

# Behaviour Driven Development

Ruby, Cucumber, PivotalTracker

Ing. Josef Pospíšil,  
Ing. Martin Rychlovský  
GravaStar s.r.o.



<http://github.com/pepe/objekty2009>



# Kdo jsme

- Mladá softwarová společnost
- Katedra informačního inženýrství ČZU
- Průkopníci agilních přístupů
- Ruby evangelizátoři

# Obrázek na začátek

Projects ▾ **PIVOTAL TRACKER** Welcome, Josef Pospíšil

nuis Dashboard | Reports | My Profile | My Accounts | Help | Sign Out

Done Current Backlog Icebox View ▾ Actions ▾ Add Story

Velocity: 13  
3 iteration average

Current

- 18 5 Oct - Current Pts: 4 of 15  
▶ ★ = core OpenId authentication (JP)  
▶ ★ - forms, ui Add identity\_url to users profile page with edit (KR)  
▶ ★ - ui restyle pagination bar (BE) Deliver  
▶ ⚙ - forms, junior, ui Check english translation of error messages (KR)  
▶ ★ = Design a implement dashboard with added charts (JP)  
▶ 🐞 - forms, public Fix prefilling data from ARES to correct fields (JP)  
▶ 🐞 - core Fix change of state if invoice is payed. (JP)  
▶ ★ - core Create users list modul (JP)  
▶ ★ - Add balance of all statistics Finish

Backlog

- ▼ 19 12 Oct Pts: 11  
▶ ⚙ Add actuality about openid (BE)  
▶ ⚙ forms, ui Redesign login form to accomodate openid field (BE)  
▶ ⚙ core, rake Redone bootstrap with factories  
▶ 🎨 Last beta release Finish  
▶ ★ = test Add features for expenses Start  
▶ ★ - test Add features for divisions Start  
▶ ★ - test Add features for profile Start  
▶ ★ = test Add features for invoices Start  
▼ 20 19 Oct Pts: 5  
▶ ★ = test Add features for dashboard Start  
▶ ★ - test Add features for companies Start  
▶ ⚙ test Mock with in memory filesystem

Icebox

- ▶ ⚙ junior, test Add helper tests Start  
▶ ★ - think about colors of action buttons and error messages when something is missing  
▶ ★ - logout button is ugly Start

© 2009 Pivotal Labs / Policies



CSRUG

# Behaviour-Driven Development

- Nic úplně nového, jen další navazující fáze
  - Test Driven Development
  - Acceptance Test Driven Planning
  - Domain Driven Design



# Tři principy založené na aktivitách

- Business a IT používá stejný způsob odkazování
- Identifikovatelná a ověřitelná hodnota
- Snižující se hodnota analýzy, designu a plánování s velkým předstihem



# Stejný slovník

- Velmi specifický
- Malý rozsahem
- => Minimalizace nedorozumění
- Stejný pro bussiness, vývojáře, testery a analytiky



# Test Driven Development

- Delší historie
- Jedno z nejpronikavejších zlepšení vývoje software
- Ne všechny výhody spojené s testováním



# Životní cyklus učení

## TDD I. část

1. Testy na existující kód v JUnit, NUnit nebo specs
2. Uspokojení a zvýšený pocit důvěry v systém
3. Testy před kódem pro pouze nezbytný kód
4. Testy jako dokumentace existujícího kódu



# Životní cyklus učení

## TDD II. část

5. Psaní testů odkrývá API a je návrhovým procesem
6. Více specifikuje chování systému, než testuje funkce
7. Interakce mezi komponentami systému
  - mocking fundamentální součást TDD



# Chování systému

- Jen první dva body spojeny s testováním
- Hlavní částí se stává chování systému
- Zvyšuje se důvěra v systém
- Potvrzení chování systému
- Nomenklatura v termínech testování



