

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Radoslav Kontúr

Automatické nasadzovanie aplikácie pomocou nástroja Capistrano

Metodika

Študijný program: Informačné systémy

Predmet: Manažment projektov softvérových a informačných systémov

Školský rok: 2011/2012

# 1 Úvod

---

Metodika sa zaoberá nasadzovaním systémov, a hlavne automatizáciou tohto procesu prostredníctvom nástroja Capistrano. Táto metodika je určená hlavne pre vývojový tím. V kapitole 2 sú identifikované role aj s ich jednotlivými zodpovednosťami. Kapitola 3 sa venuje opisu procesu nasadzovania aplikácie a opisu jeho jednotlivých krokov. V kapitole 4 je podrobne rozpísaný postup konfigurácie automatického nasadzovania aplikácie pomocou nástroja Capistrano.

## Pojmy

**Tabuľka 1** Prehľad použitých pojmov

Build	Zostavenie aplikácie. Preklad zdrojového kódu do spustiteľnej podoby.
Nasadenie aplikácie/systému	Zahrňa všetky aktivity, ktorých cieľom je uvedenie aplikácie do používania.
Report	Správa o nasadení aplikácie.
Code freeze	Zákaz manipulácie so zdrojovým kódom uloženým na spoločnom úložisku.
Podporné nástroje	Nástroje slúžiace na podporu nejakého procesu. (Automatizácia nasadzovania, Automatizácia zostavovania aplikácie)

## Súvisiace metodiky

Metodika pre automatické zostavovanie aplikácie

## Použitá literatúra

Capistrano <https://github.com/capistrano/capistrano/wiki/>

## 2 Role

---

V tejto kapitole sú identifikované jednotlivé role aj s priradenými zodpovednosťami.

**Tabuľka 2** Prehľad rolí so zodpovednosťami

<b>Rola</b>	<b>Zodpovednosti</b>
Vedúci projektu	<ul style="list-style-type: none"><li>• Prideluje pracovníkov na nasadzovanie</li><li>• Pripravuje správu o nasadení</li><li>• Zodpovedný za odoslanie dokumentov zákazníkovi</li></ul>
Vývojár	<ul style="list-style-type: none"><li>• Vydáva “code freeze”</li><li>• Zostavuje aplikáciu</li><li>• Nasadzuje aplikáciu</li></ul>
Tester	<ul style="list-style-type: none"><li>• Testuje zmeny v aplikácii</li></ul>
Manažér kvality	<ul style="list-style-type: none"><li>• Kontroluje a upravuje dokumentáciu</li></ul>
Manažér podporných nástrojov	<ul style="list-style-type: none"><li>• Zabezpečuje inštaláciu a konfiguráciu podporných nástrojov: Capistrano, Jenkins</li></ul>
Manažér plánovania	<ul style="list-style-type: none"><li>• Plánuje opravu chýb</li></ul>

### 3 Proces nasadenia aplikácie

Celý proces nasadenia aplikácie môžeme opísať nasledovným spôsobom. Po tom, čo vedúci projektu zaregistruje požiadavku o nasadenie aplikácie prideli pracovníka zodpovedného za úspešné nasadenie aplikácie. Pridelený pracovník vydá zákaz odovzdávania verzií softvérových artefaktov („code freeze“) a následne zostaví aplikáciu (build). Po zostavení musí tester otestovať funkcionality danej aplikácie. Ak nájde chyby, tak prebehne proces opravy chýb. Po doladení chýb vedúci projektu zabezpečí napísanie podrobnej správy o nasadení, ktorá musí obsahovať podrobný opis všetkých zmien, ktoré boli zapracované. Následne prebehne kontrola a prípadne úprava dokumentu manažérom kvality. Manažér podporných nástrojov zabezpečí správnu konfiguráciu nástroja na automatické nasadzovanie aplikácie. Vývojár nasadí aplikáciu a vedúci projektu zabezpečí doručenie všetkých potrebných dokumentov zákazníkovi.

