



# Extreme Java Productivity with Spring Roo and Spring 3.0

Rod Johnson

# Agenda



- 
- Motivation
  - Improving JVM developer productivity
  - Getting started with Roo
  - Roo design principles
  - Roo application architecture – Hands on
  - Roo Roadmap
- 
- Answers to frequently asked questions

# Motivation: We Have Work to Do



- Enterprise Java has a perception problem
- Everyone likes to kick sand in our face
- And we make it *easy* for them
  - Enterprise Java productivity *is* a soft target

**THE INSULT THAT MADE A MAN OUT OF "MAC"**

HEY! QUIT KICKING THAT SAND IN OUR FACES! THAT SAND IS THE WORSE NUISANCE ON THE BEACH!

LISTEN HERE, I'D SPARE YOUR PACE - ONLY YOU'RE SO BONY! YOU MIGHT GET UP AND BLOW AWAY!

THE BIG BULLY! I'LL GET EVEN SOME DAY!

OH DON'T LET IT BURN YOU LITTLE BOY!

DREN IT! I'M SICK AND TYPED OF BEING A SCARDLOW! CHARLES ATLAS SAYS HE CAN GIVE ME A REAL BODY. ALL RIGHT! I'LL GAMBLE A SHIRT AND GET THE FREE BOOK!

BOY! IF DON'T TAKE ATLAS' LONG TO DO THIS FOR ME! WHAT MUSCLES! THAT BULLY WON'T BITE ME AGAIN!

WHAT? YOU HERE AGAIN? HERE'S SOMETHING I OWE YOU!

OH MAC! YOU ARE A REAL MAN AFTER ALL!

**HERO OF THE BEACH**

HE'S ALREADY FAMOUS FOR IT!

**Let Me PROVE I Can Make YOU A NEW MAN!**

ARE you "fed up" with seeing the A Sucker walk off with the best of everything? Sick and tired of being left out, kicked or bled? - only DYNAMITE TABLETS! I know just how you feel. Because I myself was once a pale 110-pound "weak" - but I was an outcast of my society from that I decided to try them in a vain wish - The Secret of How I Got My Build.

Then I discovered a wonderful way to develop my body. That, it would wonder for me - changed me from the average "weak" I was at 110, into "The World's Most Perfectly Developed Man." And I can build up YOUR body the very same natural way - without weights, springs or machines. Only 10 minutes a day of pleasant practice - in the privacy of your room.

My "Dynamite Tablets" method has already helped thousands of other fellows become real heroes in double -

quick time. Let it help YOU! - Get next month or next year - but RIGHT NOW!

**"Dynamite Tablets"**

**Builds Muscles FAST!**

If you're like I was, you want a powerful, muscular, well-proportioned build you can be proud of any time, anywhere. You want the "hard-earned" type of physique that makes you stand out at the beach - the kind that makes other fellows green with envy.

**Mail Coupon Now for My 32-Page Illustrated Book**

Getting the coupon now for the handsome build in your life. I'll send you a copy of my new book "How to Build a NEW MAN." You know what you'll get! My complete, most complete picture plan for building your body that way! Mail coupon NOW! CHARLES ATLAS, Dept. 118, 112 E. 23rd St., New York, N.Y. 10010.

**CHARLES ATLAS, Dept. 118**  
112 East 23rd St., New York, N.Y. 10010

These "Dynamite Tablets" - shown at the end of Body of a Hero!

<input type="checkbox"/> Greater Size and Strength	<input type="checkbox"/> More Energy and Stamina
<input type="checkbox"/> Increased Blood Circulation	<input type="checkbox"/> More Muscular Progression
<input type="checkbox"/> Greater Appetite	<input type="checkbox"/> More Perfect Proportion
<input type="checkbox"/> Greater Will and Love	<input type="checkbox"/> More Self-Confidence

Send at once to me like:

I enclose \$1.00. Please send me a copy of your famous 32-page illustrated book "How to Build a NEW MAN." I'll send you a copy of my new book "How to Build a NEW MAN." You know what you'll get! My complete, most complete picture plan for building your body that way! Mail coupon NOW! CHARLES ATLAS, Dept. 118, 112 E. 23rd St., New York, N.Y. 10010.

Dear Name: \_\_\_\_\_

Address: \_\_\_\_\_

City: \_\_\_\_\_ State: \_\_\_\_\_

in Spring, Charles Atlas, is a trademark of Charles Atlas, Inc.