# Cíle BDD

- Překlenout rozdílné pohledy na informační systémy
- Minimalizovat překážky mezi specifikací, návrhem, implementací a ověřením chování systému
- Inkrementální dodávky bussiness systémů
- Pomoci agilním týmům získat potřebnou rychlosť



# Minimum Ruby

- Dynamický, interpretovaný, imperativní, objektově orientovaný jazyk
- Několik programovacích paradigmat
  - objektově orientované (SmallTalk like)
  - funkcionální (Lisp like)
  - reflektivní
- Výborný jazyk pro vytváření DSL



# Cucumber

- čistě textové dokumenty
- spustitelné jako automatické testy
- jazyk se nazívá Gherkin
- business čitelný a doménově specifický
- popisuje chování software bez detailního popisu fungování a implementace
- dokumentace a automatizované testování



# Základní struktura

- hlavní rysy - features
- předpokady - background
- scenáře - scenarios
- tabulka příkladů - examples
- jednotlivé kroky - steps
- popis kroků - steps definition



## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

## Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

## Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

```
Given I have entered <input_1> into the calculator
And I have entered <input_2> into the calculator
When I press <button>
Then the result should be <output> on the screen
```

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>

Then the result should be <output> on the screen

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

### Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

### Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

## Feature: Addition

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

## Scenario Outline: Add two numbers

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

## Examples:

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

# Popis chování v čistém textu

```
# encoding: utf-8
require 'spec/expectations'
require 'lib/calculator'

Before do
  @calc = Calculator.new
end

Given /I have entered (\d+)/ into the calculator/ do |n|
  @calc.push n.to_i
end

When /I press (\w+)/ do |op|
  @result = @calc.send op
end

Then /the result should be (.*) on the screen/ do |result|
  @result.should == result.to_f
end
```

# Definice kroku v Ruby

```

Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

Scenario Outline: Add two numbers                               # features/addition.feature:6
  Given I have entered <input_1> into the calculator      # features/step_definitions/calculator_steps.rb:9
  And I have entered <input_2> into the calculator        # features/step_definitions/calculator_steps.rb:9
  When I press <button>                                     # features/step_definitions/calculator_steps.rb:13
  Then the result should be <output> on the screen          # features/step_definitions/calculator_steps.rb:17

Examples:
| input_1 | input_2 | button | output |
| 20     | 30      | add    | 50      |      uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'

undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
features/addition.feature:7:in `Given I have entered <input_1> into the calculator'
| 2      | 5      | add    | 7      |      uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'

undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
features/addition.feature:7:in `Given I have entered <input_1> into the calculator'
| 0      | 40     | add    | 40     |      uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'

undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
features/addition.feature:7:in `Given I have entered <input_1> into the calculator'

3 scenarios (3 failed)
12 steps (3 failed, 9 skipped)
0m0.011s

```

# Spuštění a selhání testu

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
```

```
Scenario Outline: Add two numbers
  Given I have entered <input_1> into the calculator # features/addition.feature:6
  And I have entered <input_2> into the calculator # features/step_definitions/calculator_steps.rb:9
  When I press <button> # features/step_definitions/calculator_steps.rb:9
  Then the result should be <output> on the screen # features/step_definitions/calculator_steps.rb:13
  # features/step_definitions/calculator_steps.rb:17
```

```
Examples:
```

input_1	input_2	button	output
20	30	add	50

```
uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'
```

```
undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
features/addition.feature:7:in `Given I have entered <input_1> into the calculator'
| 2      | 5      | add   | 7      |
uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'
```

```
undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
features/addition.feature:7:in `Given I have entered <input_1> into the calculator'
| 0      | 4      | add   | 40    |
uninitialized constant Calculator (NameError)
./features/step_definitions/calculator_steps.rb:6:in `Before'
```

```
undefined method `push' for nil:NilClass (NoMethodError)
./features/step_definitions/calculator_steps.rb:10:in `I have entered (\d+) into the calculator'
```

