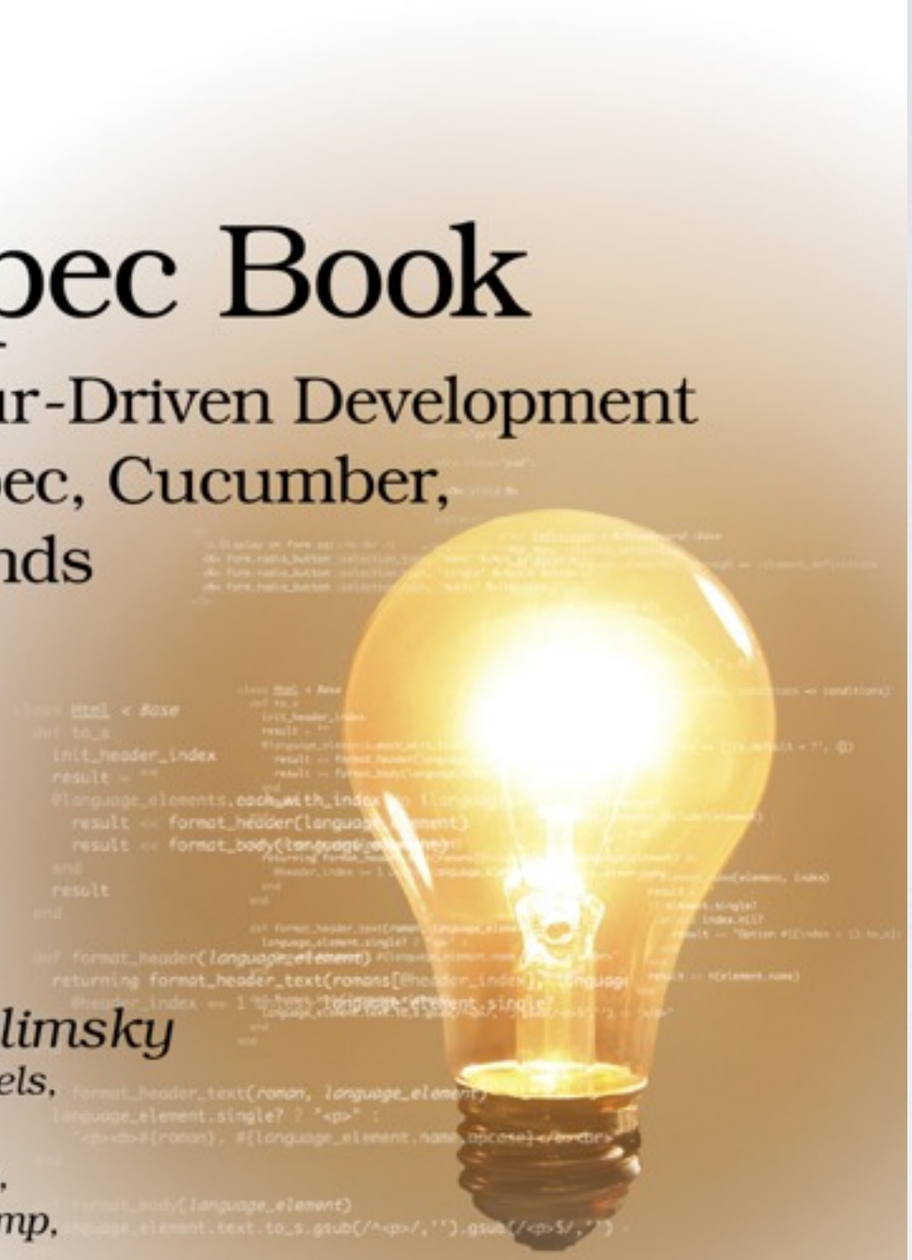
BEHAVIOR-DRIVEN DEVELOPMENT WITH RSPEC, CUCUMBER AND FRIENDS

David Chelimsky

The Pragmatic Programmers

The RSpec Book

Behaviour-Driven Development with RSpec, Cucumber, and Friends



David Chelimsky
with Dave Astels, Zach Dennis, Aslak Hellesøy, Bryan Helmkamp, and Dan North

Foreword by Robert C. Martin (Uncle Bob)

Edited by Jacquelyn Carter

The Facets of Ruby Series

2010

The Pragmatic Programmers

The RSpec Book

Behaviour-Driven Development with RSpec, Cucumber, and Friends

David Chelimsky
with Dave Astels, Zach Dennis, Aslak Hellesøy, Bryan Helmkamp, and Dan North

