

# Northern Virginia Software Symposium

Sheraton Reston

November 02 - 04, 2007

<http://www.nofluffjuststuff.com/sh/2007-11-reston>

(event schedule as of November 1, 2007)

Fri, Nov. 02, 2007						
	Diamond Ballroom A-C	Meeting Room 9-10	Meeting Room 8	Meeting Room 7	Meeting Room 6	Meeting Room 5
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Groovy: Greasing the Wheels of Java Scott Davis	Enterprise Performance and Scalability Ted Neward	Behavior-driven development in Java Andrew Glover	Java 6 Features, what's in it for you? Venkat Subramaniam	Exploring the JavaServer Faces Ecosystem Kito Mann	Code Metrics & Analysis for Agile Projects Neal Ford
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Groovy and Java: The Integration Story Scott Davis	Java Annotations: From Definition to Consumption Ted Neward	Monitoring Software Quality with Continuous Integration Andrew Glover	OSGi: A Well Kept Secret Venkat Subramaniam	AJAX and JSF: Natural Synergy Kito Mann	Agile Project Management with Mingle Neal Ford
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Real World Grails Scott Davis	Implementing SOA Neal Ford	The Busy Java Developer's Guide to Java Platform Security Ted Neward	get Fit Venkat Subramaniam	Introduction to JBoss Seam Kito Mann	Data Integration : Beyond Cutesy Mashups Brian Sletten
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	KEYNOTE: Rethinking Enterprise Applications: how do we think about enterprise apps in a post-J2EE world by Ted Neward					

Sat, Nov. 03, 2007						
	Diamond Ballroom A-C	Meeting Room 9-10	Meeting Room 8	Meeting Room 7	Meeting Room 6	Meeting Room 5
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Enterprise Applications with Spring: Part 1 Ramnivas Laddad	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford	Database Refactoring Pramod Sadalage	Creating Agile Requirements David Hussman	The Secrets of GORM Scott Davis	Give it a REST Brian Sletten
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Enterprise Applications with Spring: Part 2 Ramnivas Laddad	Productive Programmer: Automation and Canonicity Neal Ford	Recipies for Continuous Database Integration Pramod Sadalage	Getting Agile Planning and Tracking Up and Running David Hussman	Ajax development with the Yahoo! UI Library and Grails Scott Davis	RESTlet for the Weary Brian Sletten
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	Introduction to JRuby Neal Ford	Internationalization and Localization in Java David Bock	Executable Documentation David Hussman	Spring into Groovy Venkat Subramaniam	NetKernel : XML Processing for the 21st Century Brian Sletten
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Rails for JRuby Neal Ford	AspectJ for Spring Developers Ramnivas Laddad	Introducing Agility to Large Organizations David Bock	Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense Andrew Glover	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS					

Sun, Nov. 04, 2007						
	Diamond Ballroom A-C	Meeting Room 9-10	Meeting Room 8	Meeting Room 7	Meeting Room 6	Meeting Room 5
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	10 Ways to Improve Your Code Neal Ford	Introduction to Spring Security Mark Fisher	Domain Specific Languages Venkat Subramaniam	Leveraging annotations with AOP Ramnivas Laddad	Groovy 101: putting Groovy to use quickly Andrew Glover	Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Introduction to Hibernate Scott Leberknight	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Drooling with Groovy and Rules Venkat Subramaniam	JavaScript Security - Seeing the possibilities of a sand-boxed scripting language Jason Harwig	Getting Started with Grails Jason Rudolph	Leading Agile Projects: Maintaining Sustainable Agility David Hussman
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	Spring, Hibernate, and JDBC Integration Patterns Scott Leberknight	The Role of Spring in an ESB Mark Fisher	FP for Java Programmers Venkat Subramaniam	Debugging and Testing the Web Tier Neal Ford	Going Further with Grails Jason Rudolph	Requirements Drive Design and Development (RDDD) Mark Johnson
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	10 ways to use Hibernate effectively Brian Sam-Bodden	Message Driven POJOs Mark Fisher	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Advanced Selenium Neal Ford	Advanced Domain Models in Grails: Enterprise Integration Made Easy Jason Rudolph	Software Development Risk Analysis techniques Mark Johnson

# Northern Virginia Software Symposium

Sheraton Reston

November 02 - 04, 2007

<http://www.nofluffjuststuff.com/sh/2007-11-reston>  
(event schedule as of November 1, 2007)

## **Behavior-driven development in Java by Andrew Glover**

Behavior-driven development, or BDD, has attracted a lot of attention a la RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework. In this session, we'll look at what BDD is and how it shifts the traditional testing vocabulary from being test-based to behavior-based.

