

Testovanie implementácie na platforme Ruby on Rails

Metodika

Lukáš Repka

1. Úvod

Účelom tejto metodiky je popis vytvárania testov pre aplikácie vyvíjane na platforme Ruby on Rails. Ako vývojové prostredie použijeme Netbeans [1] a na testovanie využijeme knižnicu Ruby Unit Testing Framework [2].

2. Použité pojmy

Netbeans – Vývojové prostredie na tvorbu aplikácií na platforme Ruby On Rails, dostupne na [1].

Ruby Unit Testing Framework – platforma na testovanie Ruby aplikácií [2].

File – po slovensky súbor.

New File – po slovensky nový súbor.

Categories – po slovensky kategórie.

File Types - po slovensky typy súborov.

Next – po slovensky ďalej.

Class – po slovensky trieda.

Host – po slovensky hostiteľ.

Adapter – po slovensky adaptér.

Database – po slovensky databáza.

Port - po slovensky port.

Username – po slovensky používateľské meno.

Password – po slovensky heslo.

Run – po slovensky spustiť.

Test Project – po slovensky otestovať projekt.

Test File - po slovensky otestovať súbor.

3. Použitá literatúra

1. Vývojové prostredie Netbeans. Dostupné na: <http://www.netbeans.org/> (13.11.2008).
2. Ruby Unit Testing Framework.
Dostupné na : <http://ruby-doc.org/stdlib/libdoc/test/unit/rdoc/classes/Test/Unit.html> (13.11.2008).

4. Manažment procesu tvorby testov

Táto kapitola popisuje manažment testovania implementácie. Popis postupu testovania. Popis rolí, ktoré v ňom vystupujú a ich zodpovednosť.

4.1 Proces tvorby testov a ich vykonávanie

Poradie	Krok	Účastníci	Kapitola
1.	Poskytnutie zdrojových kódov na testovanie	Projektový manažér	4.3.1
2.	Rozhodnutie, ktoré zdrojové kódy potrebujú testy	Manažér kvality	4.3.2
3.	Návrh testov	Návrhár testov	4.3.3
4.	Vytvorenie testov	Tester	4.3.4
5.	Spustenie testov	Tester	4.3.5
6.	Vyhodnotenie testov	Projektový manažér	4.3.6

4.2 Sumarizácia rolí a ich zodpovednosť

Rola	Zodpovednosť
Projektový manažér	<ul style="list-style-type: none">• Naplánovať, kedy bude prebiehať testovanie• Naplánovať spúšťanie testov• Posúdiť výstup testovania
Manažér kvality	<ul style="list-style-type: none">• Rozhodnúť, ktoré triedy a ktoré metódy sa budú testovať• Rozhodnúť do akej hĺbky bude prebiehať testovanie
Programátor	<ul style="list-style-type: none">• Napísať zdrojové kódu, ktorý sa bude testov• Oznámiť, keď sú zdrojové kódy pripravené na testovanie
Návrhár testov	<ul style="list-style-type: none">• Vytvoriť množinu testovaných vlastností pre každú testovanú metódu
Tester	<ul style="list-style-type: none">• Vytvoriť vstupy pre testované funkcie• Overiť správnosť týchto vstupov• Vytvoriť testy• Spustiť naplánované testy

4.3 Opis krokov

4.3.1 Poskytnutie zdrojových kódov na testovanie

Vstup : požiadavky na testovanie

Výstup : zdrojový kód pripravený na testovanie

Účastníci : projektový manažér, programátor

Projektový manažér na základe poznatkov od programátorov rozhodne kedy sa začne testovať napísaný zdrojový kód. Na základe tohto rozhodnutia sa sprístupnia zdrojové kódy ľuďom, ktorí budú vykonávať testy.

4.3.2 Rozhodnutie, ktoré zdrojové kódy potrebujú testy

Vstup : zdrojové kódy, pripravené na testovanie

Výstup : zdrojový kód, ktorý sa bude testovať

Účastníci : manažér kvality

Manažér kvality rozhodne, ktoré zdrojové kódy potrebujú testovanie. A na základe tohto rozhodnutia sa oddelia zdrojové kódy, ktoré sa budú testovať od tých, ktoré sa nebudú. Zdrojové kódy, ktoré sa budú testovať sa pošlú ďalej.

