Auditoria e Qualidade de Sistemas

Prof. Edgard Davidson C. Cardoso

Qualidade de Sistemas

BDD – BEHAVIOR DRIVEN DEVELOPMENT

una

Dificuldades ao inicar com TDD

- Dificuldades
  - Por onde começar?
  - O que testar?
  - O teste tem que começar falhando?
- Essas eram as mesmas dificuldades de Dan North,
  - Cansado de ver esses problemas então ele idealizou o BDD

O que é BDD?

- BDD – Behavior Driven Development
  - ou Desenvolvimento Dirigido pelo Comportamento

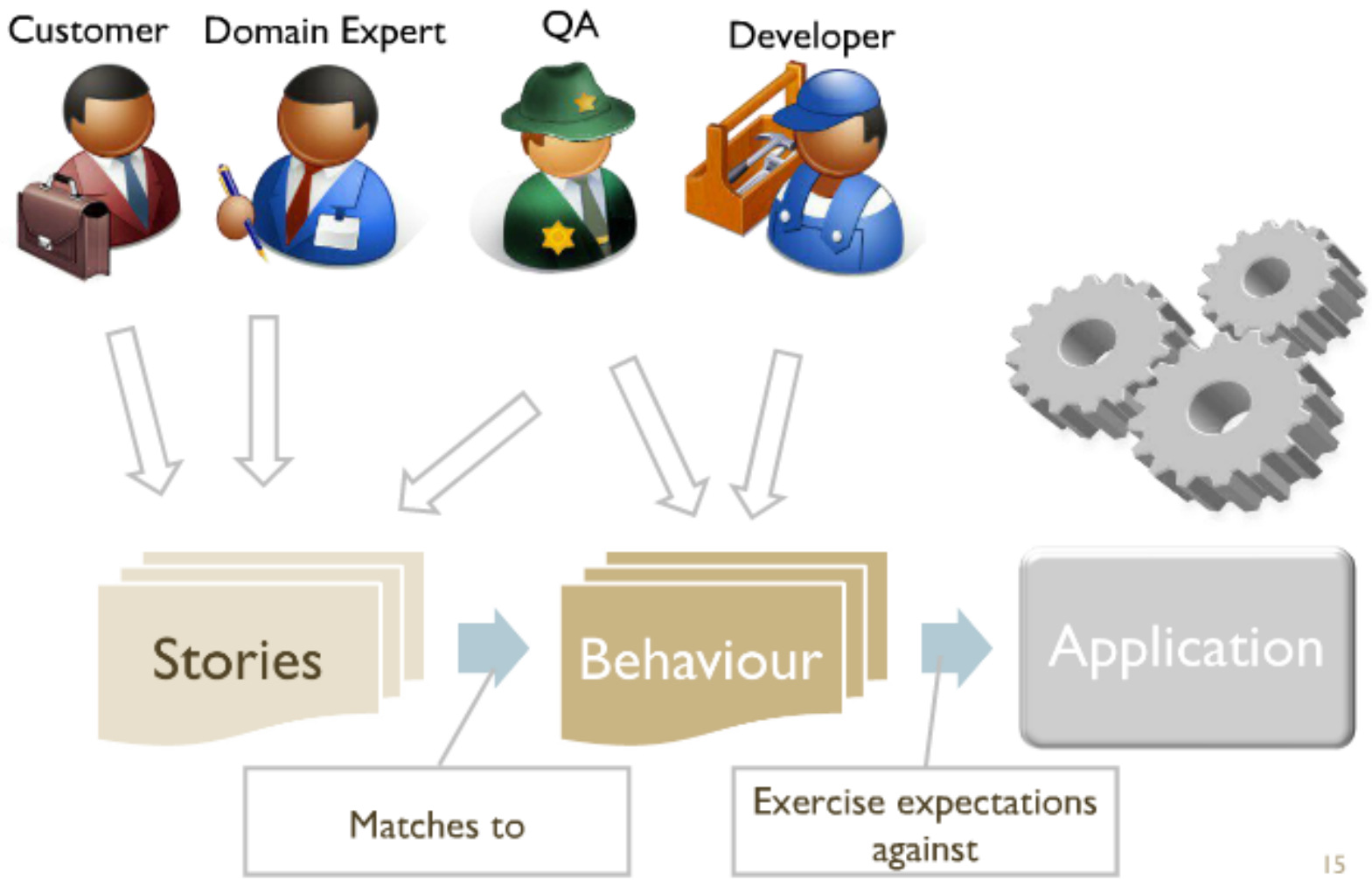
- É uma técnica de desenvolvimento ágil que visa integrar regras de negócios com linguagem de programação

- BDD provê uma documentação dinâmica

O que é BDD?

- O foco em BDD é a linguagem e interações usadas no processo de desenvolvimento de software.
- Desenvolvedores que usam BDD (Behavior-Driven developers) usam sua língua nativa em combinação com a *linguagem ubíqua*(ubiquitous language) usada no processo de desenvolvimento de software.
- Isso permite que os desenvolvedores foquem em por quê o código deve ser criado, ao invés de detalhes técnicos, e minimiza traduções entre linguagem técnica na qual o código é escrito e outras linguagens de domínio, usuários, clientes, gerência do projeto, etc.