## **Monitoring Software Quality with Continuous Integration by Andrew Glover**

The practice of continuous integration facilitates early visibility into the development process by regularly conducting software builds, thus integrating disparate software pieces earlier than later, which often times minimizes the interval between when a defect is coded and when it is discovered. Given the automated nature of continuous integration spawned builds, software teams can now start to look at their build process as something more useful than a simple compile and test process.

## **Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense by Andrew Glover**

Are your Ant builds giant XML files that scream for attention? Why not enhance your build process to act like a quality gate, much like a test suite would?

## **Groovy 101: putting Groovy to use quickly by Andrew Glover**

It has been said that Grails is the addiction and Groovy is the drug. If you want to start building slick web applications rapidly with Grails it helps to start with a solid understanding of the Groovy language itself.

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

Software development is expensive, when business rules are hard-coded in your application's source code, changes and additions to those rules translate to wasted time and money. Good object-oriented, component-based approaches can alleviate the burden of keeping up with changes in the business world but they still require that expert knowledge of the changes be passed from the decision makers to the business analysts and finally to programmers that need to implement these changes. Business Rule Engines and Business Rule Languages are based on the basic premise of separation of concerns by empowering business domain experts to express the rules of business in a way that it is directly usable by applications.

## **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.

## **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.

## **Data Integration : Beyond Cutesy Mashups by Brian Sletten**

Ever since we started doing relational joins, we've looked for ways to tie data together. The web has given us no end of new data sources to integrate but it seems like the best we can come up with is locating Starbucks on Google Maps. The problem with browser-based mashups is that they don't survive the session, we have no way of referring to the results in future queries and ultimately we don't maintain ownership or control of the process. We want control of our data and our mashup results. We want ever more ways to view, explore and requery them in multi-faceted ways. Do you know what your data integration strategy is for the next few years? Are you sure? You owe it to yourself to come find out.

## **Give it a REST by Brian Sletten**

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the

industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

### **RESTlet for the Weary by Brian Sletten**

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community.

### **NetKernel : XML Processing for the 21st Century by Brian Sletten**

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

### **Internationalization and Localization in Java by David Bock**

Internationalization and Localization in Java is easy, right? Everyone knows you just store your strings in some resource bundles, set the locale, wave your hands a little bit, and your application is good-to-go. Right? Maybe not... Java provides some great utilities to get started, but leaves you needing more when it comes to things like screen layout, cultural sensitivities, semantic differences in translation, use of color and iconography, and other issues.

### **Introducing Agility to Large Organizations by David Bock**

For several years, I was a member of a team of people caught in the middle of a 200+ person software development company, with senior management wanting "buzzword compliant process improvement" such as CMMI, and engineers wanting more "agile" solutions (and people on both sides confusing Agile with ad-hoc). We were responsible for sorting it all out. Reconciling this was a herculean effort, and can be a source of lessons learned for your own process improvement efforts. Are you trying to be more agile in your organization? Are you expecting it to be harder than it needs to be because of political and bureaucratic forces beyond your control? Do you have to "educate" your senior management to protect them from buzzwords? Come learn from my successes... and mistakes.

### **Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools by David Bock**

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? How many times have you been the "new guy" on an established project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good structural guidelines and several tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

### **Creating Agile Requirements by David Hussman**

Successful project communities balance written requirements with a healthy amount of discussion. This is at the core of requirements that could be deemed "agile". Many agile projects choose to use user stories, but others may be using use cases or other forms of written requirements. This session is for anyone wanting to improve their requirements, including the creation of good requirement and the presentation styles that help people focus on creating great software products, and stop focusing on documents.

### **Getting Agile Planning and Tracking Up and Running by David Hussman**

If your company is using agile or thinking about it, this session will show you how to plan and tracking an agile project. Examples projects will be discussed, including the glory and horrors. Various planning tools that help distributed teams will be presented as well as a collection of lo-fi tools which truly help find and address the issue that plagues so many projects: "when are we going to complete this project".

### **Executable Documentation by David Hussman**

Why is so much documentation worthless? Wouldn't it be nice if your documentation actually reflected what your system does? One way to do this is to create what is being called executable documentation or executable specifications. If you are struggling with ambiguous requirements, lack of contact with the business, or a chasm between development and testing, this session is for you.

