LUXOFT TRAINING

Разработка через тестирование
Как преподнести команде

Test Driven Development

Ivan Dyachenko <IDyachenko@luxoft.com>

Module 13

Содержание

LUXOFT Training Center

Слишком простой код

LUXOFT Training Center

- Как правило, так кажется только в момент его написания
- Через полгода (а может и через пару недель) связи этого кода с другими могут показаться не такими очевидными, а сам код – не таким уж и простым

Тесты сложнее, чем сам код

LUXOFT Training Center

- Как правило, такая ситуация возникает в двух случаях:
  - низкая квалификация разработчика
  - плохой дизайн тестируемого кода
- Возникают ситуации, когда протестировать класс практически невозможно – напр. необходимо зарегистрировать пользователя, добавить данные в БД, предоставить config-файлы и т.д.

Тесты сложнее, чем сам код

LUXOFT Training Center

- Все это говорит о плохом дизайне системы – классы очень сильно зависят друг от друга, иногда нетривиальным образом
- Новые тесты писать при этом очень сложно, они часто ломаются
- В этом случае стоит начать написание тестов с простейших классов с минимальным числом зависимостей

Тесты сложнее, чем сам код

LUXOFT Training Center

- Все это говорит о плохом дизайне системы – классы очень сильно зависят друг от друга, иногда нетривиальным образом
- Новые тесты писать при этом очень сложно, они часто ломаются
- В этом случае стоит начать написание тестов с простейших классов с минимальным числом зависимостей

Тесты сложнее, чем сам код

LUXOFT Training Center

- Каждый класс, как и его тест, должны быть небольшими и понятными
- После того, как будет покрыт простейший функционал, можно перемещаться на более «высокие» уровни системы
- Одновременно с этим производить рефакторинг кода для уменьшения связности классов

Недостаточно времени

LUXOFT Training Center

- Если сравнивать написание классов с тестами и без, то безусловно, во втором варианте времени понадобится больше
- Однако давайте посмотрим на процесс разработки и написания тестов более широко

Недостаточно времени

LUXOFT Training Center

- При наличии модульных тестов, пишется только тот код, который необходим для их успешного выполнения
- Классы становятся меньше по размеру и выполняют четко определенный набор функций
- Таким образом, **пишется меньше кода**

Недостаточно времени

LUXOFT Training Center

- Разрабатывая небольшими шагами, вы всегда знаете, что вас ждет дальше – вы двигаетесь без промедления
- Использование unit-тестов существенно сокращает время, проводимое в отладчике
- Добавив небольшую порцию кода, мы тут же убеждаемся в его работоспособности

Недостаточно времени

LUXOFT Training Center

- Внеся изменения в классы, мы тут же убеждаемся, что не нарушили работы остальных классов
- Интеграция модулей проходит легче и быстрее, поскольку каждый из них хорошо оттестирован
- Уменьшается количество ошибок в коде, а так же время на их поиск и исправление

Недостаточно времени

LUXOFT Training Center

- Более качественный дизайн полученный в ходе такой разработки, помогает в дальнейшем улучшать код, делая это очень быстро

Не умею писать тесты

LUXOFT Training Center

- В этом нет ничего страшного, т.к. не обязательно сразу начинать тестировать сложные компоненты, используя все возможности xUnit и mock-фреймворков
- Для начала можно попытаться воспроизвести в тестах действия, которые разработчик делает вручную при тестировании модуля

Не умею писать тесты

LUXOFT Training Center

- Даже небольшая и «неуклюжая» тестовая обвязка намного лучше, чем еще не написанная «гибкая система модульного тестирования, покрывающая 100% функционала»
- Начинайте с простых приемочных тестов или тестирования классов низших уровней и продвигайтесь далее в глубь системы

Двукратное увеличение кода

LUXOFT Training Center

- По мнению некоторых разработчиков, модульные тесты – это двукратное увеличение кода!
- Это не так
- На самом деле оно троекратное (или даже больше)
- Однако давайте рассмотрим и другие стороны этого вопроса

Двукратное увеличение кода

LUXOFT Training Center

- Конечно, можно спроектировать систему так, чтобы изначально обязанности классов были распределены оптимальным образом
- Однако на практике это практически невозможно:
  - никто не может увидеть всей картины целиком
  - требования к системе могут меняться

Вопросы ?

Разработка через тестирование

IDyachenko@luxoft.com

git clone git://github.com/ivan-dyachenko/Trainings.git

https://github.com/ivan-dyachenko/Trainings