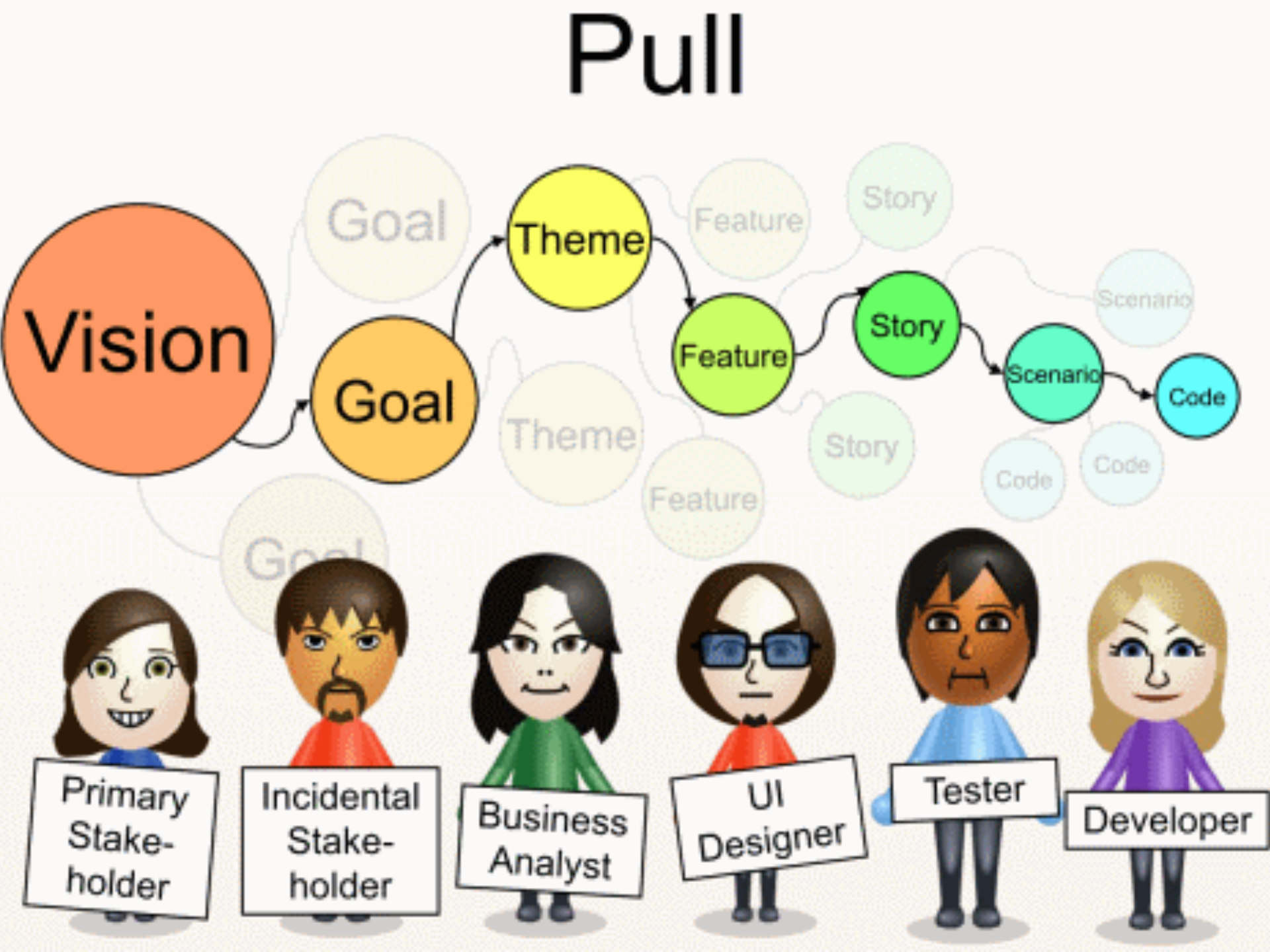
Processo BDD



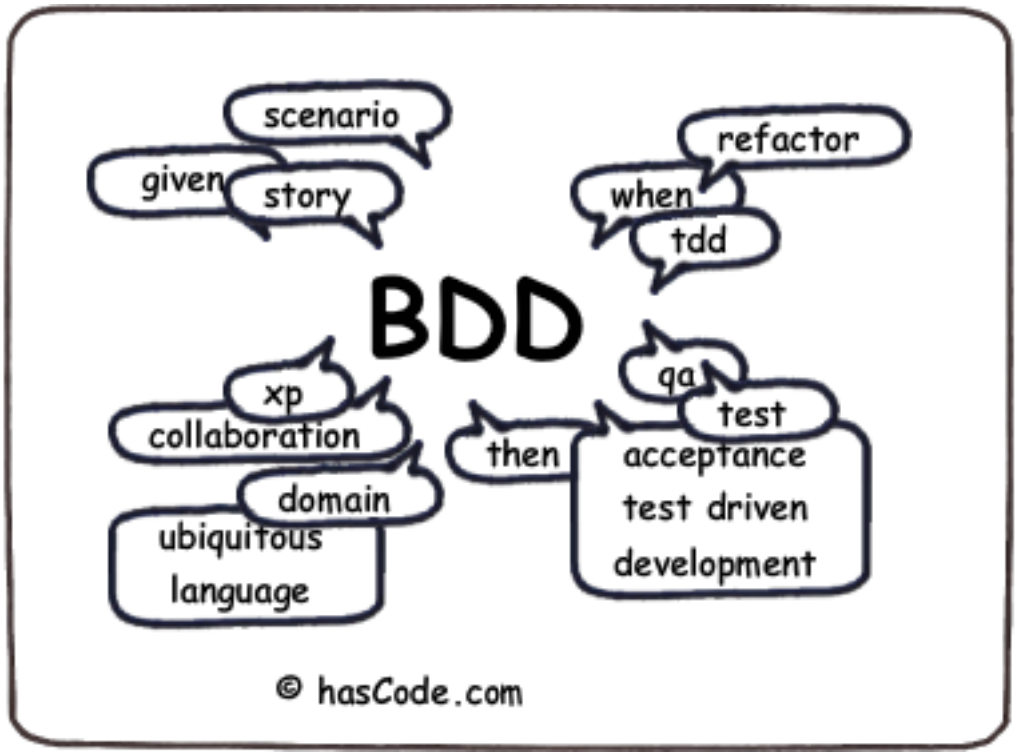
Processo BDD

- Crie cenários para uma funcionalidade específica
- Em seguida siga os passos:
  1. Para cada cenário descrito para uma funcionalidade
  2. Execute o cenário – ele vai falhar (red)
  3. Define o primeiro passo – ainda falha (red)
  4. Escreva rapidamente o código par fazer os passos pasarem – (green)
  5. Refatore o código e repita os passos 4 e 5 para cada "step" até o passo 6
  6. O cenário passa green
  7. Refatore o código da aplicacão