```
3 scenarios (3 failed)
12 steps (3 failed, 9 skipped)
0m0.011s
```

# Spuštění a selhání testu

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end

  def add
    @args.inject(0){|n,sum| sum+=n}
  end
end
```

Napsání dostatečného kódu

**Feature: Addition**

In order to avoid silly mistakes  
As a math idiot  
I want to be told the sum of two numbers

**Scenario Outline: Add two numbers**

Given I have entered <input\_1> into the calculator  
And I have entered <input\_2> into the calculator  
When I press <button>  
Then the result should be <output> on the screen

# features/addition.feature:6  
# features/step\_definitions/calculator\_steps.rb:9  
# features/step\_definitions/calculator\_steps.rb:9  
# features/step\_definitions/calculator\_steps.rb:13  
# features/step\_definitions/calculator\_steps.rb:17

**Examples:**

input_1	input_2	button	output
20	30	add	50
2	5	add	7
0	40	add	40

3 scenarios (3 passed)  
12 steps (12 passed)  
0m0.055s

# Úspěšné spuštění testu





# Opakování kroků

# PivotalTracker

- Story-based plánování projektů
- Spolupráce uvnitř týmu
- Okamžitá reakce na reálné změny
- Agilní metody vývoje
- Rozdělení na iterace
- Předpovědi rychlosti vývoje iterace



**PIVOTAL TRACKER**

Welcome, Josef Pospíšil  
[Dashboard](#) | [Reports](#) | [My Profile](#) | [My Accounts](#) | [Help](#) | [Sign Out](#)

**nuis**

Done Current Backlog Icebox View Actions Add Story

Velocity: 13  
 3 iteration average

**Current**

- 18 5 Oct - Current Pts: 4 of 15
  - core OpenId authentication (JP)
  - forms, ui Add identity\_url to users profile page with edit (KR)
  - ui restyle pagination bar (BE) Deliver
  - forms, junior, ui Check english translation of error messages (KR)
  - Design a implement dashboard with added charts (JP)
  - forms, public Fix Accept Reject prefilling data from ARES to correct fields (JP)
  - core Fix change of state if invoice is payed. (JP)
  - core Create users list Accept Reject modul (JP)
  - Add balance of all statistics Finish

**Backlog**

- ▼ 19 12 Oct Pts: 11
  - Add actuality about openid (BE)
  - forms, ui Redesign login form to accomodate openid field (BE)
  - core, rake Redone bootstrap with factories
  - Last beta release Finish
  - test Add features for expenses Start
  - test Add features for divisions Start
  - test Add features for profile Start
  - test Add features for invoices Start
- ▼ 20 19 Oct Pts: 5
  - test Add features for dashboard Start
  - test Add features for companies Start
  - test Mock with in memory filesystem Start

**Icebox**

- junior, test Add helper tests Start
- think about colors of action buttons and error messages when something is missing Start
- logout button is ugly Start

© 2009 Pivotal Labs / [Policies](#)

# PivotalTracker



CSRUG

<https://www.pivotaltracker.com>



**PIVOTAL TRACKER**

Welcome, Josef Pospíšil  
[Dashboard](#) | [Reports](#) | [My Profile](#) | [My Accounts](#) | [Help](#) | [Sign Out](#)

**nuis**

Done Current Backlog Icebox View Actions Add Story

Velocity: 13  
 3 iteration average

**Current**

- 18 5 Oct - Current Pts: 4 of 15
  - core OpenId authentication (JP)
  - forms, ui Add identity\_url to users profile page with edit (KR)
  - ui restyle pagination bar (BE) Deliver
  - forms, junior, ui Check english translation of error messages (KR)
  - Design a implement dashboard with added charts (JP)
  - forms, public Fix Accept Reject prefilling data from ARES to correct fields (JP)
  - core Fix change of state if invoice is payed. (JP)
  - core Create users list Accept Reject modul (JP)
  - Add balance of all statistics Finish