# Would-be Competitors



*Gutmans hopes that Zend can convert individuals [to PHP] who think Java has become bloated and organizations struggling to find affordable Java Enterprise Edition (Java EE) programmers*

Andi Gutmans, Zend CEO



*Why Anything and Everything Related to Java [Web] Programming and Open-Source Frameworks Sucks Major Ass in Comparison to Anything and Everything Related to Ruby and Rails.*

Obie Fernandez, Rails guy

# We Must be Honest: Some of the Criticism is Deserved: *It's too Hard to Get Started on the JVM*

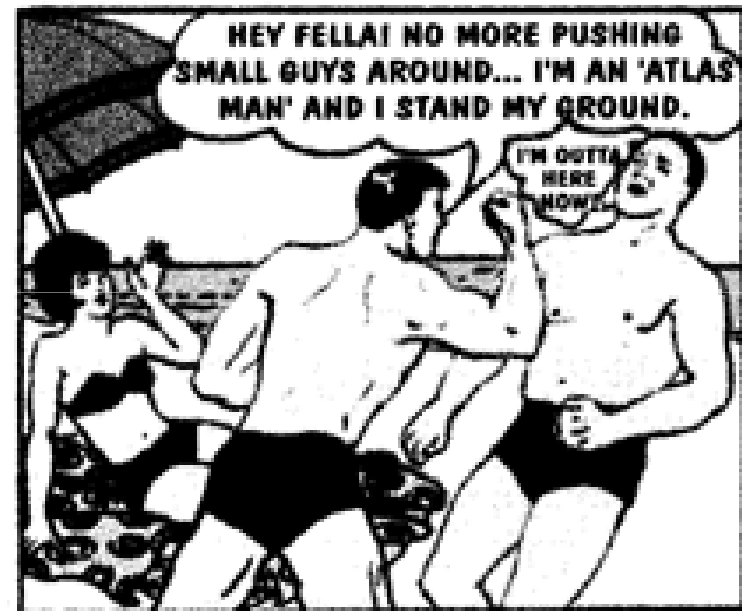
---



- Modern enterprise Java is way better than 5 years ago
- *But...it's still too hard* to...
  - Start a new Java project
  - Obtain and integrate all the necessary software
- Too much of our time is spent doing things that add too little value
  - We have great building blocks, but we need to improve the experience

# Time to Step Up and Fight Back

- Don't know about you, but *I'm as mad as hell and I'm not going to take it any more*
- We know the JVM is right for enterprise problems
- Time to show Rails/Django/PHP etc. that the JVM can compete on productivity



# Two Key Problems



No one owns  
the whole  
problem

- Non JVM technologies have demonstrated a greater focus on developer productivity, with good results
- Java "Productivity" solutions typically dead ends like MDA or bloatware from old world vendors

It's too hard  
to get started  
with projects  
on the JVM

- It's too easy to concentrate on hard problems
- We forget how much productivity is wasted from trivial problems

# Key Problem: Vendor/project fragmentation

---



- Technical problem - Different sources for:
  - Tooling
  - Build solution
  - RAD solution
  - Frameworks and libraries
  - Servers
- *No joined up thinking*
- *No one owns the whole productivity experience*



# The Result Looks like This



- 1970s American Motors Gremlin
- What happens when something is put together out of separate pieces

# Industry-Wide Trend Towards Joined-up Solutions

---



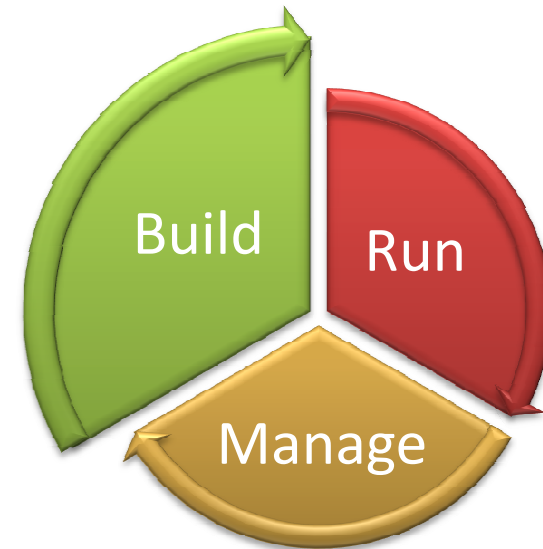
- It's great to have choice, but it's often better to use an opinionated approach to streamline things
- Trend towards joined up:
  - Stacks (vertical integration)
  - Experience(Rails)
- SpringSource is focused on tackling the whole problem
- Coherent vision for all stages of the lifecycle
  - Build/Run/Manage
- Cloud deployment option integrated with developer tooling, frameworks natural next step

