

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Национальный аэрокосмический университет
им. Н.Е. Жуковского "Харьковский авиационный институт"
Кафедра компьютерных систем и сетей

Доклад на тему:
"Test-Driven Development
(разработка через тестирование)"

Выполнил: Черненко М.Ю.
Студент группы 545

Харьков 2017

Что же такое TDD?

- ▶ техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам

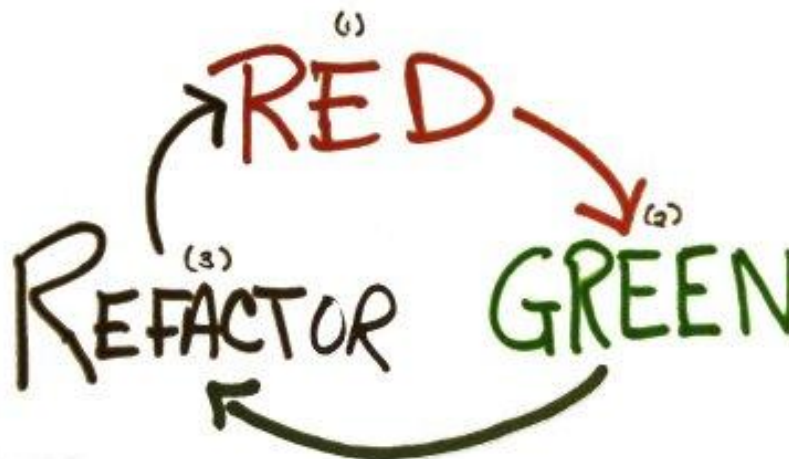
TDD

**ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT**

Принцип работы

Весь процесс разработки глобально делится на 3 этапа, так называемый RED-GREEN-REFACTOR lifecycle:

- ▶ RED - написание "падающего" теста - теста, который в данный момент не проходим
- ▶ GREEN - написание кода, который позволит пройти этот тест
- ▶ REFACTOR - рефакторинг, реструктуризация и реорганизация кода, оптимизация алгоритмов его работы. Также данный шаг касается и написанных тестов - улучшение существующих тестов для их более качественного прохода по коду



Плюсы TDD подхода

Качество тестов:

- 100% покрытие тестами
- Модульные тесты
- Изменение без страха



Качество кода:

- Модульный, гибкий и расширяемый код
- Всегда работающий код
- Меньшее время отладки
- Поощрение рефакторинга

Минусы TDD подхода



- Высокий порог вхождения
- Ошибочный тест = ошибочный код
- Поддержка большого количества тестов

Ошибки при следовании TDD

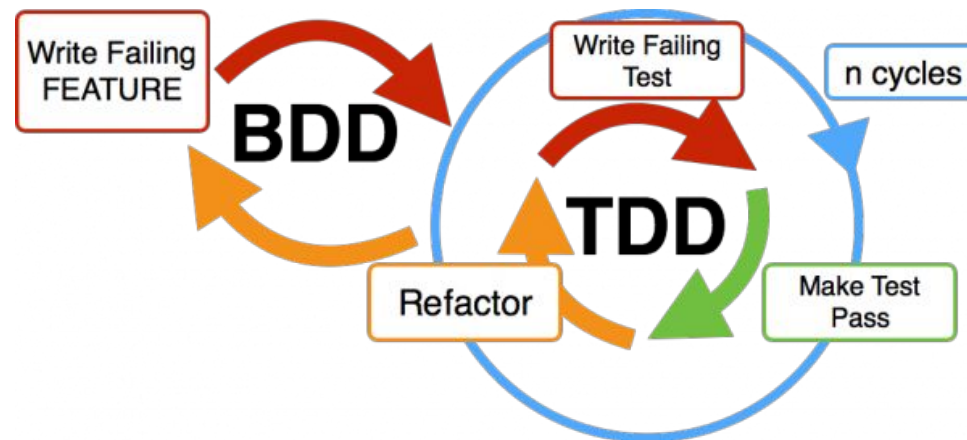
- ▶ Пропуск "красной зоны" - стадии неработающего теста
- ▶ Акцентирование внимания на "покрытие" кода
- ▶ Излишняя увлечённость рефакторингом
- ▶ Игнорирование рефакторинга
- ▶ Игнорирование слишком простых или слишком сложных сценариев
- ▶ Отсутствие "зелёного" теста больше 10 минут - хорошей практикой считается принцип одновременной работы только над одним тестом



BDD - Behavior-driven development

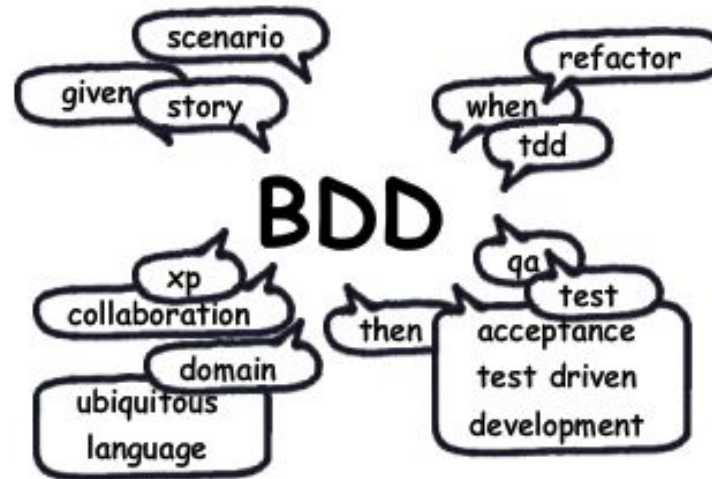
Одно из наиболее популярных ответвлений TDD, которое полагается на такие основные принципы как:

- ▶ Связь кода с требованиями
- ▶ Фокус не на тестах, а на поведении
- ▶ Запись требований с помощью обычных фраз



Принцип работы BDD

- ▶ Описание поведения заказчиком (исходное состояние/событие/проверка поведения)
- ▶ Реализация функционала
- ▶ Проверка соответствия требованиями (прошёл/не прошёл)



Выводы

- ▶ Следование методологии разработки через тестирования позволит получить на выходе работающий, полностью(насколько это возможно) оттестированный продукт, поддержка которого в будущем не составит больших проблем. Но с другой стороны TDD методология значительно увеличит порог вхождения в проект и усложнит некоторые моменты разработки. К тому же, игнорирование и нарушение основных принципов TDD принесёт лишь боль и страдания и приведёт к PDD - panic driven development'у