**Backlog**

- ▼ 19 12 Oct Pts: 11
  - Add actuality about openid (BE)
  - forms, ui Redesign login form to accomodate openid field (BE)
  - core, rake Redone bootstrap with factories
  - Last beta release Finish
  - test Add features for expenses Start
  - test Add features for divisions Start
  - test Add features for profile Start
  - test Add features for invoices Start
- ▼ 20 19 Oct Pts: 5
  - test Add features for dashboard Start
  - test Add features for companies Start
  - test Mock with in memory filesystem Start

**Icebox**

- junior, test Add helper tests Start
- think about colors of action buttons and error messages when something is missing Start
- logout button is ugly Start

© 2009 Pivotal Labs / [Policies](#)

# PivotalTracker



CSRUG

<https://www.pivotaltracker.com>



**PIVOTAL TRACKER**

Welcome, Josef Pospíšil  
[Dashboard](#) | [Reports](#) | [My Profile](#) | [My Accounts](#) | [Help](#) | [Sign Out](#)

**nuis**

Done Current Backlog Icebox View Actions Add Story

Velocity: 13  
 3 iteration average

**Current**

- 18 5 Oct - Current Pts: 4 of 15
  - core OpenId authentication (JP)
  - forms, ui Add identity\_url to users profile page with edit (KR)
  - ui restyle pagination bar (BE) Deliver
  - forms, junior, ui Check english translation of error messages (KR)
  - Design a implement dashboard with added charts (JP)
  - forms, public Fix Accept Reject prefilling data from ARES to correct fields (JP)
  - core Fix change of state if invoice is payed. (JP)
  - core Create users list Accept Reject modul (JP)
  - Add balance of all statistics Finish

**Backlog**

- ▼ 19 12 Oct Pts: 11
  - Add actuality about openid (BE)
  - forms, ui Redesign login form to accomodate openid field (BE)
  - core, rake Redone bootstrap with factories
  - Last beta release Finish
  - test Add features for expenses Start
  - test Add features for divisions Start
  - test Add features for profile Start
  - test Add features for invoices Start
- ▼ 20 19 Oct Pts: 5
  - test Add features for dashboard Start
  - test Add features for companies Start
  - test Mock with in memory filesystem Start

**Icebox**

- junior, test Add helper tests Start
- think about colors of action buttons and error messages when something is missing Start
- logout button is ugly Start

© 2009 Pivotal Labs / [Policies](#)

# PivotalTracker



CSRUG

<https://www.pivotaltracker.com>



The screenshot shows the Pivotal Tracker interface with three main boards:

- Current:** Contains stories for iteration 18 (5 Oct - Current). One story is marked as "Done" (core OpenId authentication (JP)). Other stories include "forms, ui Add identity\_url to users profile page with edit (KR)", "ui restyle pagination bar (BE)", "forms, junior, ui Check english translation of error messages (KR)", "Design a implement dashboard with added charts (JP)", "forms, public Fix prefilling data from ARES to correct fields (JP)", "core Fix change of state if invoice is payed. (JP)", "core Create users list modul (JP)", and "Add balance of all statistics".
- Backlog:** Contains stories for iteration 19 (12 Oct). One story is marked as "Done" (Last beta release). Other stories include "Add actuality about openid (BE)", "Redesign login form to accomodate openid field (BE)", "Redone bootstrap with factories", "test Add features for expenses", "test Add features for divisions", "test Add features for profile", "test Add features for invoices", "test Add features for dashboard", "test Add features for companies", and "test Mock with in memory filesystem".
- Icebox:** Contains stories for future planning. One story is marked as "Done" (junior, test Add helper tests).

Global navigation includes Projects, Dashboard, Reports, My Profile, My Accounts, Help, and Sign Out. The Velocity is listed as 13 (3 iteration average).