# Two SpringSource Solutions For JVM Developer Productivity, Targeted at Different Audiences

---



- Grails
  - Uses the power of the #1 dynamic language on the JVM
  - Large and rapidly growing community
- Roo
  - Use intelligent code generation to provide improved productivity in Java

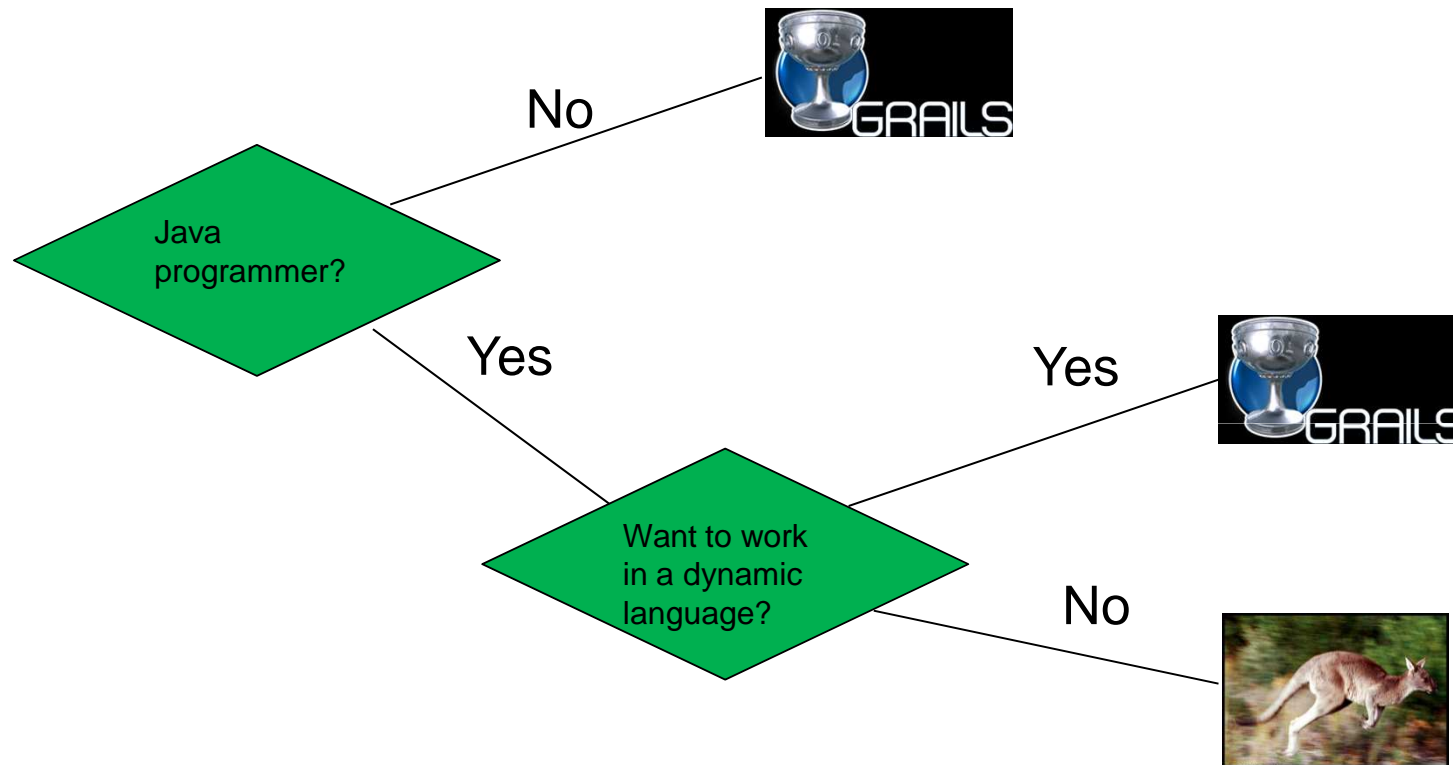


# Commonality



- Dramatically improved productivity
- Convention over configuration
  - Learns lessons from other platforms such as Ruby on Rails
- Elimination of busywork
  - Enable developers to write only code that add value
- Both Grails and Roo build on the solid Spring underpinnings
  - You don't need to throw away your server/management solution
  - Easy to work with existing code and skill set