Foreword by Robert C. Martin (Uncle Bob)

Edited by Jacquelyn Carter

The Facets of Ruby Series

DAVID CHELIMSKY

Lead developer/maintainer of RSpec

Contributor of Cucumber and Rails

Software Engineer at DRW Trading Group

TDD

Test-Driven Development

TDD

Kent Beck, 2003

@KentBeck

TDD

practice that involves writing tests
before writing the code being tested

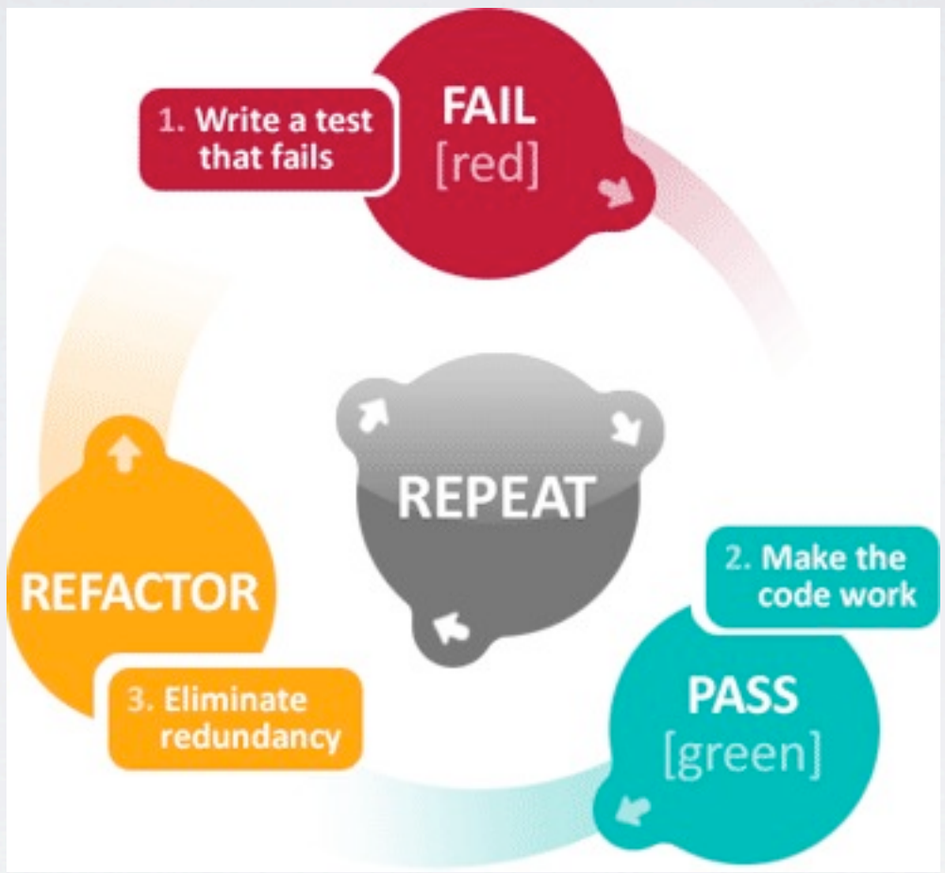
TDD

1. Begin by writing a very small test for code that does not yet exist

2. Run the test
   (naturally it fails)

3. Write just the enough code to make that test pass
   No more

TDD


1. Write a test that fails
FAIL
[red]
REPEAT
2. Make the code work
PASS
[green]
3. Eliminate redundancy
REFACTOR

TDD

Os problemas começam

The idea of unit testing that often leads new TDDers to verify things such as making sure that a `register()` method stores a `Registration` in a collection and that collection is specifically an `Array`.

esse nível de detalhe

This sort of detail in a test creates a **dependency** in the test on the **internal structure** of the object being tested.

TDD

This dependency means that if other requirements guide us to change the `Array` to a `Hash`, this test will fail

even though the **behavior** of the object hasn't changed.

TDD

essa fragilidade

This brittleness can make test suites much more **expensive to maintain** and is the primary reason for test suites to become **ignored** and, ultimately, **discarded**.

BDD

Behavior Driven Development

BDD

Dan North, 2003

@tastapod

BDD

BDD puts the focus on **behavior** instead of structure

BDD

The problem with testing an object's internal structure is that we're testing what an **object is** instead of **what it does**.

What an object **does** is significantly **more important**.