# PivotalTracker



CSRUG

<https://www.pivotaltracker.com>



**PIVOTAL TRACKER**

Welcome, Josef Pospíšil  
[Dashboard](#) | [Reports](#) | [My Profile](#) | [My Accounts](#) | [Help](#) | [Sign Out](#)

**nuis**

Done Current Backlog Icebox View Actions Add Story

Velocity: 13  
 3 iteration average

**Current**

- 18 5 Oct - Current Pts: 4 of 15
  - core OpenId authentication (JP)
  - forms, ui Add identity\_url to users profile page with edit (KR)
  - ui restyle pagination bar (BE) Deliver
  - forms, junior, ui Check english translation of error messages (KR)
  - Design a implement dashboard with added charts (JP) Finish
  - forms, public Fix prefilling data from ARES to correct fields (JP) Accept Reject
  - core Fix change of state if invoice is payed. (JP) Accept Reject
  - core Create users list modul (JP) Accept Reject
  - Add balance of all statistics Finish

**Backlog**

- ▼ 19 12 Oct Pts: 11
  - Add actuality about openid (BE)
  - forms, ui Redesign login form to accomodate openid field (BE)
  - core, rake Redone bootstrap with factories
  - Last beta release Finish
  - test Add features for expenses Start
  - test Add features for divisions Start
  - test Add features for profile Start
  - test Add features for invoices Start
- ▼ 20 19 Oct Pts: 5
  - test Add features for dashboard Start
  - test Add features for companies Start
  - test Mock with in memory filesystem Start

**Icebox**

- junior, test Add helper tests Start
- think about colors of action buttons and error messages when something is missing Start
- logout button is ugly Start

© 2009 Pivotal Labs / [Policies](#)

# PivotalTracker



CSRUG

<https://www.pivotaltracker.com>



**PIVOTAL TRACKER**

Welcome, Josef Pospíšil  
[Dashboard](#) | [Reports](#) | [My Profile](#) | [My Accounts](#) | [Help](#) | [Sign Out](#)

**nuis**

Done Current Backlog Icebox View Actions Add Story

Velocity: 13  
 3 iteration average

**Current**

- 18 5 Oct - Current Pts: 4 of 15
  - core OpenId authentication (JP)
  - forms, ui Add identity\_url to users profile page with edit (KR)
  - ui restyle pagination bar (BE) Deliver
  - forms, junior, ui Check english translation of error messages (KR)
  - Design a implement dashboard with added charts (JP)
  - forms, public Fix Accept Reject prefilling data from ARES to correct fields (JP)
  - core Fix change of state if invoice is payed. (JP)
  - core Create users list Accept Reject modul (JP)
  - Add balance of all statistics Finish

**Backlog**

- ▼ 19 12 Oct Pts: 11
  - Add actuality about openid (BE)
  - forms, ui Redesign login form to accomodate openid field (BE)
  - core, rake Redone bootstrap with factories
  - Last beta release Finish
  - test Add features for expenses Start
  - test Add features for divisions Start
  - test Add features for profile Start
  - test Add features for invoices Start
- ▼ 20 19 Oct Pts: 5
  - test Add features for dashboard Start
  - test Add features for companies Start
  - test Mock with in memory filesystem Start

**Icebox**

- junior, test Add helper tests Start
- think about colors of action buttons and error messages when something is missing Start
- logout button is ugly Start

© 2009 Pivotal Labs / [Policies](#)