Pull

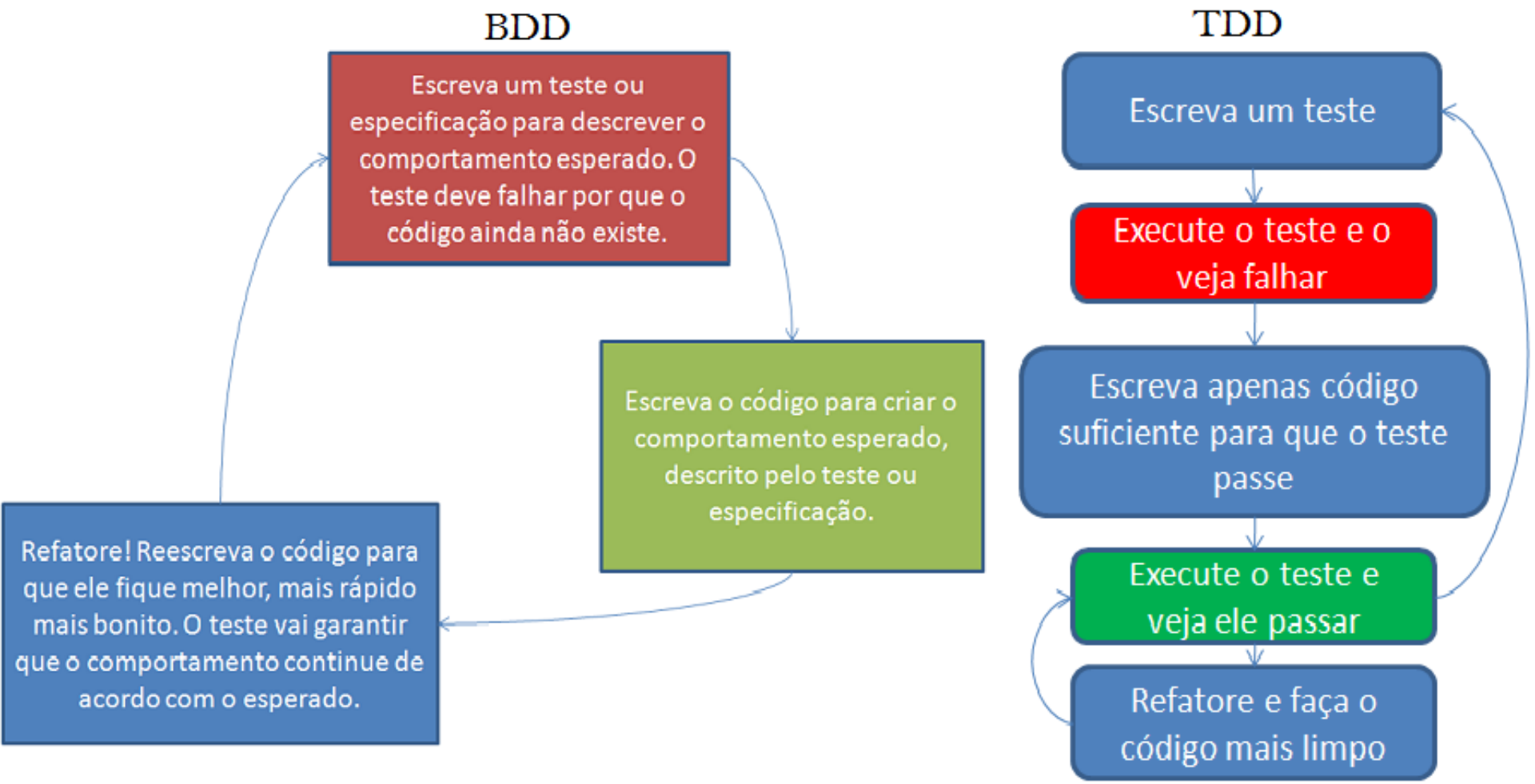

Vision
Goal
Theme
Feature
Story
Scenario
Code
Primary Stake-holder
Incidental Stake-holder
Business Analyst
UI Designer
Tester
Developer