BDD

Stakeholders don't usually care that data is being persisted in an relational database.

"the database" generally mean is that it's stored somewhere and they can get it back.

BDD

We believe that most of the problems that software development teams face are **communication problems**

BDD aims to help communication by **simplifying the language** we use to describe scenarios in which the software will be used

BDD

Given some context

When some event occurs

Then I expect some outcome

BDD

Given some context

When some event occurs

Then I expect some outcome

BDD triad

RSPEC

RSPEC

Provides a Domain Specific Language with which you can express executable examples of the expected behavior of your code.

RSPEC

BDD is an approach to software development that combines Test-Driven Development, Domain Driven Design and Acceptance Test-Driven Planning.

RSpec helps you do the TDD part of that equation, focusing on the documentation and design aspects of TDD.

RSPEC

Imagine that you were talking to a customer requesting software for her bank. Part of that conversation might well look like this:

You: Describe an account when it is first created.
Customer: It should have a balance of $0.

RSPEC

You: Describe an account when it is first created.
Customer: It should have a balance of $0.

```
describe Account, "when first created" do
  it "should have a balance of $0" do
    ...
  end
end
```

RSPEC

```
describe Account, "when first created" do
  it "should have a balance of $0" do
    account = Account.new
    account.should be_empty
  end
end
```

RSPEC

```
describe Account, "when first created" do
  it "should have a balance of $0" do
    account = Account.new
    account.should be_empty
  end
end
```

RSPEC

We're talking about the specification of an **object**, not a **system**

You *could* specify application behavior with RSpec

RSPEC

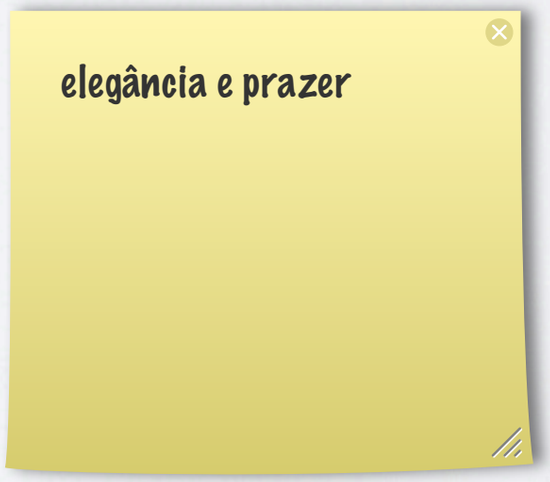
For specifying application behavior, we want something that communicates in broader strokes

only an **outline** is given, **without fine details**

CUCUMBER

CUCUMBER

BDD with elegance and joy



CUCUMBER

BDD is a agile methodology

It takes some of its cues from Extreme Programming

sinais

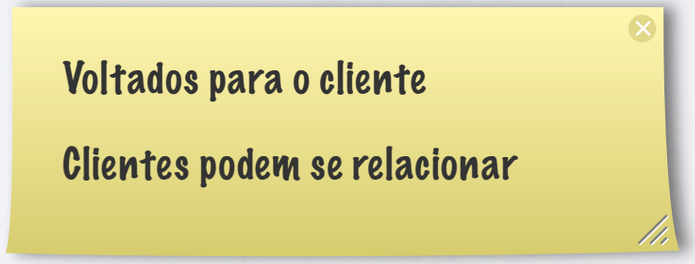
Including a variation if Acceptance Test-Driven Development called Acceptance Test-Driven Planning

ACCEPTANCE TEST-DRIVEN PLANNING

We use customer acceptance tests to drive the development of code

these are the result of a collaborative effort between the customer and the delivery team

they are **customer facing** and must be **expressed** in a language and format that **customers can relate to**



CUCUMBER

Cucumber reads plain-text descriptions of application features with example **scenarios**

