

# Pacific Northwest Software Symposium

Sheraton Bellevue

April 18 - 20, 2008

<http://www.nofluffjuststuff.com/conference/seattle/2008/04/index.html>

(event schedule as of April 20, 2008)

Fri, Apr. 18, 2008					
	Snoqualmie North	Snoqualmie South	Phoenix	Newport	Bel-Air
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Evolutionary SOA Neal Ford	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	Guerilla Unit Testing Part 1: TestNG with Code Coverage Howard Lewis Ship	Know your Java? Venkat Subramaniam
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	10 Ways to Improve Your Code Neal Ford	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Advanced Rules Programming with Drools Brian Sam-Bodden	Guerilla Unit Testing Part 2: The Weird and Wonderful EasyMock Howard Lewis Ship	Caring about your Code Quality Venkat Subramaniam
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Regular Expressions in Java Neal Ford	Grails for Struts Developers: A Groovy Alternative Scott Davis	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	The Busy Java Developer's Guide to Annotations Ted Neward	FP on JVM Venkat Subramaniam
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, Apr. 19, 2008					
	Snoqualmie North	Snoqualmie South	Phoenix	Newport	Bel-Air
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Productive Programmer: Acceleration & Automation Neal Ford	YSlow: Building Your Website for Speed Scott Davis	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	Design Patterns in Java and Groovy Venkat Subramaniam	The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Productive Programmer: Canonicity & Focus Neal Ford	Real World JSON Scott Davis	Spring in Action: Fundamentals for Developing Spring Apps Craig Walls	Testing with Groovy Venkat Subramaniam	The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward
12:30 - 1:15 PM	LUNCH				
1:30 - 3:00 PM	Code Metrics & Analysis for Agile Projects Neal Ford	Ajax development with the Yahoo! UI Library and Grails Scott Davis	Spring Cleaning: Tips for managing XML clutter in your Spring configuration Craig Walls	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Improving Java with JRuby Brian Sam-Bodden
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Test Driven Design Neal Ford	DSL in Groovy Venkat Subramaniam	From "Hello World" to Real World : Building Web Apps with Spring OSGi Craig Walls	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Improving Code Quality Nathaniel Schutta
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Apr. 20, 2008					
	Snoqualmie North	Snoqualmie South	Phoenix	Newport	Bel-Air
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Designing for Ajax, part 1 Nathaniel Schutta	MOPping Up Groovy Venkat Subramaniam	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	The Busy Java Developer's Guide to Debugging Ted Neward	A (re)introduction to Spring Security Craig Walls
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Designing for Ajax, part 2 Nathaniel Schutta	Dynamic Languages in a Spring Application Architecture Venkat Subramaniam	10 ways to use Hibernate effectively Brian Sam-Bodden	The Busy Java Developer's Guide to Monitoring Ted Neward	Exploring Maven 2 Craig Walls
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Spring-WS: Contract first web-services for Spring Craig Walls	Acceptance Testing Application Behavior Venkat Subramaniam	The Busy Developer's Guide to Scala Ted Neward	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	Introduction to Tapestry 5 Howard Lewis Ship
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	GIS for Web Developers: Adding Where to Your Application Scott Davis	Spring Dynamic Modules for OSGi Service Platforms Venkat Subramaniam	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Project Smells Nathaniel Schutta	Pragmatic Patterns with Tapestry 5 IoC Howard Lewis Ship

# Pacific Northwest Software Symposium

Sheraton Bellevue

April 18 - 20, 2008

<http://www.nofluffjuststuff.com/conference/seattle/2008/04/index.html>  
(event schedule as of April 20, 2008)

## **Beginning Drools - Rule Engines in Java by Brian Sam-Bodden**

Drools is an open source pure-Java implementation of a forward chaining rules engine. Drools can be used in a J2SE or J2EE application and allows you to express rules programatically or by building domain specific rule languages. Learn how Business Rules with Drools can make your Java applications more flexible and robust.