Contexto BDD



una

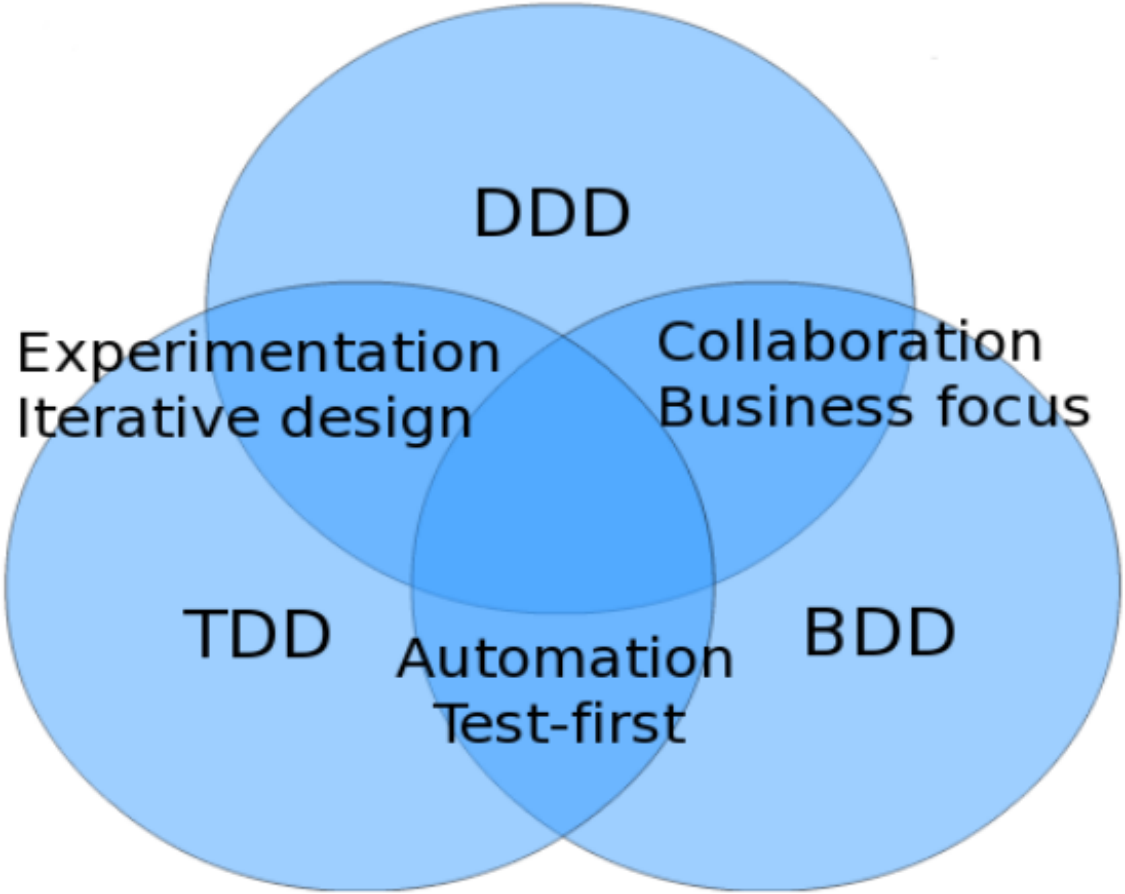
Ciclo BDD e TDD



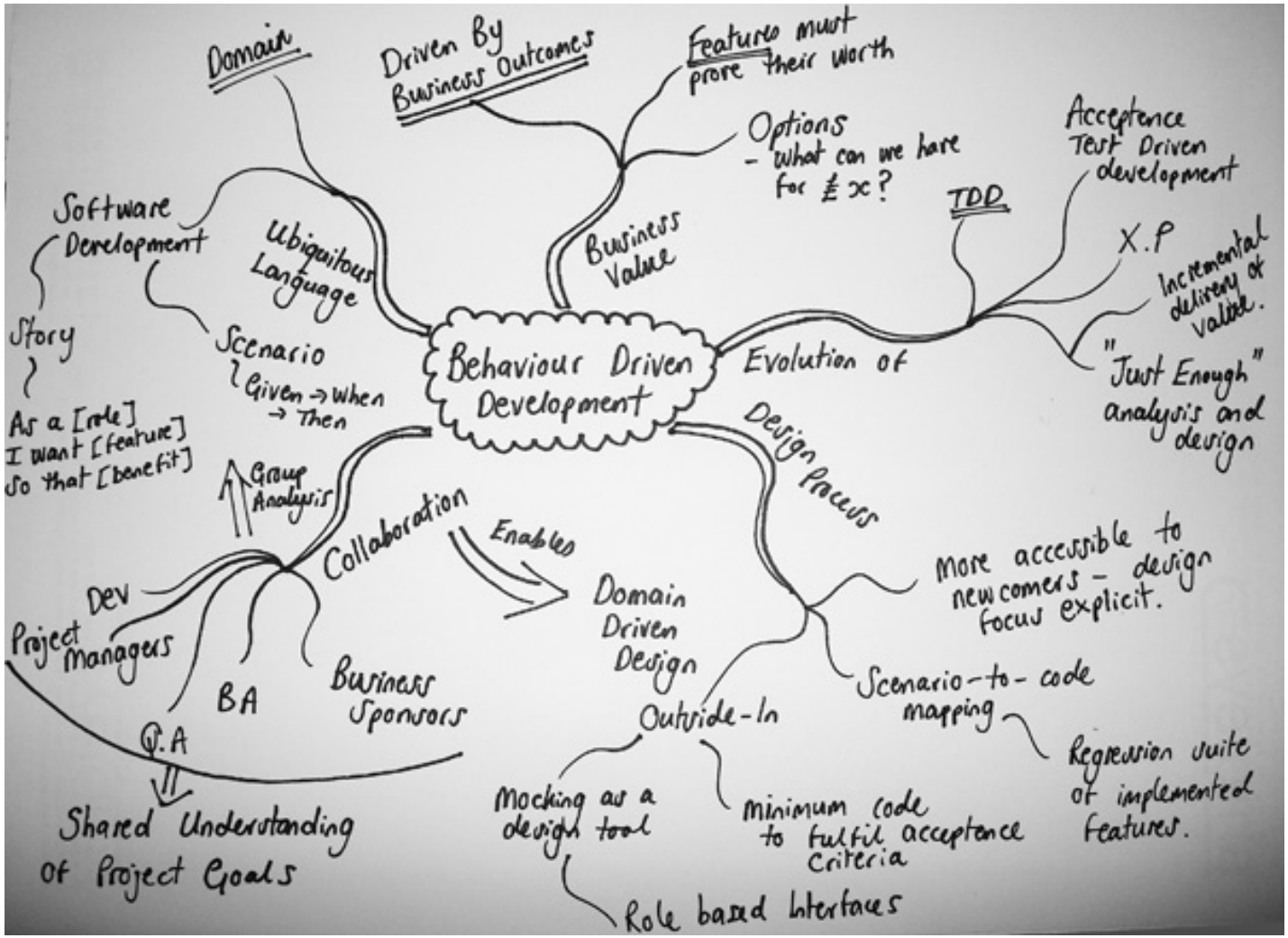
Comparação TDD e BDD

Fase	TDD	BDD
1	Escrever um teste	Escrever uma especificação do comportamento esperado
2	Executar os testes e ver eles falhando	Executar as especificações e ver elas falhando
3	Escrever código suficiente para que o teste passe	Escrever código suficiente para implementar o comportamento esperado/fazer a especificação passar
4	Executar os testes e ver eles passando	Executar as especificações e ver elas passando
5	Refatorar o código	Refatorar o código