4.3.3 Návrh testov

Vstup : zdrojové kódy, ktoré sa budú testovať

Výstup : návrh testov

Účastníci : návrhár testov

Návrhár na základe požiadaviek na funkčnosť systému navrhne pre každú testovanú metódu, čo a ako sa bude na nej testovať. Aké typy vstupov sa použijú.

4.3.4 Vytvorenie testov

Vstup : navrhnuté testy

Výstup : vytvorené testy

Účastníci : tester

Tester na základe návrhu testov vytvorí množinu vstupov a k nim odpovedajúce výstupy. Potom spraví metódy, ktoré vyskúšajú jednotlivé vstupy a overia ich s predpokladanými výstupmi.

4.3.5 Spustenie testov

Vstup : vytvorené testy

Výstup : výstup testovania (overenie funkcionality kódu)

Účastníci : tester, projektový manažér

Na základe plánov projektového manažéra tester pripraví testy na spustenie a poskytne výstup testovania projektovému manažérovi na spracovanie.

4.3.6 Vyhodnotenie testov

Vstup : výstup testovania

Výstup : funkčný kód alebo nefunkčný kód

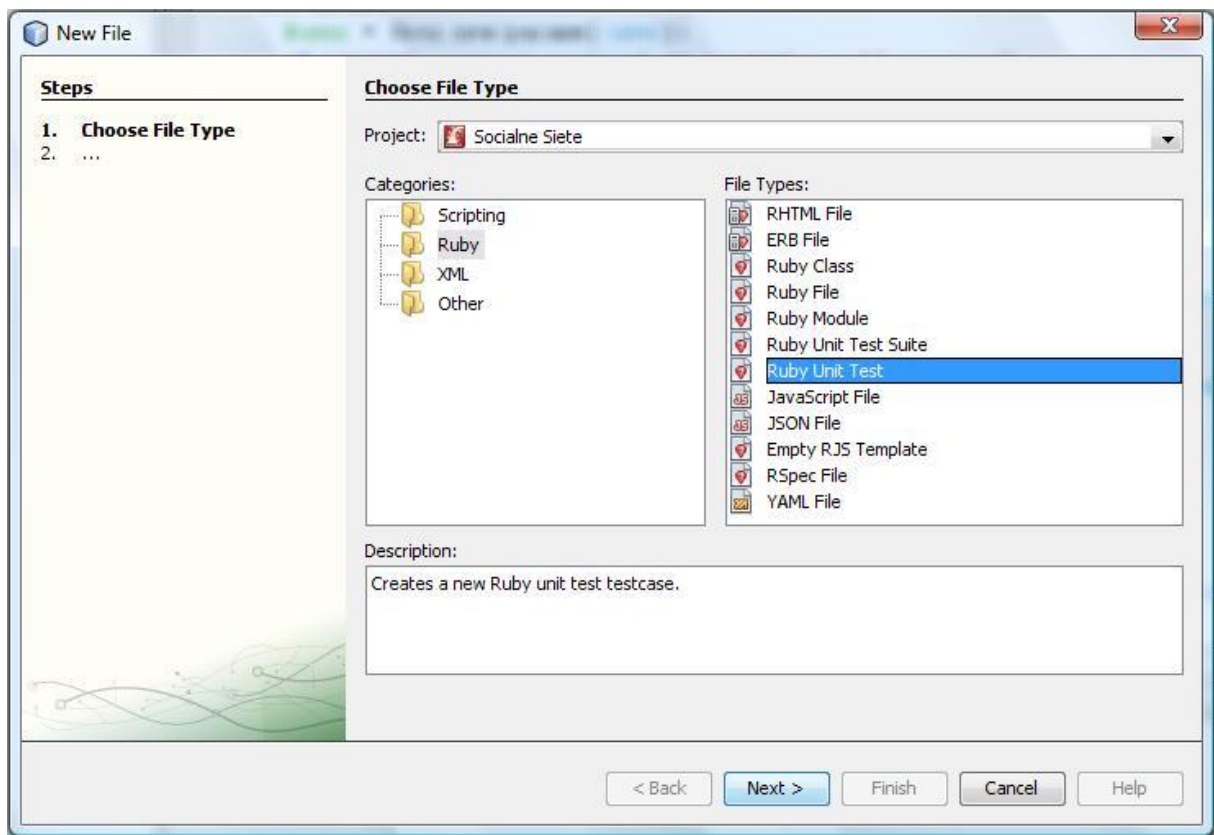
Účastníci : projektový manažér

Na základe výstupu testov projektový manažér rozhodne, či sú zdrojové kódy funkčné alebo nie. V prípade, že zdrojové kódy sú funkčné, odošlú sa na schválenie. V prípade, že zdrojové kódy nie sú funkčné, odošlú sa naspäť programátorom na opravenie.

5. Tvorba testov k modelom

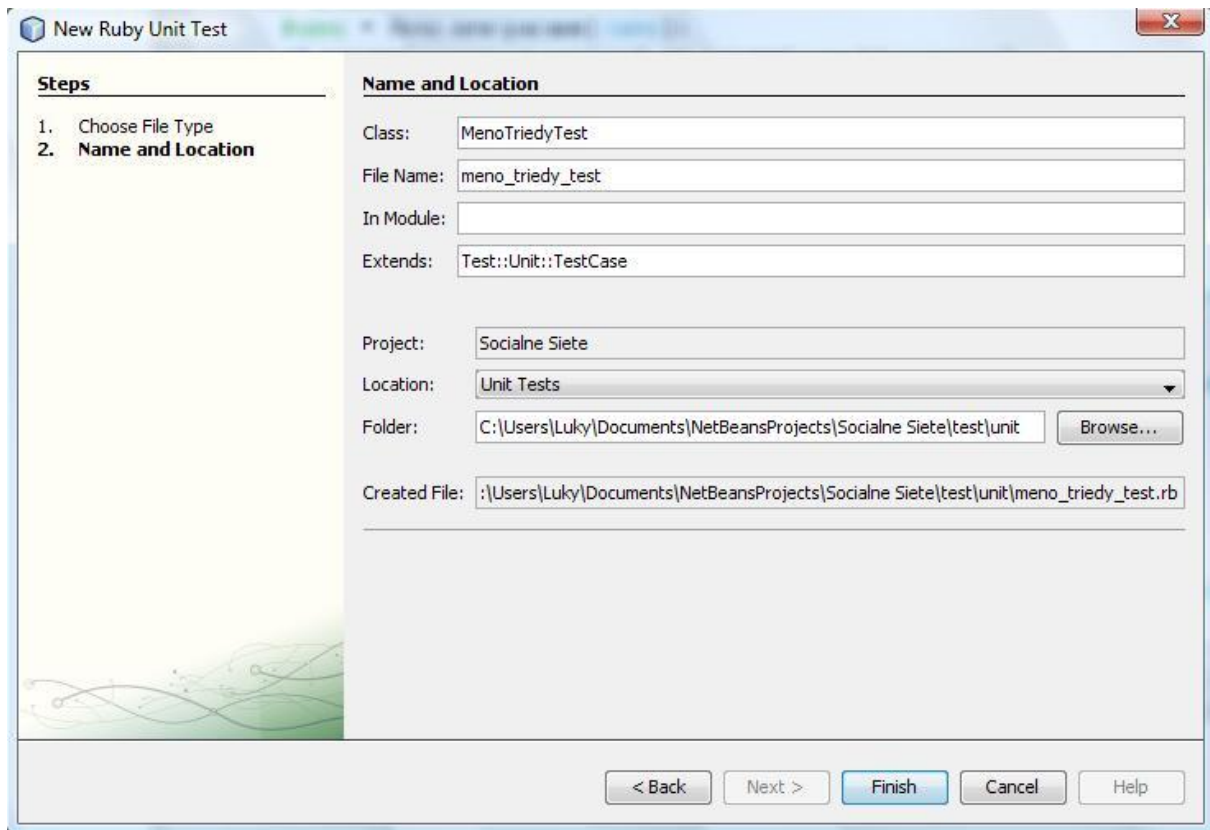
5.1 Vytvorenie súboru s testami

Na vytvorenie súboru s testami použijeme integrovaného pomocníka programovacieho prostredia Netbeans. Tohto pomocníka aktivujeme stlačením menu **File** a ďalej otvorením **New File**. Následne sa zobrazí okno, aké je zobrazené na obr. 1.