## **Advanced Rules Programming with Drools by Brian Sam-Bodden**

In this session you'll learn some of the more advanced features of Drools; a pure-Java Rule Engine. This session will walk through the construction of an advanced Rules application covering such topics as: - Fine control and monitoring of a Working Memory session - Using Decision Tables - Advanced Rule Language Features - Building Domain Specific Languages - Managing your Rules **Prerequisite:** *Beginning Drools*

## **Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden**

Learn how to build featured rich applications using the Eclipse Rich Client Platform. The Eclipse platform is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

## **Boosting Programmer productivity with Mylyn by Brian Sam-Bodden**

Mylyn is a task-focused toolkit for the Eclipse IDE that allows developers to focus on tasks in a way that they never been able to do before. Mylyn eliminates the constant context switching produced by typical ways IDEs are used. No more scrolling/browsing/searching/tagging/sending emails with progress updates... Mylyn provides a new way of working that allows you to focus on specific tasks by reducing information overload. Mylyn also provides a framework for integrating with the most commonly usage task tracking systems and version control systems. In this talk you'll learn how Mylyn can boost your productivity as a Java developer by letting you get the most out of your IDE.

## **Improving Java with JRuby by Brian Sam-Bodden**

Learn how JRuby can bring simplicity to the complex and rich APIs available in the Java platform. In this session you'll learn how to use JRuby to tackle some common tasks in Java SE and Java EE as well as how to abstract and simplify complex APIs.

## **Beginning Object-Relational Mapping with Hibernate by Brian Sam-Bodden**

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

## **10 ways to use Hibernate effectively by Brian Sam-Bodden**

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

## **Spring in Action: Fundamentals for Developing Spring Apps by Craig Walls**

Spring has been one of the most exciting frameworks to emerge in the past few years. With Spring you can decouple your application's objects, enrich them with AOP, and apply transactional boundaries and security to them declaratively. It simplifies data access, remoting, web services, and JMS. It comes with its own web

framework. And, even though Spring eliminates much of the need for EJBs, it will still integrate nicely with any EJBs you may have lying around. What's not to love?

### **Spring Cleaning: Tips for managing XML clutter in your Spring configuration by Craig Walls**

The biggest complaint about Spring is the vast amount of XML required to configure an application. In this presentation, I'll show you ways to reduce or even eliminate much of the XML required to configure Spring.

### **From "Hello World" to Real World : Building Web Apps with Spring OSGi by Craig Walls**

You're already designing your web application in layers (aren't you?). Then why are you still deploying it as one big WAR file? In this presentation, I'll show you how to use Spring and OSGi to build a web application in individual bundles that can be managed and deployed independent of each other.

### **A (re)introduction to Spring Security by Craig Walls**

Spring Security (formerly known as Acegi Security) is a very powerful and flexible security framework for Java. Based on the Spring Framework, Spring Security provides declarative method and web level security including a wealth of options for meeting your application's specific security needs.

### **Exploring Maven 2 by Craig Walls**

Learn how to clean up your build process with Maven 2.

### **Spring-WS: Contract first web-services for Spring by Craig Walls**

Many web-service platforms make web-services easy by simply SOAP-ifying an object's interface. That's certainly a quick way to get started with web-services, but what happens when the object's interface changes?

### **Guerilla Unit Testing Part 1: TestNG with Code Coverage by Howard Lewis Ship**

Part one (of two) covers the TestNG unit testing framework, and shows how it integrates with tools such as Emma or Cobertura (for code coverage) and Selenium (for integration testing).

### **Guerilla Unit Testing Part 2: The Weird and Wonderful EasyMock by Howard Lewis Ship**

In part two (of two) we go in depth on EasyMock, the weird and wonderful tool for creating mock objects on the fly. We'll do a good bit of live coding as we examine how to use, tame and extend this powerful tool.

### **Introduction to Tapestry 5 by Howard Lewis Ship**

Tapestry 5 is a complete rewrite of Tapestry from the ground up. It takes everything good about Tapestry and cranks the volume up to eleven, while removing the frustrating parts of using Tapestry. This session takes the wraps off this new and innovative technology, showing off important new features such as live class reloading (the ability to change your Java classes and continue using the application without interruption or redeployment), the simplified coding model, and the total lack of XML. This session is of interest to those already using Tapestry 4, and those new to Tapestry and ready to jump on the bandwagon.

### **Pragmatic Patterns with Tapestry 5 IoC by Howard Lewis Ship**

Everyone likes the Gang of Four design patterns, but it's not always clear just how to make use of them in your day to day coding efforts. Hidden inside Tapestry 5 is an Inversion of Control (IoC) container that is structured around several common patterns (Chain of Command, Strategy, Facade and Filter Chain will be covered). This isn't academic navel-gazing ... this is about leveraging the common patterns so that you can write code you can easily test, and about creating frameworks and toolkits that can be easily extended. We'll see how Tapestry uses these patterns, and go from there into how you can apply the same techniques to your own projects, resulting in better, cleaner, more testable code.

