

# Spring Roo

MoroSystems, s.r.o.

Stanislav Hybášek

Consultant & Executive

[standa@morosystems.cz](mailto:standa@morosystems.cz)

[@standahybasek](#)

# MoroSystems, s.r.o.

- Zakázkový vývoj webových aplikací, informačních systémů a portálových řešení
- Outsourcing vývoje SW nad Java/J2EE
- Odborný weblog VsadNaJavu.cz
- Internetový startup Hráči.info
- Spring Framework, Hibernate, GWT, Apache Wicket, Liferay, ...

# MoroSystems reference

**WINCOR**  
**NIXDORF**

**DATALAN**  
IT WORKS



**lundegård**

Internet  
**BillBoard**

**creative.cz**  
where pixels make love

 **OldanyGroup**  
your smart way to IT



**oxyonline**

**PRAGUE DESIGN**  
WEB DESIGN, E-BUSINESS, CORPORATE IDENTITY

**Inventive Software**

# Poslání Roo

Zvýšit produktivitu java (JEE) vývoje

<https://jira.springframework.org/browse/ROO-1>

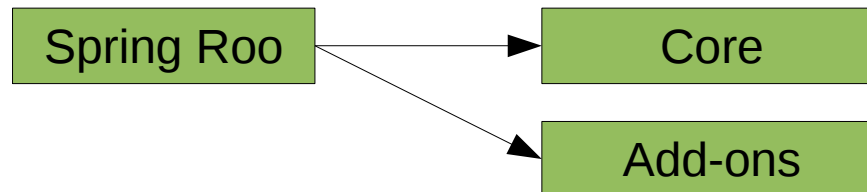
# Roo je:

- Nástroj pro usnadnění vývoje JEE aplikací
- Generátor kódu
- Rozšiřitelný
- Snadno použitelný
- „a little genie who sits in the background and handles the things I don't want to worry about“  
*Ben Alex, Founder and Project Lead – Spring Roo*

# Roo je development tool

- NENÍ plugin do IDE
- NENÍ knihovna pro použití v projektu
- NENÍ k dispozici za běhu (development time only)

# Roo je development tool



## Příkazová řádka (shell):

```
standa@standadell:~/devel/MyProject$ /opt/springsource/roo-1.1.0.RC1/bin/roo.sh
```



```
1.1.0.RC1 [rev 51cb976]
```

```
Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
```

```
roo> █
```

# Roo je generátor kódu

- aktivní:

