

Elgsena parentas programavimas. Priemimo testavimas - Cucumber.

**Judrusis programavimas Ruby
(Agile Development with Ruby)**

**Saulius Grigaitis
RubyOnRails LT**



- Naudotojo pasakojimais („User Stories“) paremtas testavimas:
 - Apimantis naudotojo pasakojimą
 - Suprantamas:
 - Užsakovui
 - Testuotojui
 - Programuotojui
 - Automatizuotas
 - Priėmimo (angl. „acceptance“ lygio)

Cucumber: Naudotojo pasakojimas (“User Story”) I

Feature: Books stock management

In order register new book

As books shop manager

I want to add register new book via WEB interface

Scenario: Register new book

Given I am on the new book page

When I fill in “Title” with “Paryžiaus Katedra”

And I fill in “Price” with “34”

And I press “Create”

Then I should see “Paryžiaus Katedra”

And I should see “34 LT”

Scenario: Delete book

Given there are 4 books

When I delete the first book

Then there should be 3 books left

Gherkin – kalba, kurią supranta “Cucumber”

- Suprantama verslo žmonėms
- Dalykinės srities kalba(angl. “DSL”)
- Leidžia apibūdinti programinės įrangos elgseną, nedetalizuojant kaip ta elgsena yra realizuota
- “Gyva” dokumentacija
- Automatizuoti testai
- Treetop gramatika

- Grupavimas – patogiu grupuoti pagal sritį, pvz:
 - books_steps.rb
 - authentication_steps.rb
- Given, When ir Then
 - Jų realizacija yra vienoda, tačiau rekomenduojama naudoti juos pagal reikšmę
 - Given – apima pradinis duomenis, kontekstą
 - When – apima veiksmus, kuriuos atlieka naudotojas
 - Then – apima rezultatų analizę

```
When /^I fill in "(.*)" with "(.*)"$/ do |field, value|  
  fills_in(field, :with => value)  
end
```

```
When /^I press "(.*)"$/ do |button|  
  clicks_button(button)  
end
```

```
Then /^I should see "(.*)"$/ do |text|  
  response.body.should =~ /#{text}/m  
end
```

```
Given /I am on the books list in admin panel/ do  
  visits "/admin/books"  
end
```

```
When /I delete the first book/ do  
  clicks_link "Destroy"  
end
```

```
Then /there should be (\d+) books left/ do |n|  
  Book.count.should == n.to_i  
  response.should have_tag("table tr", n.to_i + 1)  
end
```


Scenario Outline: Delete book

Given there are <initial_amount> books

When I delete the first book

Then there should be <after_amount> books left

Examples:

initial_amount	after_amount
100	99
1	0

Given the follow books exist:

title	price	
Paryžiaus katedrai	34	
Vargdieniai	28	

```
Given /the following books exist:/ do |books_table|
  books_table.hashes.each do |book|
    # bloką vykdys tiek kartų, kiek yra duomenų
    eilučių, pvz. pirmą kartą "book" bus {'title' =>
    'Paryžiaus katedra', 'price' => '34'}
  end
end
```

Cucumber: būsenos pernešimas žingsniuose

- `@variable` - bendas kintamasis žingsniuose
 - Labai paprasta naudotis
 - Didina testų sankibą!

- Palaiko apie 20 kalbų, tame tarpe ir lietuvių, pvz:

Sąvybė: Sudėtis

Norėdamas išvengti kvailų klaidų

Aš noriu, kad man pasakytų dviejų skaičių sumą

Scenarijus: dviejų skaičių sudėtis

Duota įvedžiau 50 į skaičiuotuvą

Ir įvedžiau 70 į skaičiuotuvą

Kai aš paspaudžiu "add"

Tada rezultatas ekrane turi būti 120

Ir rezultato klasė turi būti "Fixnum"

- JRuby ir Java – galima naudoti su įprasta Java
- IronRuby ir .Net – veikia net ir ten ;)
- FunFX and Flex – leidžia testuoti Adobe Flex programas

Cucumber – WEB programoms, parašytoms bet kokia kalba

- Naudojantys įprastas naršykles (gana lėti):
 - FireWatir – naudoja FF (GNU/Linux, Windows, OSX)
 - Watir – naudoja IE (Windows)
 - SafariWatir – naudoja Safari (OSX)
 - ChromeWatir – naudoja Google Chrome
- Naudojantis savo naršyklių variklius (greitesni):
 - Celerity – naudoja HtmlUnit, palaiko JavaScript !
 - Mechanize – nepalaiko JavaScript
- Nenaudojantys naršyklių (spartuoliai):
 - Webrat – standartinis Cucumber'yje

Cucumber - daug įvairių priedų

- DatabaseCleaner – DB būsenos atstatymui
- EmailSpec – el. laiškų testavimui
- FeaturesReport – sąvybių ataskaitų generavimas PDF formatu
- Testjour – išskirstytas testų vykdymas
- ...

<http://wiki.github.com/aslakhellesoy/cucumber/related-tools>

?