**Tabuľka 3** Prehľad krokov procesu nasadzovania aplikácie

Názov kroku	Zodpovedná rola	Číslo kapitoly, kde je krok opísaný
Pridelenie pracovníka zodpovedného za nasadenie	<i>vedúci projektu</i>	3.1
Zákaz odovzdávania verzií softvérových artefaktov	<i>vývojár</i>	3.2
Zostavenie aplikácie	<i>vývojár</i>	3.3
Kontrola chýb	<i>tester</i>	3.4
Proces opravy chýb	<i>vývojár</i> <i>manažér plánovania</i> <i>vedúci projektu</i>	3.5
Vypracovanie podrobnej správy o nasadení	<i>vedúci projektu</i>	3.6
Kontrola dokumentácie dodávky	<i>manažér kvality</i>	3.7
Konfigurácia automatického	<i>manažér podporných</i>	3.8

nasadzovania aplikácie	<i>nástrojov</i>	
Nasadenie aplikácie	<i>vývojár</i> <i>vedúci projektu</i>	3.9

### 3.1 Pridelenie pracovníka zodpovedného za nasadenie

Vstup: *požiadavka o nasadenie aplikácie*

Výstup: *report o nasadení systému doplnený o prideleného vývojára*

Zodpovedný: *vedúci projektu*

Vedúci projektu vyberie vývojára, ktorý bude mať na starosti úspešné nasadenie systému. Keďže pre nasadenie aplikácie nie je nutná žiadna špecializácia ani znalosť konkrétnej časti systému, tak vedúci projektu vyberá vývojára iba na základe vyťaženia a dostupnosti jednotlivých vývojárov.

### 3.2 Zákaz odovzdávania verzií softvérových artefaktov („code freeze“)

Vstup: *report o nasadení systému*

Výstup: *report o nasadení systému, správa o zákaze odovzdávani verzií softvérových artefaktov*

Zodpovedný: *vývojár*

Vývojár, ktorý ide nasadzovať aplikáciu, musí v prvom rade upovedomiť všetky osoby, ktoré sa priamo podieľajú na vývoji systému o tom, že nikto nemôže pridávať žiadne zmeny k aktuálnej verzii zdrojového kódu. Jedná sa o „code freeze“. Je to hlavne z toho dôvodu, aby po zostavení a kontrole aplikácie nenastali zmeny, o ktorých by daný vývojár nevedel. Tento krok neplatí ak ide vývojár nasadzovať verziu z lokálneho stroja.

### 3.3 Zostavenie aplikácie (build)

Vstup: *report o nasadení systému*

Výstup: *report o nasadení systému, zostavená aplikácia*

Zodpovedný: *vývojár*

Vývojár musí byť schopný pred nasadením úspešne zostaviť aplikáciu. V tomto kroku vývojár využíva podporné nástroje na automatické zostavovanie aplikácie, napr.: (Jenkins, Maven, Ant...).

### 3.4 Kontrola chýb

Vstup: *report o nasadení systému, zoznam zmien od posledného nasadenia*

Výstup: *report o nasadení systému doplnený o údaj, či je zostavená aplikácia v korektnom stave*

Zodpovedný: *tester*

Tester v tejto fáze otestuje aplikáciu podľa *zoznamu zmien od posledného nasadenia*, či všetky zmeny poskytujú správnu funkcionálnosť.

### **3.5 Proces opravy chýb**

Vstup: *aplikácia (zdrojový kód)*

Výstup: *aplikácia (opravený zdrojový kód)*

Zodpovedný: *vývojár, manažér plánovania, vedúci projektu*

Manažér plánovania naplánuje opravu chýb. Vedúci projektu rozdelí úlohy jednotlivým vývojárom, ktorí opravujú dané chyby. (Podrobnejšie sa týmto procesom zaoberajú iné metodiky, napr. Metodika pre správu zmien v prostredí PHP M2010/10/2)

### **3.6 Vypracovanie podrobnej správy o nasadení**

Vstup: *report o nasadení systému*

Výstup: *report o nasadení systému, podrobná správa o zmenách v aplikácii od posledného nasadenia*

Zodpovedný: *vedúci projektu*

Vedúci projektu, je zodpovedný za vypracovanie podrobnej správy o všetkých zmenách, ktoré nastali od posledného nasadenia systému (od poslednej dodávky).

### **3.7 Kontrola dokumentácie dodávky**

Vstup: *report o nasadení systému, podrobná správa o zmenách v aplikácii od posledného nasadenia*

Výstup: *skontrolované dokumenty*

Zodpovedný: *manažér kvality*

Manažér kvality skontroluje všetky dokumenty, a prípadne upraví tak aby boli čo najvyššej kvality.

### **3.8 Konfigurácia automatického nasadzovania aplikácie**

Vstup: *report o nasadení systému*

Výstup: *report o nasadení systému, správa o nakonfigurovaní automatického nasadzovania aplikácie*

Zodpovedný: *manažér podporných nástrojov*

Ak sa jedná o prvotnú konfiguráciu, tak manažér podporných nástrojov prevedie aj inštaláciu potrebných nástrojov. V konfigurácii je najdôležitejšie zvoliť cieľové prostredie, teda kam sa má systém nasadiť.

