

# BDD в web разработке

# Оглавление

1. Особенности
2. Процесс написания
3. Jenkins
4. Отчеты

BDD - это разработка теста через поведение.

BDD - это эмулятор живого человека на сайте.

# Особенности

- 1) Пишут не тесты, а шаги, которые надо пройти, для достижения какого - либо результата имитируя поведение пользователя
- 2) Тестирую не код сайта, а поверхность, то с чем работает клиент
- 3) Написанный тест легко читается и понимается разработчиком, начальством, клиентам.

# Процесс написания

## Шаг 1

- Получив заказ на доработку, создается пользовательская история с примерами в спеклоге
- пользовательская история написана на языке gherkin

## Шаг 2

- Я в вижуал студии начинаю писать тест на эту пользовательскую историю, пока тест не написан я помечаю его тегом @...
- для того что бы в отчетах было видно, что работы над тестом ведутся

# Шаг 3

- После завершения работы над тестом я запускаю его и он становится в отчетах красным (На тест нужно еще написать функционал)

[Scenario: Сделать заказ на сайте от Юридического лица \(розница\), Variant 3 retry #1 retry #2](#)

0%



[Scenario: Сделать быстрый заказ на сайте \(В Корзине\), 3 511 руб. retry #1 retry #2](#)

0%



[Scenario: Сделать быстрый заказ на сайте \(В Оформление Заказа\), 3 511 руб. retry #1 retry #2](#)

0%



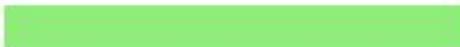
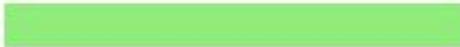
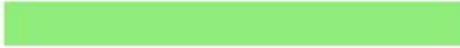
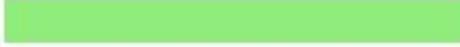
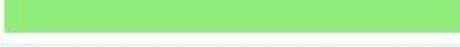
## Шаг 4

- Когда тест написан я кидаю его в репозиторий, где его видят программисты и начинают над ним работу, а сам приступаю к другому тесту

## Шаг 5

- После работы программистов и реализации функционала теста, он становится зеленым, что означает, что тест проходит успешно

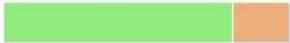
# Зеленые тесты

<a href="#">Scenario: Сделать заказ на сайте от Физического лица (розница), Variant 0</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Физического лица (розница), Variant 1</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Физического лица (розница), Variant 2</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Физического лица (розница), Variant 3</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Юридического лица (розница), Variant 0</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Юридического лица (розница), Variant 1</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Юридического лица (розница), Variant 2</a>	100%	
<a href="#">Scenario: Сделать заказ на сайте от Юридического лица (розница), Variant 3</a>	100%	
<a href="#">Scenario: Сделать быстрый заказ на сайте (В Корзине), 3 511 руб.</a>	100%	
<a href="#">Scenario: Сделать быстрый заказ на сайте (В Оформление Заказа), 3 511 руб.</a>	100%	

## Шаг 6

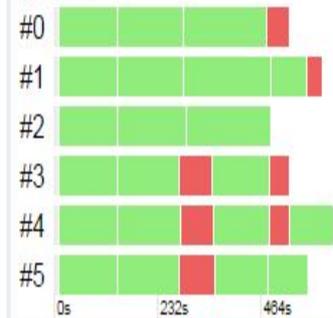
- после реализации функционала мы на руках имеем автотест, отчеты которого можно продемонстрировать заказчику.

# Отчет заказчику

Success rate	Tests	Succeeded	Failed	Pending	Ignored	Skipped
80% 	20 (test executions: 28)	16	4	0	0	0

## Test Timeline Summary

### thread



## Test Result View



## Feature Summary

Feature	Success rate	Tests	Succeeded	Failed	Pending	Ignored	Skipped
<a href="#">US10 - Сделать заказ на сайте</a>	80% 	20 (test executions: 28)	16	4	0	0	0

- **Количество зеленых тестов – единственный надежный критерий оценки того, что уже сделано и что предстоит.**

К сожалению, практика показывает, что разработчики, пусть и из благих побуждений, могут рапортовать «да, да, завтра все заработает» месяцами.