BDD X DDD x TDD



Mapa Mental do BDD



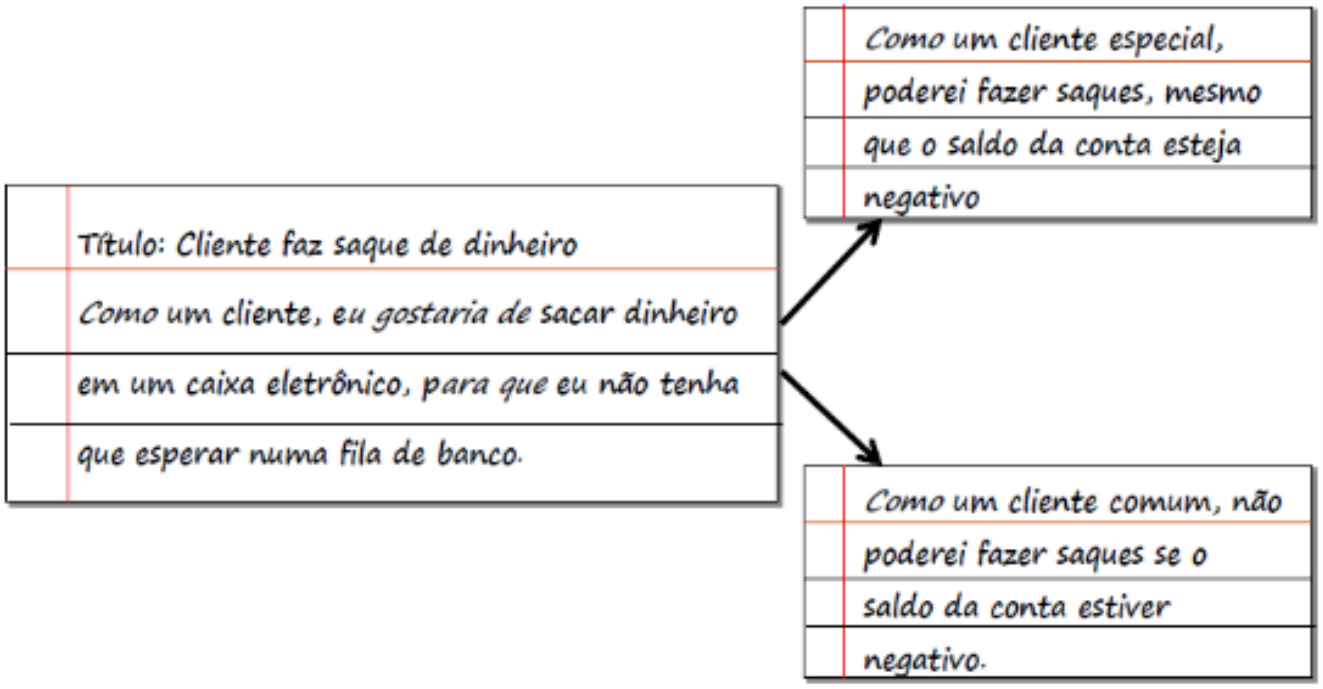
Ferramentas de Behavior, Mock, Teste

Pyhton	Ruby	Java	.net
• Pyhistorian • Pyccuracy • Freshen • pyCukes • pMock • Mockito • Lubidrio • Shoud-DSL	• Cucumber • Webrat • Rspec • Remakable • Mocha • Factory Girl • Machinist • Object Doddy	• Jbehave • Easyb • Mockito • EasyMock • Jmock • Homcrest • Junit • TestNG	• NBehave • Cuke4nuke • SpecFlow • Nspec • Mspec • Specunit • Cucumber + IronRuby

Template

```
Título (uma linha descrevendo a história)

Narrativa:
Como [o papel]
Eu quero [recurso]
Assim que [benefício]

Critérios de Aceitação: (apresentado como Cenários)

Cenário 1: Título
Dado contexto []
  E [um pouco mais de contexto] ...
Quando [eventos]
Então [resultado]
  E [outro resultado ...]

Cenário 2: ...
```

Exemplo



Estória do Usuário

No Arquivo Texto

```
Story: Cliente faz saque de dinheiro
Como um cliente, eu gostaria de sacar dinheiro
em caixa eletrônico, para que eu não tenha
que esperar numa fila de banco

Scenario: Cliente especial com saldo negativo
Given um cliente especial com saldo atual de -200 reais
When for solicitado um saque no valor de 100 reais
Then deve efetuar o saque e atualizar o saldo da conta para -300 reais

Scenario: Cliente comum com saldo negativo
Given um cliente comum com saldo atual de -300 reais
When solicitar um saque de 200 reais
Then não deve efetuar o saque e deve retornar a mensagem Saldo Insuficiente
```

