LUXOFT TRAINING

Разработка через тестирование Best Practice

Test Driven Development

Ivan Dyachenko <IDyachenko@luxoft.com>

Module 11

Содержание

LUXOFT Training Center

Best Practices

LUXOFT Training Center

- **Best practice** (Лучшая практика) – формализация уникального успешного практического опыта
- Согласно этой идее, в любой деятельности существует оптимальный способ достижения цели, который, оказавшись эффективным в одном месте, может оказаться столь же эффективным и в другом

Начинаем с наименее зависимых

- Если вы начнете тестировать «высокоуровневый» метод, то он может завалиться из-за некорректного значения, возвращаемого второстепенным методом внутри него
- Т.о. вы тратите дополнительное время на поиск источника проблемы
- В итоге вы все равно оттестируете сначала второстепенный метод

Простые тесты

LUXOFT Training Center

В идеале, тесты должны состоять из трёх простых шагов:

- Подготовка входных параметров
- Вызов тестируемого метода
- Проверка выходных параметров

Простые тесты

LUXOFT Training Center

- Любой тест завершается либо **успехом**, либо **провалом**
- Не должно быть «наполовину(частично) успешных» тестов
- Если тест провален, то проваленным считается и весь тестовый набор

Простые тесты

LUXOFT Training Center

- Стремитесь к тому, чтобы тесты содержали не более одной проверки
- Множество проверок в одном тесте может вызвать падение производительности
- Если не пройдет первая проверка, то остальные даже не выполнятся
- Данная практика помогает более точно находить места возникновения дефектов

Быстрые тесты

LUXOFT Training Center

- Чем быстрее выполняются тесты, тем чаще их можно запускать – быстрее получать фидбек
- Со временем количество тестов увеличивается, как и общее время их выполнения
- Даже один медленный тест «тормозит» выполнение всего тестового набора (слабое звено)

Константы в проверках

LUXOFT Training Center

- В первом случае, мы должны реализовать внутри теста ту же логику, что и в тестируемом методе
- В случае ошибки в методе, мы должны будем править как метод, так и тест
- Вторая проверка более понятна и проста в сопровождении

```
returnVal = methodUnderTest(input);
assertEquals(returnVal,
computeExpectedValue(input));
assertEquals(returnVal, 12);
```

Независимые тесты

LUXOFT Training Center

- Модульный тест должен тестировать поведение одного метода
- Тестирование нескольких методов одновременно может существенно увеличить время на рефакторинг и отлов ошибок

```
int return1 = myClass.add(1, 2);
int return2 = myClass.add(-1, -2);
assertTrue(return1 - return2 == 0);
```

- В случае ошибки будет трудно сказать, какой из методов отработал неправильно

Независимые тесты

LUXOFT Training Center

- Тесты должны быть независимы друг от друга
- Порядок выполнения не должен влиять на результат
- Тесты не должны использовать общих данных, описывающих состояние тестируемого объекта

Изолированные тесты

LUXOFT Training Center

- Модульные тесты должны быть изолированы от окружения такого как:
  - доступ к базе данных
  - вызов вебсервисов
  - переменные окружения
  - файлы настроек
  - системная дата и время
- Используйте заглушки – они легко пишутся, повторно используются и быстро работают

Именование и комментарии

LUXOFT Training Center

- Имя тест-метода должно четко определять, какой метод он тестирует
- В противном случае это ведет к увеличению затрат на поддержку и рефакторинг кода и тестов
- Любые нестандартные ситуации должны быть хорошо откомментированны

Сообщения об ошибках

LUXOFT Training Center

- Сообщения должны информативными и позволять однозначно идентифицировать причину ошибки
- Хорошие сообщения улучшают документирование кода

Тестирование private методов

- По данному вопросу нет однозначного мнения
- С одной стороны, мы должны быть уверены в поведении каждого из методов класса
- С другой стороны, цель модульного тестирования – проверка поведения интерфейсов класса, вне зависимости от внутренней реализации

Разделение по бизнес-модулям

LUXOFT Training Center

- Создавайте наборы тестов для каждого из модулей
- Используйте иерархический подход
- Уменьшайте время выполнения наборов, разбивая их на подмодули
- Маленькие наборы можно выполнять чаще

Измеряйте покрытие

LUXOFT Training Center

- Зачастую, трудно понять, насколько хорошо оттестирована та или иная часть кода
- Современные инструменты анализа покрытия интегрируются в IDE и инструменты автоматических сборок (ant, maven), и дают наглядное представление степени покрытия кода на различных уровнях

Полная автоматизация

Каждый из этих шагов должен выполняться автоматически:

- выполнение тестов
- сбор результатов
- определение успешности выполнения тестов
- нотификация о результатах (e-mail, IM, страница на dashboard'е проекта, иконка в трее, флажок в IDE, сигнальный огонь и т.д.)

Вопросы ?

Разработка через тестирование

IDyachenko@luxoft.com

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

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