

Методика разработки через тестирование

Лекция по курсу «Основы тестирования ПО»

© 2019 Парамонов Илья Вячеславович

Понятие разработки через тестирование

Определение

Разработка через тестирование (test driven development, TDD) — это методика **итеративной** разработки, основанная на **написании тестов до написания кода**, проверяемого этими тестами

- Основная идея TDD — использовать тесты, написанные на основе спецификации, как индикатор готовности соответствующей функциональности
- В момент написания теста разработчик фокусируется на спецификации, а не на реализации, и потому не имеет предвзятого отношения к коду

Цикл разработки через тестирование

Разработка через тестирование — это циклический процесс

Фазы цикла TDD

1. Написание теста
2. Написание кода
3. Рефакторинг

Данный цикл многократно повторяется в ходе разработки

Фаза 1: написание теста

Описание фазы

В данной фазе разработчик **выбирает фрагмент функциональной спецификации**, который он будет реализовывать, и **разрабатывает тест на эту функциональность** в предположении, что **код уже существует**

Написанный тест должен завершаться неуспешно

Выполнение фазы 1: написание теста

- Выбранный фрагмент спецификации должен быть минимальным по объёму (атомарным)
- Если необходимые для написания теста классы, методы, функции существуют, разработчик их использует
- Если их нет, разработчик добавляет пустые заглушки
- Код должен успешно компилироваться
- Если тест выполняется успешно, то данная функциональность либо уже существует, либо тест разработан методически неверно (непоказательный тест)

Фаза 2: написание кода

Описание фазы

В данной фазе разработчик пишет **минимально необходимый код, предоставляющий функциональность**, которая заставит тест **выполняться успешно**

Все тесты, включая только что написанный, должны завершаться успешно

Выполнение фазы 2: написание кода

- Код не должен пытаться предоставить функциональность, не покрытую тестом
- Код должен быть осмысленным и направленным на предоставление функциональности, а не просто на выполнение теста
- На данном этапе код может быть «грязным», содержащим дублирование, не удовлетворяющим требованиям к оформлению и т. д.
- Результаты данной фазы фиксируются в системе управления версиями

Фаза 3: рефакторинг

Определение

Рефакторинг — это процесс изменения исходного кода, **не затрагивающий его функциональность**, но **улучшающий его читаемость и сопровождаемость**

Описание фазы

В данной фазе написанный код **перерабатывается** так, чтобы он удовлетворял **требованиям, предъявляемым к его качеству** (осмысленные имена, отсутствие дублирования, короткие методы, другие требования корпоративного стандарта)

Все тесты должны завершаться успешно

Выполнение фазы 3: рефакторинг

- Рефакторинг позволяет гарантировать соблюдение кодом архитектуры проекта, а также выполнять изменение этой архитектуры короткими итерациями
- Правильность выполнения рефакторинга гарантируется выполнением тестов
- При необходимости тесты также могут быть подвергнуты рефакторингу
- Результаты данной фазы фиксируются в системе управления версиями

- Отделение фаз позволяет разработчику лучше фокусироваться на различных аспектах процесса разработки
- Работа со спецификацией до написания кода обеспечивает улучшение архитектуры и большее внимание к потребностям пользователя
- Вероятность регрессии минимизируется за счёт хорошего покрытия кода
- Разработанные тесты могут служить документацией, которая всегда актуальна

- Высокий порог вхождения
- Сложно применять в областях, где требуется функциональное, а не модульное тестирование (например, в разработке пользовательских интерфейсов)
- Сама по себе методика не гарантирует выработки качественной высокоуровневой архитектуры
- Разработанные тесты требуют поддержки и уязвимы к серьёзным изменениям архитектуры