Qualidade de Sistemas

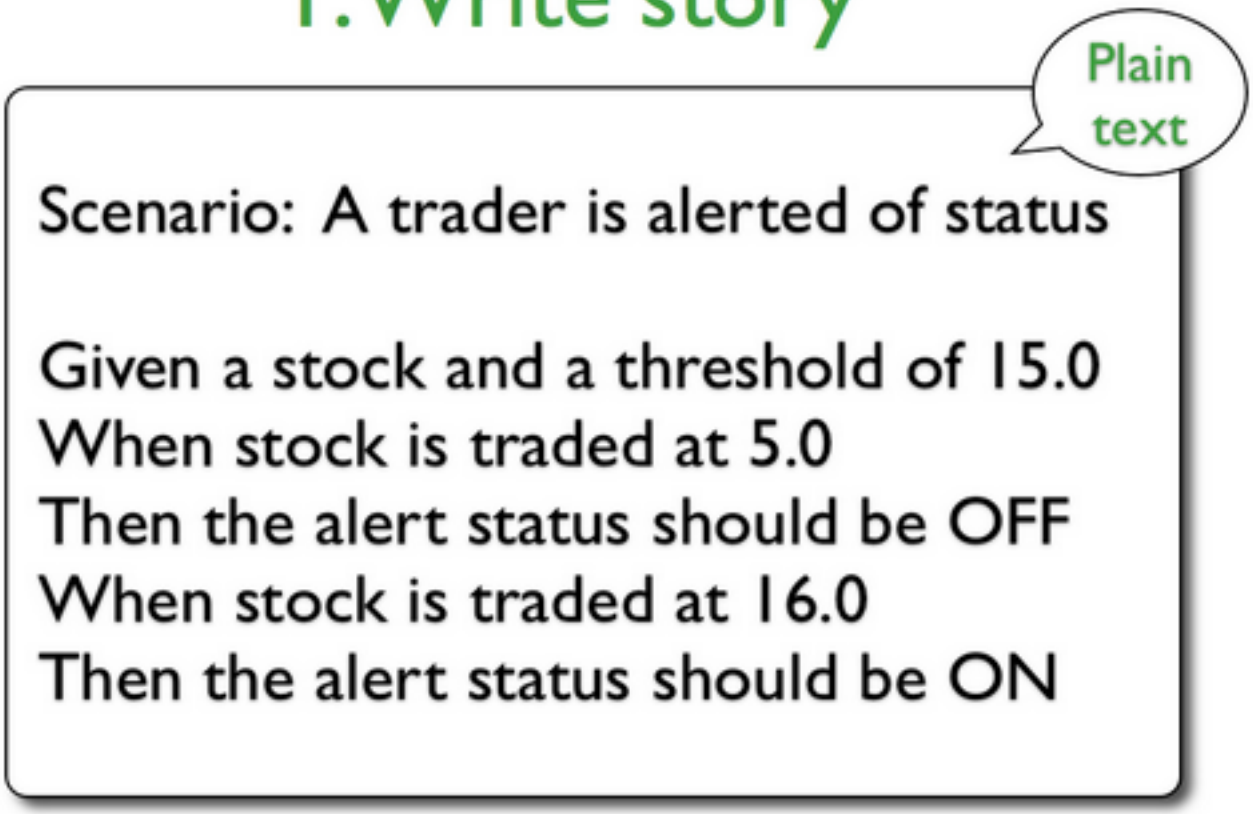
JBEHAVE

O que é o jbehave

- JBehave é um framework de Behaviour-Driven Development (BDD).
  - BDD é uma **evolução** do test-driven development (TDD) e de Acceptance-Teste Driven Design(ATDD)
  - Se destina a tornar essas práticas mais acessíveis e intuitivas para novatos e experts.
  - Ela muda o vocabulário de ser "**baseado em teste**" para "**baseado em comportamento**"
  - Se posiciona-se como uma **filosofia de design.**

JBehave em 5 passos

1. Write story



JBehave em 5 passos

2. Map steps to Java

POJO

```
public class TraderSteps {
    private TradingService service;   // Injected
    private Stock stock; // Created

    @Given("a stock and a threshold of $threshold")
    public void aStock(double threshold) {
        stock = service.newStock("STK", threshold);
    }
    @When("the stock is traded at price $price")
    public void theStockIsTraded(double price) {
        stock.tradeAt(price);
    }
    @Then("the alert status is $status")
    public void theAlertStatusIs(String status) {
        assertThat(stock.getStatus().name(), equalTo(status));
    }
}
```

JBehave em 5 passos

3. Configure Stories

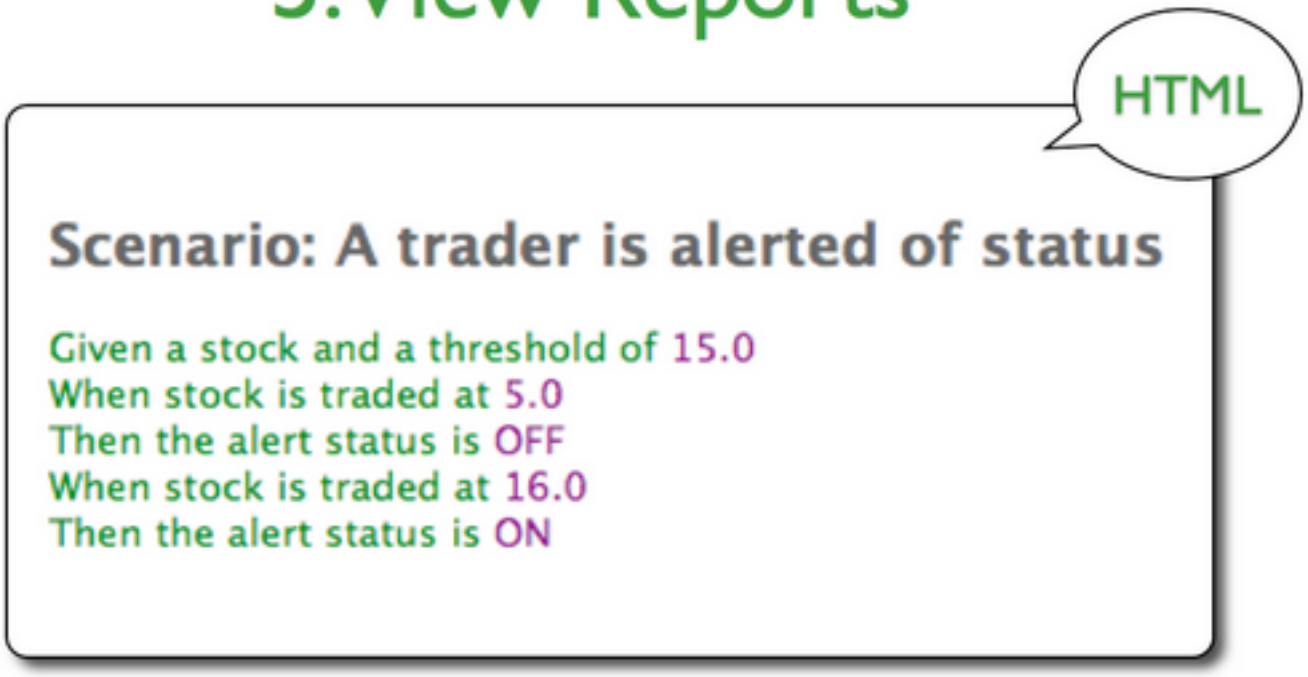
Only once

```
public class TraderStories extends JUnitStories {

    public Configuration configuration() {
        return new MostUsefulConfiguration()
            .useStoryLoader(new LoadFromClasspath(this.getClass()))
            .useStoryReporterBuilder(new StoryReporterBuilder()
                .withCodeLocation(codeLocationFromClass(this.getClass()))
                .withFormats(CONSOLE, TXT, HTML, XML));
    }

    public List<CandidateSteps> candidateSteps() {
        return new InstanceStepsFactory(configuration(),
              new TraderSteps(new TradingService())).createCandidateSteps();
    }

    protected List<String> storyPaths() {
        return new StoryFinder().findPaths(codeLocationFromClass(this.getClass()),
"**/*.story");
    }
}
```