### **Improving Code Quality by Nathaniel Schutta**

It seems that software follows the second law of thermodynamics - in other words, code tends towards disorder. Of course it doesn't have to be that way, and we have a number of tools and techniques that we can apply to keep our code in tip top shape. This talk will discuss ten things you can do to fight back!

### **Designing for Ajax, part 1 by Nathaniel Schutta**

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decided where to sprinkle

the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

### **Designing for Ajax, part 2 by Nathaniel Schutta**

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

### **JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta**

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

### **Project Smells by Nathaniel Schutta**

We all know that code can have a certain...odor but frankly so can projects. Everyone has their favorite horror story or tale of a death march. In this talk, we'll discuss common project smells and what you as a developer can do to maintain your sanity - and your hair line!

### **Evolutionary SOA by Neal Ford**

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

### **10 Ways to Improve Your Code by Neal Ford**

No one writes perfect code: even the best developers fall into bad habits and traps. This talk illustrates blind spots and helps you write better code.

### **Regular Expressions in Java by Neal Ford**

Regular expressions should be an integral part of every developer's toolbox, but most don't realize what an important topic it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

### **Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford**

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

### **Productive Programmer: Acceleration & Automation by Neal Ford**

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

### **Productive Programmer: Canonicity & Focus by Neal Ford**

Getting work done in modern office environments is a daunting task. This session tackles 2 of the things that drag down developer productivity: lack of focus and creeping repetition.

### **Code Metrics & Analysis for Agile Projects by Neal Ford**

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

### **Test Driven Design by Neal Ford**

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

### **Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis**

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

### **Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis**

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

### **Grails for Struts Developers: A Groovy Alternative by Scott Davis**

Struts is the defacto web framework for Java web developers. It has been with us since 2001. Struts enjoys unprecedented success -- most surveys place its market share between 60% and 70%. It introduced a whole generation of web developers to the phrase 'MVC' (Model / View / Controller). Grails 1.0 was released in 2008. It marries the modern features of Rails with the need for legacy support for Struts. Grails is inspired by Rails, but it is not a simple port of the project to Java. It takes the ideas of Rails, but expresses them in familiar Java libraries like Spring and Hibernate. It also leverages a new dynamic language for the JVM called Groovy.

### **YSlow: Building Your Website for Speed by Scott Davis**

How optimized is your website? YSlow, a FireFox/FireBug plugin, doesn't pull any punches. It gives any website an A, B, C, D, or F rating based on 14 individual analysis points. You'll be amazed (or depressed) at what YSlow thinks of your site. In this talk, we'll walk through these points step by step, learning what Yahoo! (the creator of this utility) does to keep its web properties running as quickly as possible.

### **Real World JSON by Scott Davis**

JavaScript Object Notation is becoming a familiar delivery platform for Web 2.0 content. JSON gives you all of the flexibility of a RESTful web service without the hassle of trying to deal with deeply nested, complex XML in a language that is conspicuously lacking in native XML support. In this talk, we look at popular websites (like Yahoo!) that offer JSON output. We look at client-side JavaScript code that effortlessly consumes JSON in the browser. We even look at ways to easily generate JSON from Java Servlets (using JSON.org libraries) and the native support for JSON that Grails offers out of the box.

### **Ajax development with the Yahoo! UI Library and Grails by Scott Davis**

Yahoo! is a company that eats its own dog food. They open sourced the Ajax code that drives many of their own websites, including their eponymous homepage, Yahoo! Mail, and Yahoo! News. Come see first hand how the various pieces of the library work together as a seamless whole. We'll look at some of the everyday useful widgets like the onscreen JavaScript logger (which effectively brings Log4J-style logging to JavaScript) and the calendar components. We'll see how event handling is managed in a cross-browser fashion. We'll look at tabbed interfaces, multi-level menus, and panels and dialog boxes that end up making your website look more like a OS-level desktop than a traditional webpage.