# Grails and ROO: Choosing between the Two Best Productivity options on the JVM



Whatever you want to do, the days of creating projects by hand are over – Bye Bye Boilerplate

- Important you also understand the benefits that Grails can bring
- Grails creator Graeme Rocher will take you through *Grails in Depth* this afternoon
- In this session I'm going to focus on Roo

- Takes a code generation approach using a static language (Java)
  - Generates additional artifacts at development time that enable productivity benefits
- Makes it easier to program the code that adds value in Java

# Roo design philosophy

---



- No runtime elements
  - Generates code at development time
- Enables you to write only the Java code that adds value
- DDD-style approach
  - Driven by entity model
  - Makes it easy to move logic into entities, removes redundant layer
- Creates entire project in seconds
- Sophisticated round tripping
  - Continues to add value throughout the development lifecycle



# *But doesn't code generation suck?*



- *Yes* – If it assumes that writing code is evil
  - MDA approach
  - Doodeware
- *No* - If it aims to prioritize writing the *right* code
- Roo starts with the assumption that you *want* to write code rather than play with models
- Note
  - Rails also generates code

# Benefits of Code Generation



- (Obvious) Eliminate need to write code that is implied already – DRY
  - JavaBean methods
  - Web layer code
- Consistency
- Adds leverage, reduces barriers to integrating other technologies
  - `install security`

- 
- Roo is Spring at runtime
  - Uses Spring features you may not have seen, but essentially Spring best practices
  - Uses code generation and AspectJ to simplify the Java code that you need to write

- Roo is an open source Spring community project
- You can download it separately
- Or get it with STS
  - I'll be demonstrating it with STS (2.1.0SR01-e3.5)
- **If you use Eclipse and Spring, you should use STS – It's free!**
  - Now also adding high quality Groovy and Grails support

# A Building Block: Spring Shell



- Interactive shell helps you
  - Create projects
  - Add to projects
- Monitors the application, updating generated artifacts as you work
  - Maintains its own model
- Tab completion
- Can run standalone or inside STS

- Roo scripts amount to a higher level language
- Can type commands at the Roo prompt
- Or start with existing Roo scripts
  - Save the commands you've typed
- Benefits
  - Very concise; Lots of information per character typed
  - Facilitate communication of application patterns
  - Don't constrain writing regular Java code







# *Why does Roo use AspectJ?*



- The aim is to avoid busywork, make each line of Java count
  - Generates JavaBean methods, JPA entity annotations, toString methods etc.
- You Do NOT need to write code in AspectJ to use Roo
- Delivers other benefits
  - @Configurable
  - @Transactional
- Can easily add further aspects if you choose

# *What If I Want to Get Off?*



- 
- You are not locked into Roo
  - Can stop using it at any time and maintain code yourself
  - Can even go as far as removing Roo-generated aspects with AJDT “push-in refactoring”
  - Means it’s very safe to start using Roo

# Extending Roo

---



- Roo add-ons work at development time
  - Different from Grails plugins which have a runtime element
- Add-ons automatically picked up in Roo shell
  - Add further completions
- Add-on SPI will stabilize as Roo goes GA

# Roadmap: Present Release



- Spring Roo 1.0.0.RC2
- Just released
- Very few known issues; refer readme.txt
- Not recommend for third-party add-ons yet
  - Further refinement to add-on internals are likely
- Minor changes to command syntax possible

# 1.0.0.GA Roadmap



- 1.0.0.GA will have further add-on work
- More documentation is a major priority
- Cloud Foundry integration will also ship
  - Roo applications already deploy to Cloud Foundry
  - We're already build commands to upload, deploy and undeploy your Roo apps directly from within Roo!
  - Visit [cloudfoundry.com](http://cloudfoundry.com) and obtain an account today
- Targeting final release in November

# 1.1.0 Roadmap



- More Spring project integration
  - Spring Integration
  - Spring Blaze DS
  - Spring Batch etc
- Generation IV web client technology support
  - Flex, GWT etc
- Further improvements to Web MVC model

# Project Resources

---



- [www.springsource.org/roo](http://www.springsource.org/roo) for links
- Spring Forums is an excellent source of support
  - Dedicated Roo forum at [forum.springsource.org](http://forum.springsource.org)
  - Roo team actively monitor forum and answer queries
- Public Subversion and Jira is available

## •Roo delivers real productivity gains for Java

- Builds on Java's strengths
- Popular, proven technologies you already know
- No runtime
- No lock-in
- Easy to learn and easy to use



- Active, supported, open source project