### **3.9 Nasadenie aplikácie**

Vstup: *správa o nakonfigurovaní automatického nasadzovania aplikácie*

Výstup: *správa o nasadení aplikácie*

Zodpovedný: *vývojár, vedúci projektu*

Ak je správne nakonfigurované automatické nasadzovanie aplikácie, tak vývojár zadá iba jednoduchý príkaz, ktorým sa aplikácia nasadí. Ak má byť aplikácia nasadená u zákazníka, tak vedúci projektu je navyše zodpovedný za odoslanie všetkých potrebných dokumentov zákazníkovi.

## 4 Automatické nasadenie aplikácie použitím nástroja Capistrano

---

1. Na lokálnom stroji nainštalujte (ak ešte nie je nainštalovaný) nástroj na automatické nasadzovanie aplikácie – Capistrano. Capistrano stačí nainštalovať ako gem:

```
gem install Capistrano
```

2. Ak inštalácia prebehla v poriadku, nakonfigurujte Capistrano nasledovným spôsobom:

- a. Na lokálnom disku sa nastavte do priečinka kde máte projekt aplikácie, ktorú chcete nasadiť.

Napr. `cd /home/rkontur/dev_webtex`

- b. Nainštalujte Capistrano pre váš projekt:

```
capify .
```

- c. Otvorte vygenerovaný súbor `deploy.rb`, ktorý sa nachádza v priečinku `config`. Použite textový editor:

Napr. `vim config/deploy.rb`

- d. Nastavte názov aplikácie, zmenením riadku

```
set :application, "názov_aplikacie"
```

- e. Nastavte umiestnenie repozitáru vášho nástroja pre kontrolu kódu vášho projektu.

```
set :repository, "adresa_repozitáru", kde
adresa_repozitáru:
git@adresa_repo/pouziv_meno/nazov_aplikacie.git
```

Napr.: `git@gitbus.fiit.stuba.sk:~xkukar/webtex/dev_webtex.git`

- f. Nastavte typ nástroja pre kontrolu kódu. Možné povolené hodnoty sú napr.: `:git`, `:mercurial`, `:subversion`...

Napr.: `set :scm, :git`

- g. Ďalej je možné nastaviť umiestnenie, kam na server sa má aplikácia nasadiť. Pre toto nastavenie slúži hodnota `deploy_to`. Ak túto hodnotu nenastavíte, tak Capistrano automaticky nasadí aplikáciu do `/u/apps/#{application}`

Napr.: `set :deploy_to, "/var/www/apps/#{application}"`



- h. Nasledujúce tri riadky sa týkajú nastavenia servera. Hodnoty „your web-server here“, „your app-server here“ a „your primary db-server here“.
- Nahraďte adresou domény servera, kam chcete nasadiť aplikáciu.

```
role :web, "147.175.159.172"

role :app, "147.175.159.172"

role :db, "147.175.159.172", :primary => true
```

- i. Zmažte riadok:

```
role :db, "your slave db-server here"
```

- j. Ďalšie nastavenia sa týkajú nastavenia používateľa Capistrana:

Nastavenie mena používateľa:

```
set :user, "user_name"
```

Nastavenie, či má byť aplikácia nasadená pod administrátorskými právami:

```
set :use_sudo, false (odporúča sa, aby bola vypnutá z
bezpečnostných dôvodov)
```

- k. Na záver treba pridať konfiguráciu Passengera:

```
namespace :passenger do
  desc "Restart Application"
  task :restart do
    run "touch #{current_path}/tmp/restart.txt"
  end
end

after :deploy, "passenger:restart"
```

3. Uložte a zatvorte konfiguračný súbor `deploy.rb`
4. Teraz je potrebné nakonfigurovať nástroj Passenger na strane servera. Otvorte konfiguračný súbor aplikačného servera (Apache2) a pridajte nasledujúce riadky:

```
LoadModule passenger_module /usr/lib/ruby/gems/1.8/gems/passenger-
1.0.1/ext/apache2/mod_passenger.so
```

```
SpawnServer /usr/lib/ruby/gems/1.8/gems/passenger-  
1.0.1/bin/passenger-spawn-serverRailsRuby /usr/bin/ruby1
```

5. Preto aby mohla byť aplikácia nasadená je nutné prvé vytvorenie adresárov a nastavenie ich práv na serveri. Na lokálnom stroji zadajte príkaz:

```
cap deploy:setup
```

6. Overenie, či je všetko pripravené na nasadenie aplikácie môžete previesť spustením príkazu

```
cap deploy:check
```

7. Aplikáciu nasadíte príkazom:

```
cap deploy
```