Roo generuje soubory (.java, .xml, .properties...) na základě příkazu

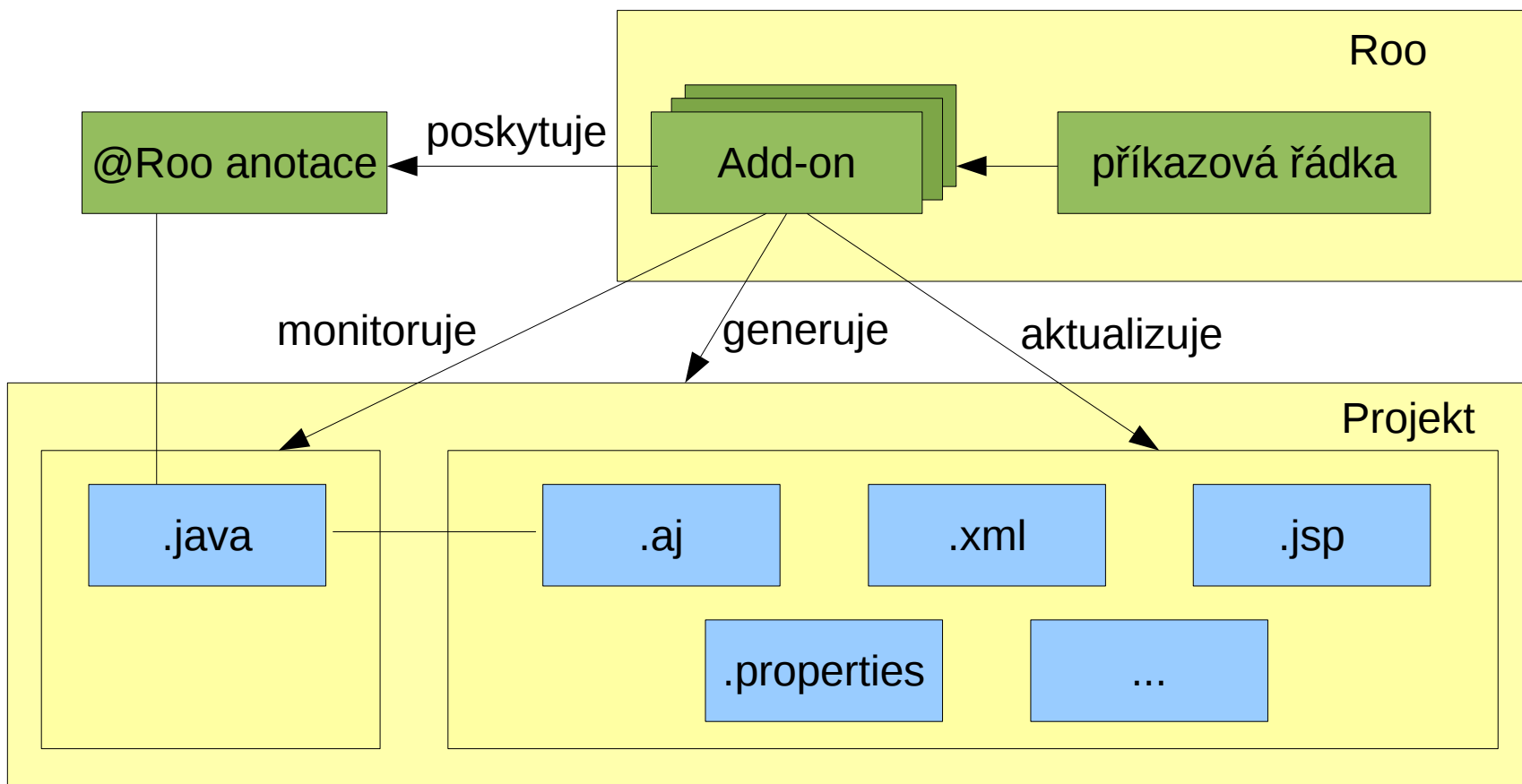
- pasivní:

Roo monitoruje (metadata - @Roo\*)

Roo generuje a upravuje soubory soubory (.aj, .jsp – nikdy .java) jako reakce na změny v projektu