Obr. 1. Vytvorenie nového súboru s testami.

V menu **Categories** vyberieme zložku Ruby a v menu **File Types** vyberieme položku Ruby Unit Test. Následne stlačíme tlačítko **Next**. Po stlačení tlačítka sa nám zobrazí obrazovka zobrazená na obr.2.



Obr. 2 Vytvorenie nového súboru s testami, krok 2.

Na tejto obrazovke vyplníme v políčku **Class** meno triedy. Meno triedy vytvoríme tak, že zoberme meno triedy, ktorú ideme testovať a za toto meno doplníme slovo Test (napr. pre triedu UserModel sa bude testovacia trieda volať UserModelTest). Ďalej je potrebné vyplniť **File Name**, kde je potrebné vyplniť meno súboru. Meno súboru je zložené z jednotlivých slov v mene triedy začínajúcich malým písmenom a oddelených podčiarkovníkom (napr. pre triedu UserModelTest bude meno súboru user_model_test). V novo vytvorenom súbore sa objaví kostra zdrojového kódu, ktorá vypadá nasledovne :

```
$.unshift File.join(File.dirname(__FILE__), '..', 'lib')
```

```
require 'test/unit'
require 'user_model'
```

```
class UserModelTest < Test::Unit::TestCase
  def test_foo
    assert(false, 'Assertion was false.')
    flunk "TODO: Write test"
    # assert_equal("foo", bar)
  end
end
```

5.2 Vytvorenie testov

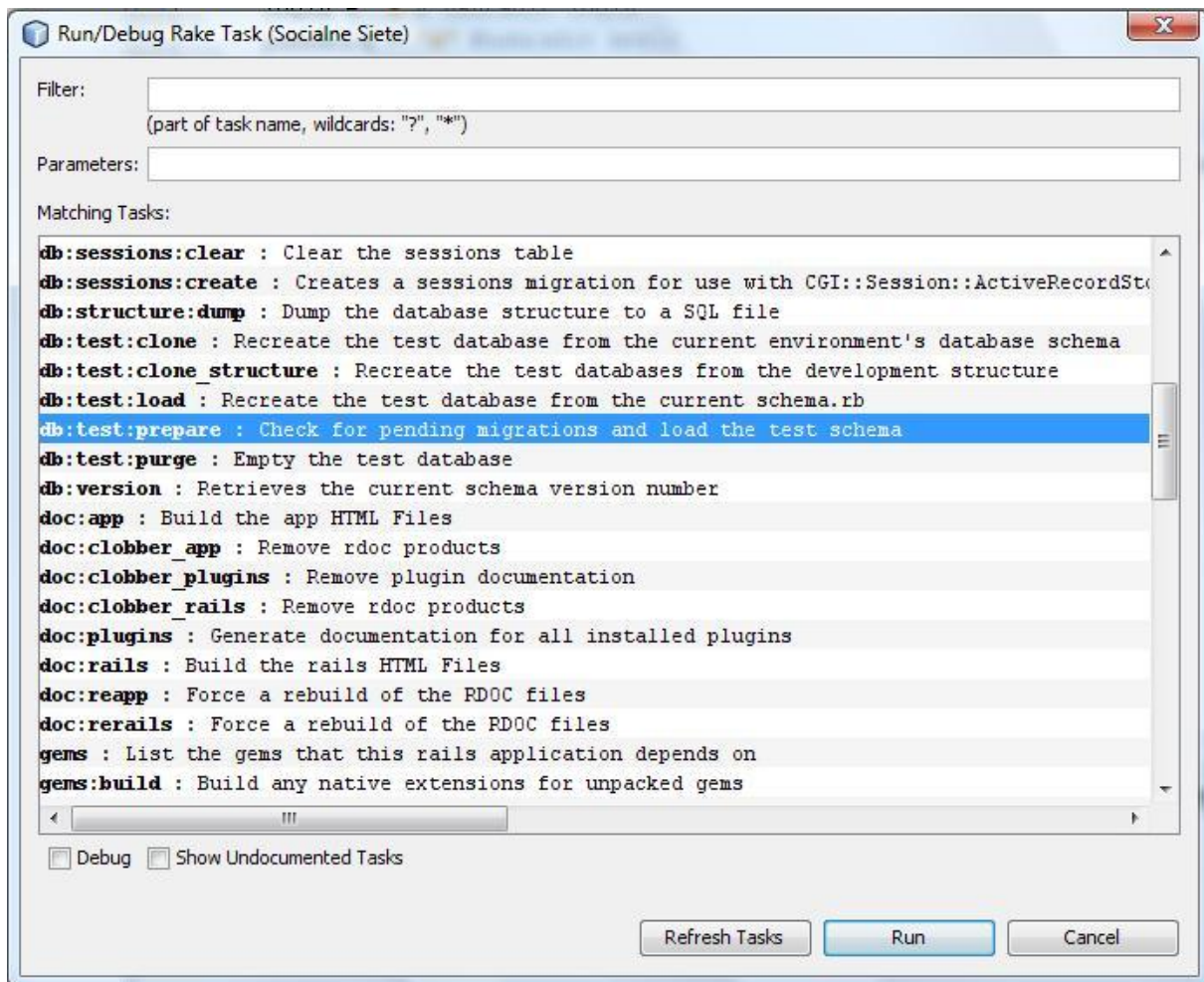
Následne je v tomto súbore nevyhnutne vymazať metódu *test_foo*, ktorá je automaticky vygenerovaná pri vytvorení súboru a nahradiť ju metódami, ktoré otestujú funkčnosť všetkých metód nachádzajúcich sa na modeli. Na otestovanie funkcionality metódy je potrebné vytvoriť metódy, ktoré otestujú ako sa správa testovaná metóda v prípadoch, keď je zavolaná so správnym vstupom, ale aj s nesprávnym vstupom. Na otestovanie správneho vstupu použijeme metódu *assert_equal*, pričom ako parametre jej zadáme očakávaný vstup a ako druhý parameter použijeme zavolanie testovanej metódy s správnymi parametrami. Na otestovanie nesprávnych parametrov použijeme metódu *assert_not_equal* a ako jej parametre zadáme očakávaný výstup a zavolanie testovanej metódy s nesprávnymi parametrami. Na otestovanie, či metóda vracia nil, použijeme metódu *assert_nil* a ako jej parameter zadáme zavolanie testovanej metódy s parametrami, s ktorými by mala metóda vracieť nil.