uses the scenario steps to **automate** interaction

CUCUMBER

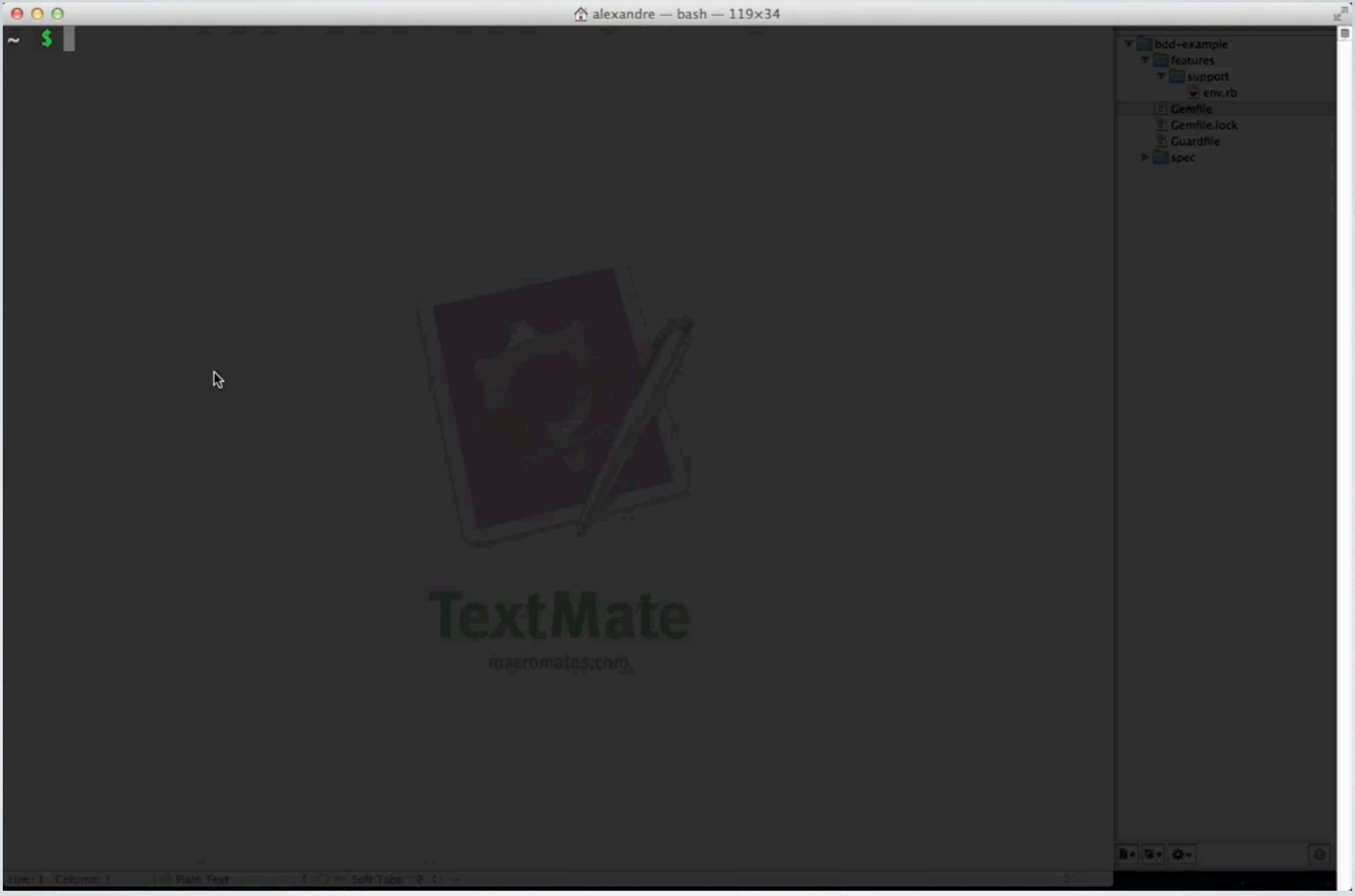
```
Feature:  check bank statement online
  In order to reduce the time I spend going to the bank
  As a bank customer
  I want to check my bank statement online

  Scenario: check statement
    Given I have 5 bank releases
    When  I get my bank statement
    Then  I should have my 5 releases sorted by date
```

BDD

We use **Cucumber** to describe the **behavior of applications**

We use **RSpec** to describe the **behavior of objects**

IN ACTION


alexandre — bash — 119×34
~ $
bdd-example
features
support
env.rb
Gemfile
Gemfile.lock
Guardfile
spec
TextMate
macromates.com
Line: 1 Column: 1
Plain Text
Soft Tabs: 2

CUCUMBER PT-BR

```
Funcionalidade: Adição
  Para evitar erros bobos
  Como um péssimo matemático
  Eu quero saber como somar dois números

  Cenário: Adicionar dois números
    Dado que eu digitei 50 na calculadora
    E que eu digitei 70 na calculadora
    Quando eu aperto o botão de soma
    Então o resultado na calculadora deve ser 120
```

JAVA BDD

C++ BDD

CppSpec

cbehave

A Behavior Driven Development Framework for C