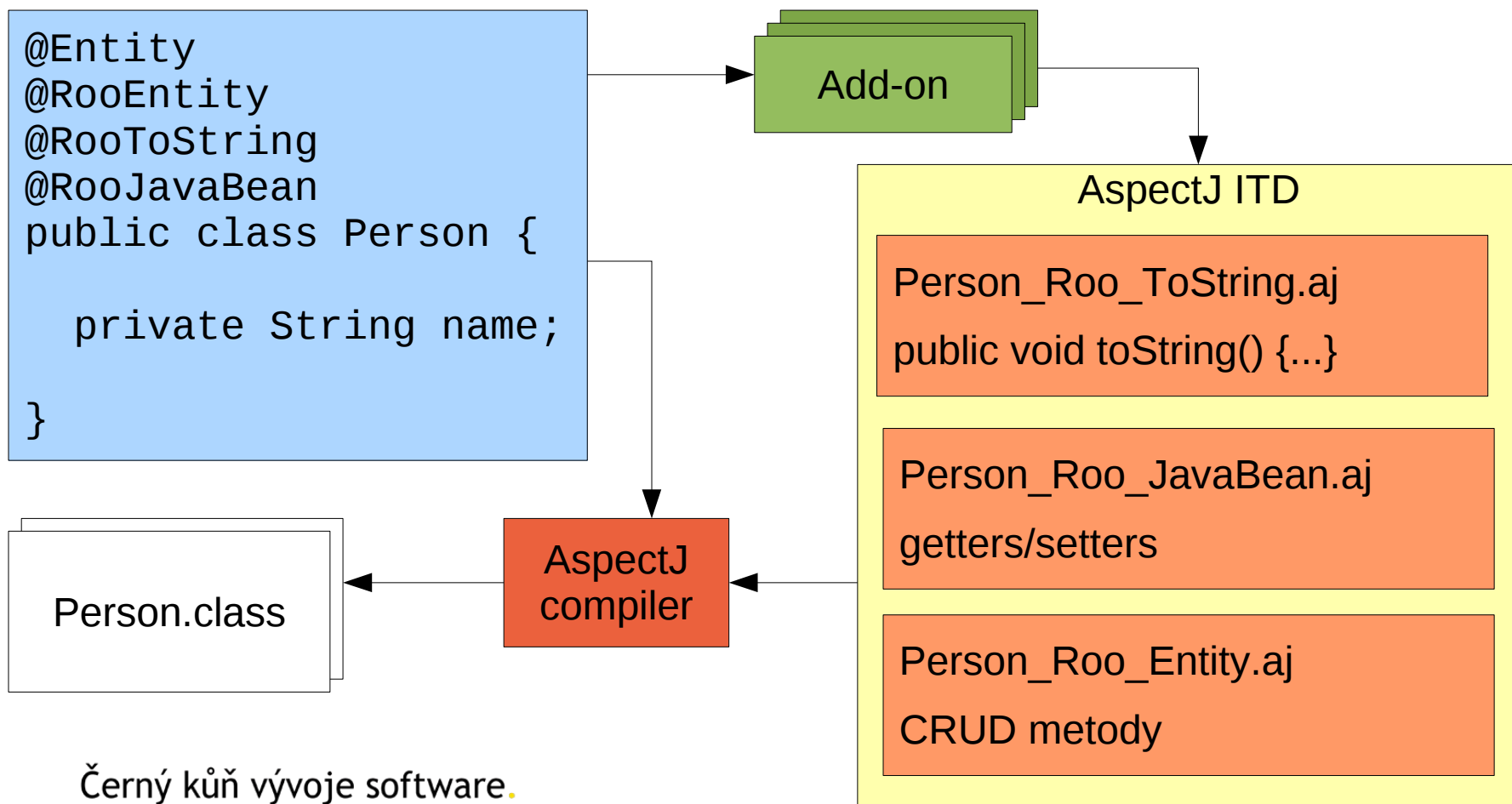


# Jak to funguje



Černý kůň vývoje software.

# Jak to funguje



# Roo je rozšířitelný

- Tvorba vlastních Add-ons

```
roo> addon create advanced --topLevelPackage cz.jug.roo.addons.myaddon
Created /home/standa/devel/MyProject/pom.xml
Created /home/standa/devel/MyProject/readme.txt
Created /home/standa/devel/MyProject/legal
Created /home/standa/devel/MyProject/legal/LICENSE.TXT
Created SRC_MAIN_JAVA
Created SRC_MAIN_RESOURCES
Created SRC_TEST_JAVA
Created SRC_TEST_RESOURCES
Created SRC_MAIN_WEBAPP
Created SRC_MAIN_RESOURCES/META-INF/spring
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/MyaddonCommands.java
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/MyaddonOperations.java
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/MyaddonOperationsImpl.java
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/MyaddonMetadata.java
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/MyaddonMetadataProvider.java
Created SRC_MAIN_JAVA/cz/jug/roo/addons/myaddon/RooMyaddon.java
Created ROOT/src/main/assembly
Created ROOT/src/main/assembly/assembly.xml
Created SRC_MAIN_RESOURCES/cz/jug/roo/addons/myaddon
Created SRC_MAIN_RESOURCES/cz/jug/roo/addons/myaddon/configuration.xml
cz.jug.roo.addons.myaddon roo> █
```