JBehave em 5 passos

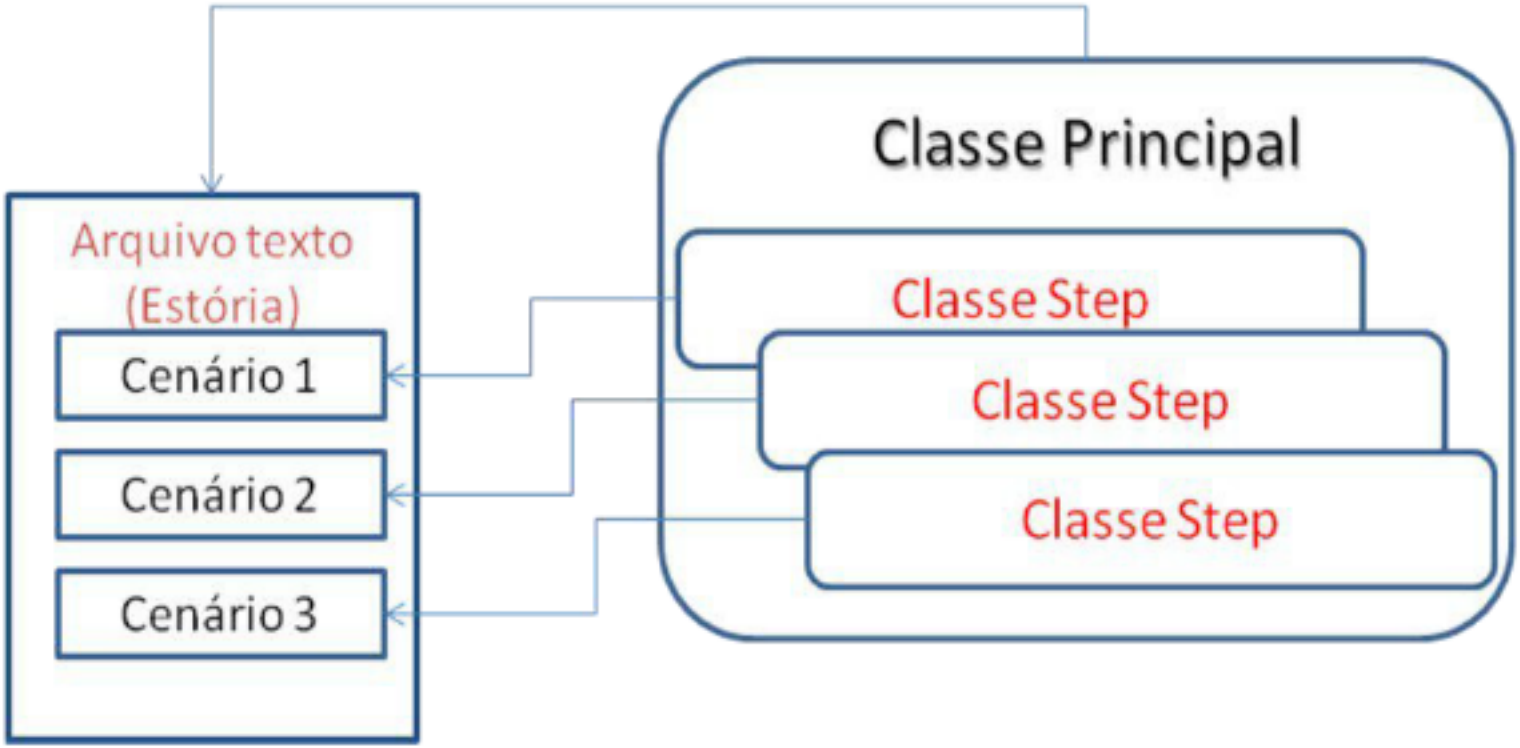
JBehave em 5 passos

5.View Reports

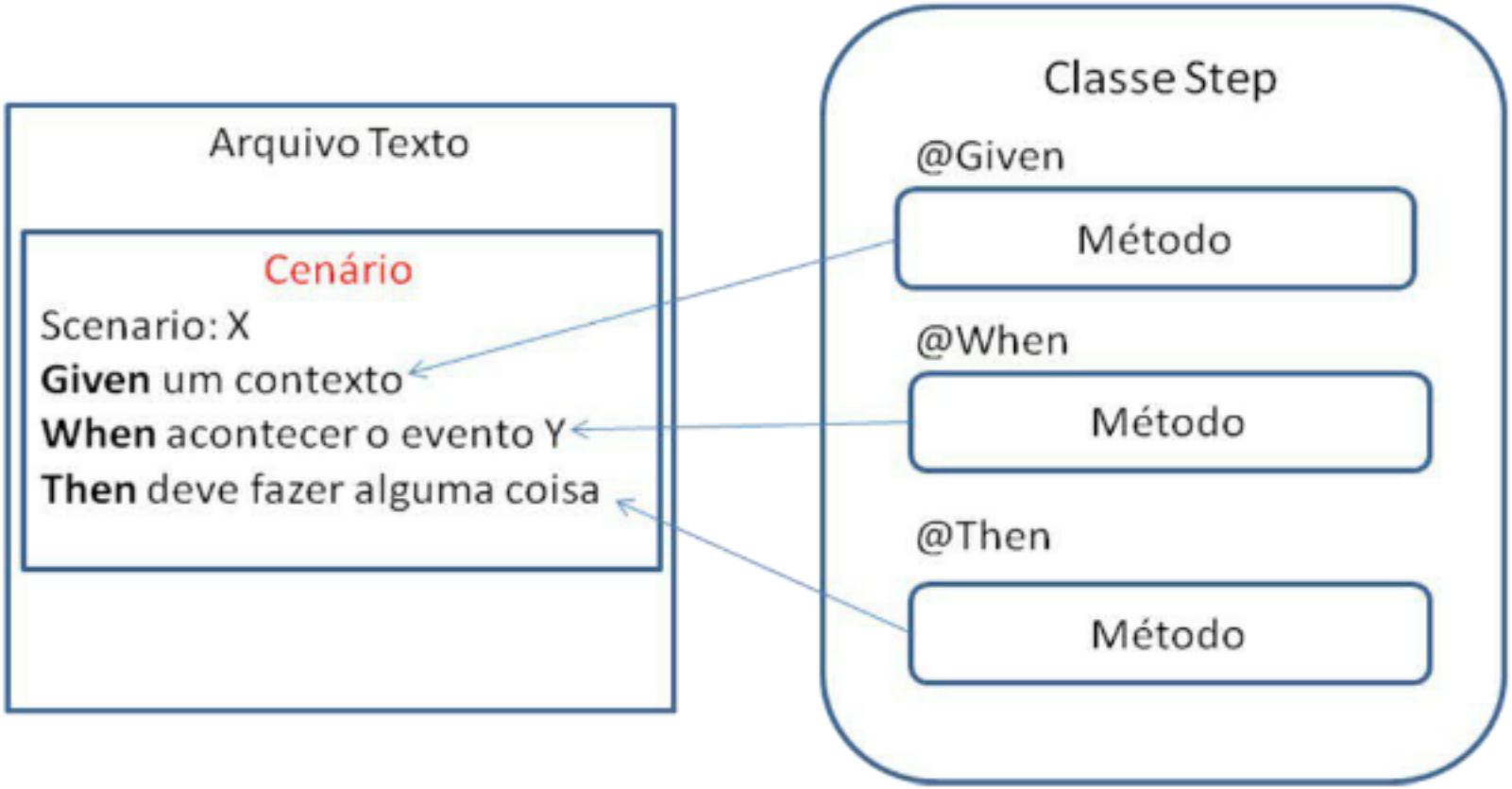


una

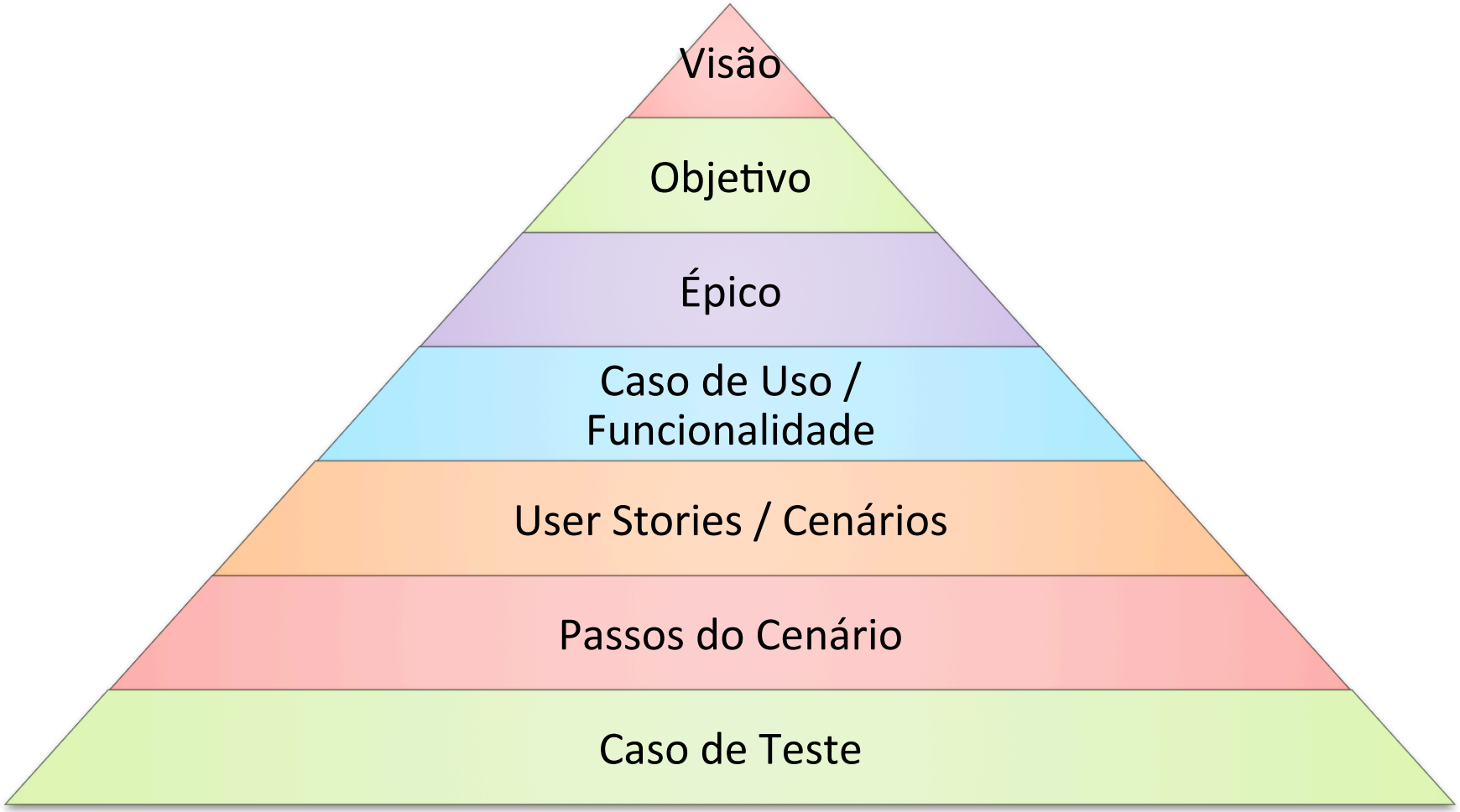
Como JBehave Funciona



Como JBehave Funciona



Pirâmide BDD



Sintaxe

Title Recadastramento de Senha
Narrative:
As a Usuario
I want recadastrar minha senha
So that continue tendo acesso ao sistema, com outra senha

User Story

Scenario: Validar nova senha
Given Um usuário de nome: <nome> e login: <login> e a senha: <senha>
When verifico se a senha é segura
Then Deve retornar a mensagem: <mensagem>

Definição do Cenário

Examples:
nome	login	senha	mensagem
Marcelo Zeferino	zeferino	1234	A senha deve conter ao menos 5 caracteres.
Kurt Cobain	kurt	12345	

Tabela de Exemplos