# PivotalTracker



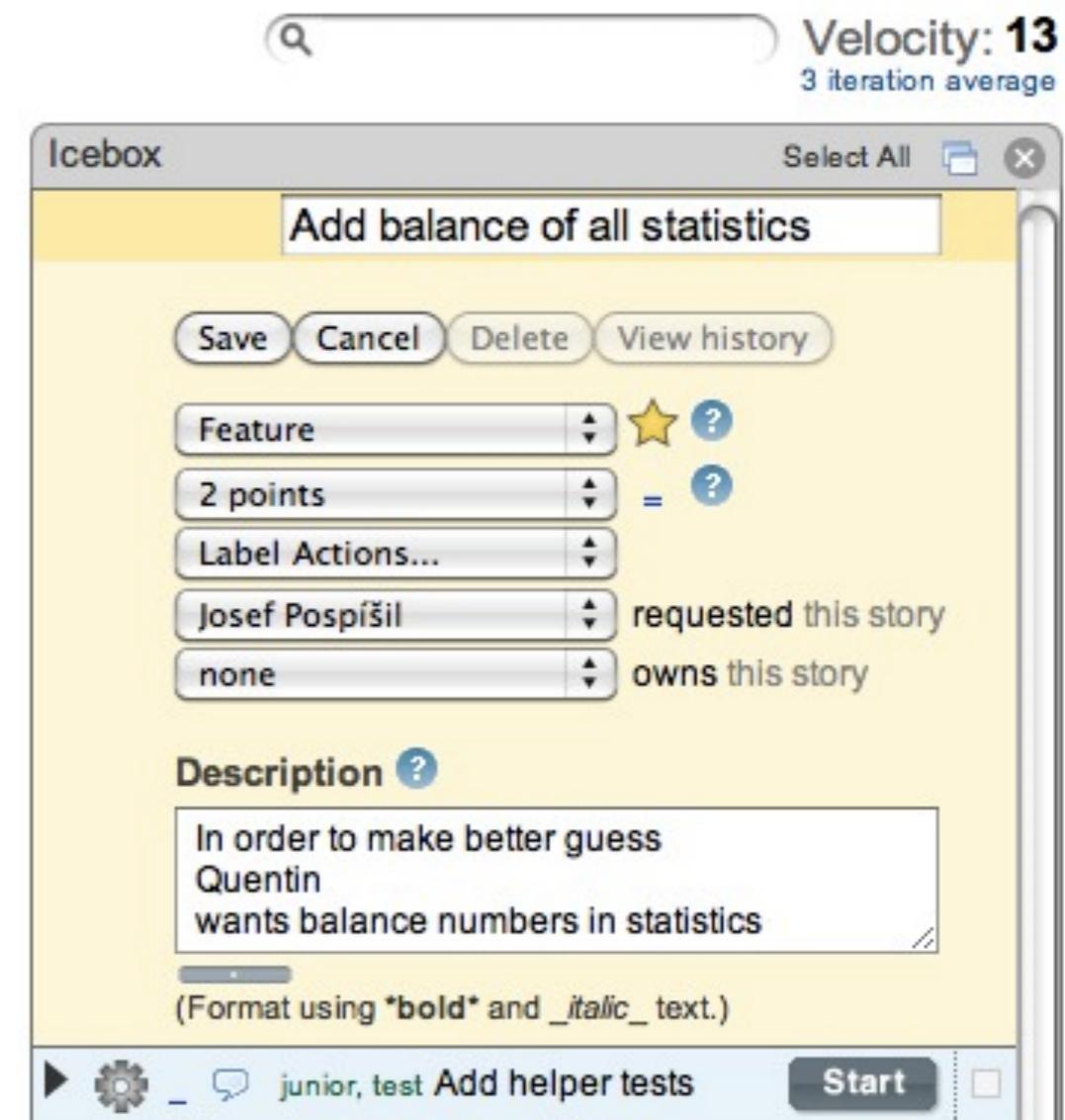
CSRUG

<https://www.pivotaltracker.com>



# Přidání story

- Jméno story
- Musí být feature ;)
- Obodování náročnosti
- Označení akce
- Formátované description
- Uložit a do backlogu



# Přínosy použití

- kombinace všech popisovaných postupů
- využití v systému edoktorand.czu.cz
- zlepšení komunikace se zákazníkem
- zlepšení zpětné vazby
- velmi dobré výsledky i ve vlastních projektech
- na něco jsem zapomněl?

# Konec

Jen a pouze této přednášky!