# Roo je rozšířitelný

- @RooMyAddon

```
package cz.jug.roo.addons.myaddon;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 * Trigger annotation
 * @since 1.1
 */
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.SOURCE)
public @interface RooMyaddon {
}
```

# Roo je rozšířitelný

pro shell:

- MyaddonCommands
- MyaddonOperations
- MyaddonOperationsImpl

pro práci s anotací:

- MyaddonMetadata
- MyaddonMetadataProvider

# Roo je rozšířitelný

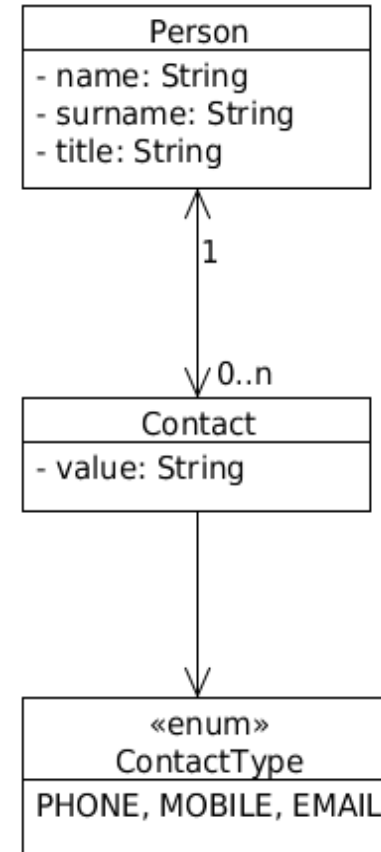
- Špatná dokumentace (neúplná, chybí)
- open-source, tedy zdrojové kódy
- komunita
  - fórum (<http://forum.springsource.org/>)
  - helpdesk (<https://jira.springframework.org/>)

# Verze Roo

- 1.0.0 (12/2009)
- náš projekt začínal v květnu 2010 –  
verze 1.1.0.M1 (5/2010)
  - první podpora gwt
- 1.1.0.M2 (7/2010)
- 1.1.0.M3 (8/2010)
- současná verze: 1.1.0.RC1 (10/2010)

# Ukázka

- vytvoření aplikace
- datový model (via shell)
- logování
- spring mvc view
- spring security
- spuštění aplikace
- gwt view
- import do IDE





# Ukázka

- log.roo

```
// Spring Roo 1.1.0.RC1 [rev 51cb976] log opened at 2010-10-25 18:22:40
project --topLevelPackage cz.jug.springroo.contactbook
persistence setup --provider HIBERNATE --database HYPERSONIC_IN_MEMORY
entity --class ~.entities.Person
field string --fieldName name --notNull
field string --fieldName surname --notNull
field string --fieldName title
entity --class Contact
field string --fieldName myValue --notNull
field reference --fieldName person --type Person
focus --class Person
field set --fieldName contacts --element Contact --cardinality ONE_TO_MANY --mappedBy person
enum type --class ContactType
enum constant --name PHONE
enum constant --name MOBILE
enum constant --name EMAIL
focus --class Contact
field enum --fieldName type --type ContactType --notNull --enumType STRING
logging setup --level INFO
logging setup --level DEBUG --package PROJECT
controller all --package ~.web
```

# Postřehy z používání

- STS
- výkonnostní problémy (AspectJ),  
velká paměťová náročnost
- Roo & svn
- potřeba společného předka?

# Postřehy z používání

- abstraktní třída musí implementovat všechny metody z rozhraní (buť abstraktně)
- Roo je teprve na začátku
  - (např. @RooToString v M2)
- ustálené použití

# Odkazy

<http://www.springsource.org/roo>

<https://jira.springframework.org/>

<https://jira.springframework.org/browse/ROO-1>

<http://forum.springsource.org/>