



Behavior Driven Development

Или же
BDD, ТЫ КТО ТАКОЙ?

Выполнил студент 4
курса ИСиП
Лешкевич С.С.

BDD (сокр. от англ. **Behavior-driven Development**, дословно «разработка через поведение») — это процесс разработки программного обеспечения, являющаяся ответвлением от методологии разработки через тестирование (TDD).

BDD методология является расширением TDD в том смысле, что перед тем как написать какой-либо тест, необходимо сначала описать желаемый результат от добавляемой функциональности на предметно-ориентированном языке. После того как это будет проделано, конструкции этого языка переводятся специалистами или специальным программным обеспечением в описание теста.

BDD фокусируется на следующих вопросах:

- С чего начинается процесс.
- Что нужно тестировать, а что нет.
- Сколько проверок должно быть совершено за один раз.
- Что можно назвать проверкой.
- Как понять, почему тест не прошёл.

Исходя из этих вопросов, BDD требует, чтобы имена тестов были целыми предложениями, которые начинаются с глагола в сослагательном наклонении и следовали бизнес целям. Описание приемочных тестов должно вестись на гибком языке пользовательской истории, например,

Как [роль того, чьи бизнес интересы удовлетворяются] я хочу, чтобы [описание функциональности так, как она должна работать], для того чтобы [описание выгоды].

Критерии приёмки должны быть описаны через сценарий, который реализует пользователь, чтобы достигнуть результата.

Язык Gherkin

Ключевое слово на английском языке	Русскоязычная адаптация	Описание
Story (Feature ^[3])	История	Каждая новая спецификация начинается с этого ключевого слова, после которого через двоеточие в сослагательной форме пишется имя истории.
As a	Как (в роли)	Роль того лица в бизнес-модели, которому данная функциональность интересна.
In order to	Чтобы достичь	В краткой форме какие цели преследует лицо.
I want to	Я хочу, чтобы	В краткой форме описывается конечный результат.
Scenario	Сценарий	Каждый сценарий одной истории начинается с этого слова, после которого через двоеточие в сослагательной форме пишется цель сценария. Если сценариев в одной истории несколько, то после ключевого слова должен писаться его порядковый номер.
Given	Дано	Начальное условие. Если начальных условий несколько, то каждое новое условие добавляется с новой строки с помощью ключевого слова And.
When	Когда (прим.: что-то происходит)	Событие, которое инициирует данный сценарий. Если событие нельзя раскрыть одним предложением, то все последующие детали раскрываются через ключевые слова And и But.
Then	Тогда	Результат, который пользователь должен наблюдать в конечном итоге. Если результат нельзя раскрыть одним предложением, то все последующие детали раскрываются через ключевые слова And и But.
And	И	Вспомогательное ключевое слово, аналог конъюнкции.
But	Но	Вспомогательное ключевое слово, аналог отрицания.

Следующий пример демонстрирует спецификацию поведения с использованием языка Gherkin.

<pre>Story: Returns go to stock As a store owner In order to keep track of stock I want to add items back to stock when they're returned. Scenario 1: Refunded items should be returned to stock Given that a customer previously bought a black sweater from me And I have three black sweaters in stock. When they return the black sweater for a refund Then I should have four black sweaters in stock. Scenario 2: Replaced items should be returned to stock Given that a customer previously bought a blue garment from me And I have two blue garments in stock And three black garments in stock. When they return the blue garment for a replacement in black Then I should have three blue garments in stock And two black garments in stock.</pre>	<p>История: Возвращённый товар должен быть учтён на складе</p> <p>Как владелец магазина Чтобы следить за запасами на складе Я хочу восстанавливать записи о товарах, которые возвращаются на склад.</p> <p>Сценарий 1: Возвращенные товары должны размещаться на складе Дано то, что ранее покупатель приобрёл у меня чёрный свитер И на моём складе уже есть три точно таких же. Когда покупатель возвращает приобретенный свитер Тогда я должен видеть, что сейчас на складе 4 чёрных свитера.</p> <p>Сценарий 2: Замененные предметы должны быть возвращены на склад Дано то, что клиент приобрёл у меня одежду синего цвета И на моём складе есть два этих наименования синего цвета И три наименования чёрного цвета. Когда клиент возвращает одежду синего цвета, чтобы заменить на такую же, но чёрную Тогда я должен видеть, что сейчас на складе три наименования для одежды синего цвета И два наименования для одежды чёрного цвета.</p>
---	---

Примеры BDD фреймворков

C/C++

- ▶ Catch
- ▶ CBehave

Кроссплатформенные

- ▶ Cucumber
- ▶ Squish
- ▶ Yulup

Python

- ▶ behave[4]
- ▶ lettuce[5]
- ▶ radish[6]
- ▶ Robot Framework[7]

Ruby

- ▶ RBehave
- ▶ RSpec

Java

- ▶ JBehave
- ▶ Jnario
- ▶ JGiven
- ▶ Vividus Framework

PHP

- ▶ Behat
- ▶ Codeception



Спасибо за внимание!