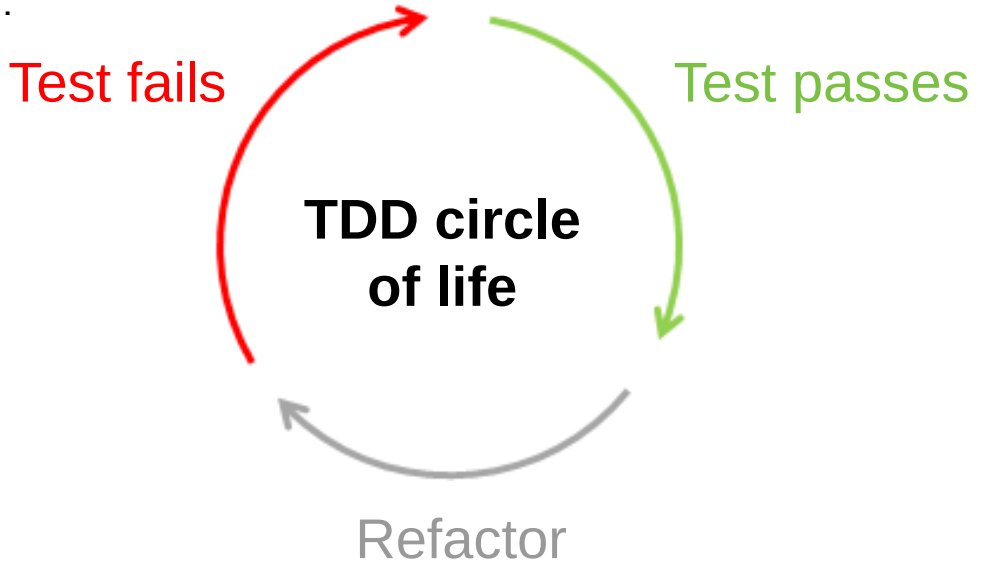
Test-driven development (TDD, Разработка через тестирование)

Один из наиболее эффективных подходов к модульному тестированию - это **подготовка автоматизированных тестов** до начала основного кодирования (разработки) программного обеспечения (**test first approach**). .

Это называется разработка от тестирования (**test-driven development**) или подход тестирования вначале.

При этом подходе создаются и интегрируются небольшие куски кода,
для которых запускаются тесты, написанные до начала кодирования.

Разработка ведётся до тех пор, пока все тесты не будут успешно пройдены.



[2] "Monogomous TDD," 8th Light.

[4] "How Do We Learn?" Ron Jeffries.

[5] "Gold Plating (Software Engineering)," Wikipedia.

[6] "Notes on Structured Programming," Eindhoven University of Technology.

[7] "Top 5 TDD Mistakes," Telerik by Progress.

[8] K. Beck. "Extreme Programming Explained: Embrace Change." Addison-Wesley. 2000.

[9] "Goto Fail, Heartbleed, and Unit Testing Culture," Martin Fowler.

[10] "Refactoring: Improving the Design of Existing Code," Martin Fowler.