5.3 Spustenie testov

Pred prvým spustením testov je potrebné pripraviť databázu na testovanie. V prvom rade je potrebné nastaviť prístup k nej v konfiguračnom súbore *database.yml*, ktorý sa nachádza v zložke *Configuration*. Do tohto súboru je potrebné vyplniť údaje o IP adrese databázy v poli **host**, o type databázy v poli **adapter**, meno databázy v poli **database**, porte databázy v poli **port**, používateľské meno v poli **username** a používateľské heslo v poli **password**. Príklad naplnenia súboru je v kóde, ktorý nasleduje :

```
test:
  host: 127.0.0.1
  adapter: mysql
  database: meno_databazi
  port: 3306
  username: meno_pouzivatela
  password: heslo_pouzivatela
```

Následne je potrebné pripraviť tabuľky v databáze. Toto spravíme tak, že pravým kliknutím na projekt v ľavej časti obrazovky vývojového prostredia Netbeans otvoríme menu, kde vyberieme **Run/Debug Rake Task**. Potom sa nám zobrazí okno zobrazené na obr.3.



Obr.3 Príprava tabuliek v databáze pre testy

V otvorenom okne vyberieme položku **db:test:prepare** a stlačíme tlačítko **Run**. Po skončení procesu máme pripravenú databázu na spustenie testov.

Testy modelov spustíme tak, že v menu **Run** spustíme položku **Test Project**. Týmto spustíme všetky testy v projekte. Ak chceme spustiť iba jeden konkrétny test, vyberieme a otvoríme súbor s testom. Následné pravým tlačítkom myši klikneme v zdrojovom kóde, čím sa nám zobrazí menu, kde vyberieme položku **Test File**.