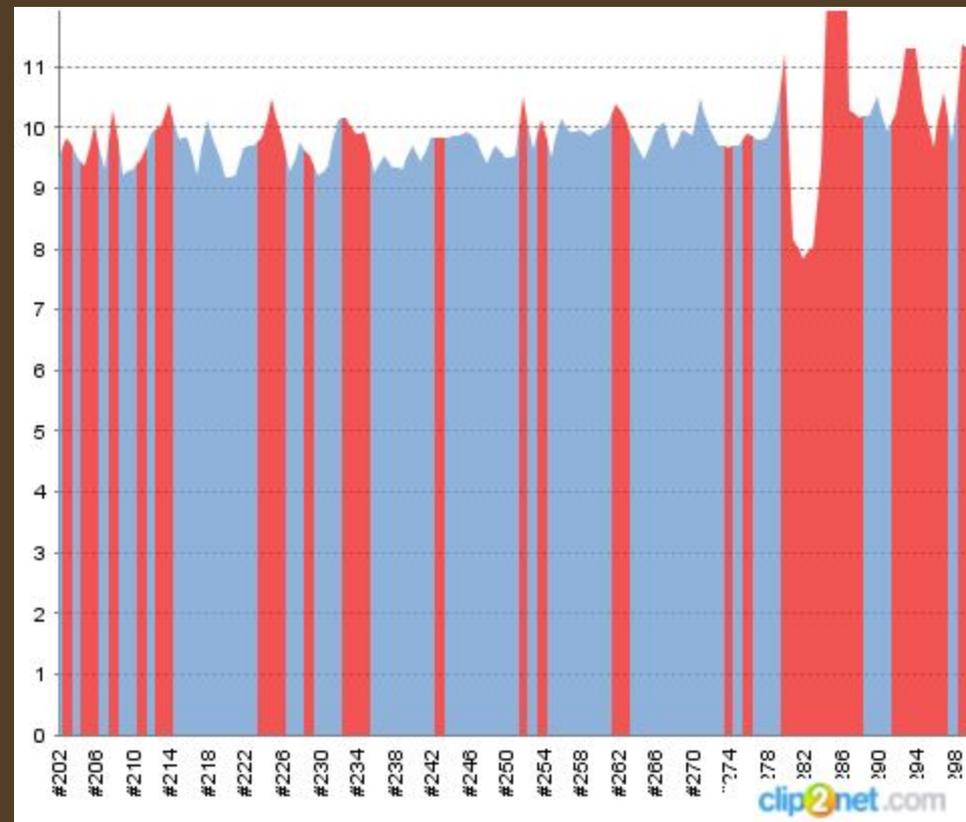
График не будет врать: **фича готова, когда все ее тесты зеленые.**

# Jenkins

- После того как я залил тест в репозиторий, его автоматически начнет запускать робот (Jenkins)
- Jenkins запускает тесты тогда, когда я хочу, что бы тесты запускались.

# Сборки Jenkinsa

✓ #257	17.11.2016 11:45
✓ #256	17.11.2016 11:30
✓ #255	17.11.2016 11:15
! #254	17.11.2016 11:00
✓ #253	17.11.2016 10:45
! #252	17.11.2016 10:30
✓ #251	17.11.2016 10:15
✓ #250	17.11.2016 10:00
✓ #249	17.11.2016 9:45
✓ #248	17.11.2016 9:30
✓ #247	17.11.2016 9:15
✓ #246	17.11.2016 9:00
✓ #245	17.11.2016 8:45
✓ #244	17.11.2016 8:30
! #243	17.11.2016 8:15
✓ #242	17.11.2016 8:00
✓ #241	17.11.2016 7:45



- Тесты проходят и если тест, по каким либо причинам не проходят, то Jenkins дает знать письмом на почту, оповещением на телефон, или просто сообщением в рабочий чат

Mailroom - 1ceo@1service.ru - Nov-15 3:28 PM

**Внимание! На сайте проблемы.**

Honey\_Order\_test - Build # 83 - Failure: Check console output at [http://78.155.197.197:8080/job/Honey\\_Order\\_test/83/](http://78.155.197.197:8080/job/Honey_Order_test/83/) to view the results.

Настоящим письмом сообщаем что на сайте невозможно выполнить заказ. Команда 1сервиса уже в курсе проблемы и начала расследование! [http://78.155.197.197:8080/job/Honey\\_Order\\_test/HTML\\_Report/](http://78.155.197.197:8080/job/Honey_Order_test/HTML_Report/) dima-plosk@mail.ru

# Отчеты

- Jenkins так же делает подробные отчеты по тесту, по ним можно понять сколько времени выполнялся тот, или иной шаг, на каком шагу тест упал, сколько тестов упали, а сколько прошли
- Эти отчеты можно предъявлять заказчику

Спасибо за внимание