### **GIS for Web Developers: Adding Where to Your Application by Scott Davis**

Based on the book GIS for Web Developers, this talk demonstrates how you can build your own Google Maps in-house using nothing but open source software. The Portland, Oregon Transit Authority recently migrated from a proprietary web mapping solution to the suite of 100% free and open source software discussed in this book. We look at Java-based clients, Java-based servers, and everything in between. We

also discuss integrating free, public domain data from sources like the US Census Bureau and the USGS. If you're looking for real-world examples of AJAX in use, you'll find it here. If you're looking for real-world examples of web services in use, you'll find it here.

### **The Busy Java Developer's Guide to Annotations by Ted Neward**

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

### **The Busy Java Developer's Guide to Concurrency (Part 1: Threads) by Ted Neward**

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the `java.util.concurrent` packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

### **The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) by Ted Neward**

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown. **Prerequisite:** *The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

### **The Busy Java Developer's Guide to Java Platform Security by Ted Neward**

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

### **The Busy Java Developer's Guide to ClassLoaders by Ted Neward**

If you've ever gotten a ClassCastException and just knew the runtime was wrong about it, or found yourself copying .jar files all over your production server just to get your code to run, then you probably find the Java ClassLoader mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--ClassLoaders aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose ClassLoader problems. And once you've got that, and hear about ClassLoaders' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like ClassLoaders after all... maybe.

### **The Busy Java Developer's Guide to Debugging by Ted Neward**

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to have some basic ideas about bug-tracking in your toolbox. Learn to make use of the wealth of tools that the Java Standard Platform makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

### **The Busy Java Developer's Guide to Monitoring by Ted Neward**

Crashes? Outages? Slow response? We all know that it's never your code that causes these things, but for some reason those pesky system administrators still insist on paging you at 4AM to come in and fix those problems, anyway. For some reason, they just keep expecting you to support this thing, even after QA said it was OK!

### **The Busy Developer's Guide to Scala by Ted Neward**

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.

### **The Busy Java Developer's Guide to Hacking (on) the JDK by Ted Neward**

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

### **Know your Java? by Venkat Subramaniam**

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.

### **Caring about your Code Quality by Venkat Subramaniam**

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.

### **FP on JVM by Venkat Subramaniam**

Functional Programming Languages (FPLs) have been around for a long time. A lot of features that we get excited about in dynamic languages are common place in FPLs. FPLs are gaining importance due to various changes in our industry. What's exciting is that you can use them on the JVM. In this presentation we will dig into the details of what makes FPLs so interesting and look at ways to use them on the JVM?in your Java projects.

### **Design Patterns in Java and Groovy by Venkat Subramaniam**

You're most likely familiar with the Gang-of-four design patterns and how to implement them in Java. However, you wouldn't want to implement those patterns in a similar way in Groovy. Furthermore, there are a number of other useful patterns that you can apply in Java and Groovy. In this presentation we'll look at two things: How to use patterns in Groovy and beyond Gang-of-four patterns in Groovy and Java.

### **Testing with Groovy by Venkat Subramaniam**

Groovy's dynamic and Meta Programming capabilities makes it a great tool for unit testing. In this talk we will take a look at how we can use Groovy for unit testing and creating mocks both for testing Groovy code and Java code.

### **DSL in Groovy by Venkat Subramaniam**

DSL or Domain Specific Languages focus on a domain or problem at hand. They're expressive, but their restricted scope keeps them simple and small from the user point of view. However, designing them is not easy. In this presentation we will explore the features of Groovy and show how they can be used to create DSLs.

### **MOPping Up Groovy by Venkat Subramaniam**

Groovy's dynamic capabilities makes it an attractive language for Meta Programming. There are several facilities to intercept method calls to do AOP kind of operations. You can also inject methods dynamically. You can also easily perform method synthesis as well. In this presentation we will take a look at techniques that make Groovy pretty Hip for MOP.

### **Dynamic Languages in a Spring Application Architecture by Venkat Subramaniam**

Dynamic languages provide some great capabilities for adding dynamic behavior to your Spring applications. In this presentation we will take a look at the various options for mixing dynamic languages with Spring.

### **Acceptance Testing Application Behavior by Venkat Subramaniam**

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

### **Spring Dynamic Modules for OSGi Service Platforms by Venkat Subramaniam**

OSGi is a specification that helps with versioning Java modules at runtime. Spring helps with dependency injection of Java components and beans. Spring has embraced OSGi and allows you to integrate different

OSGi implementations into your Spring applications. In this presentation we will look at the rational for mixing Spring and OSGi and look at code examples of the same.