#### **Leading Agile Projects: Finding Your Groove in the First 4 Iterations by David Hussman**

Summary: There are many books about agile but few provide a path for guiding you through the beginning of an agile project. Whether you are preparing for your first agile project, or taking the lead for the first time, this session will provide a guided tour filled with practical advice and a pile of anecdotes.

#### **Leading Agile Projects: Maintaining Sustainable Agility by David Hussman**

Once your agile project is rolling along, there are many bumps and roadblocks which can derail the train. Whether you are leading the project formally or informally, there are techniques you can use to keep the project alive and innovative. This session will cover skills and techniques for leading sustainable project communities.

#### **JavaScript Security - Seeing the possibilities of a sand-boxed scripting language by Jason Harwig**

JavaScript's popularity in recent years has brought with it the attention of hackers, white and black. Both sides looking for ways to do things that weren't intended with the scripting language.

#### **Getting Started with Grails by Jason Rudolph**

Grails is an open-source web application framework that's all about getting things done. Grails combines best-of-breed Java technologies (including Hibernate and Spring), convention over configuration, and the powerful and dynamic Groovy language. Together with these elements and Groovy's ability to seamlessly integrate with your existing Java code, Grails finally legitimizes rapid web application development for the Java platform.

#### **Going Further with Grails by Jason Rudolph**

Grails provides a tremendous jump-start to any web application. With easy scaffolding and zero-configuration ORM, you're up and running in no time. But what's needed to effectively move from a vanilla Grails project to a fully-customized application fit for public consumption?

#### **Advanced Domain Models in Grails: Enterprise Integration Made Easy by Jason Rudolph**

Have you seen someone develop a Rails or Grails application in a matter of minutes, only to later discover that their domain model and database schema followed conventions that are different from your existing systems? Or perhaps you're interested in using Grails, but you don't want to duplicate your existing Java domain classes in Groovy. In this session, we'll see how Grails makes it easy to hook into your pre-defined schemas or existing entity classes, while still getting all the rapid application development (RAD) goodness that Grails has to offer.

#### **Exploring the JavaServer Faces Ecosystem by Kito Mann**

This session examines the ecosystem that is growing around JavaServer Faces.

#### **AJAX and JSF: Natural Synergy by Kito Mann**

With the emergence of AJAX as a preferred way of building web user interfaces, JavaServer Faces (JSF) has proved itself to be a natural fit for integrating AJAX with Java sever-side logic.

#### **Introduction to JBoss Seam by Kito Mann**

JBoss Seam is a popular open-source application framework for Java Platform, Enterprise Edition (Java EE) 5.0. For web application developers, a significant benefit of Seam is that it greatly enhances JavaServer Faces technology. This session explains key Seam features such as tight integration with EJB3, Hibernate and JPA integration, conversations, RESTful web pages, and so on.

#### **Introduction to Spring Security by Mark Fisher**

Spring Security (a.k.a. Acegi) enables self-contained, consistent solutions for securing your applications. The interceptor-based approach is non-invasive even when extended to accommodate domain-specific

requirements. The two main security processes (authentication and authorization) are decoupled in order to provide flexibility across a wide variety of providers and strategies.

### **The Role of Spring in an ESB by Mark Fisher**

An Enterprise Service Bus (ESB) brings flow-related functionality such as message routing and transformation to a Service-Oriented Architecture. An ESB also provides a layer of abstraction with endpoints for various protocols and transports. These features promote decoupling of integration logic from business functions, flexibility in the transport layer, and pluggability of POJO services.

### **Message Driven POJOs by Mark Fisher**

One of the most exciting new features of Spring 2.0 is its support for Message-Driven POJOs. It is now possible to receive JMS messages asynchronously and delegate the handling of those messages to simple objects. If your POJO has a return value, it will automatically be sent to a reply destination. Spring's messaging containers support configurable pooling of concurrent consumers and offer full integration with Spring's transaction management.

### **Requirements Drive Design and Development (RDDD) by Mark Johnson**

Validate that requirements are not missed during the design and development process by creating Requirements document test fixtures to clarify and validate the requirements between the end users, business analysts, architects, and developers early in the project.

### **Software Development Risk Analysis techniques by Mark Johnson**

Once you leave academic "hello world" projects, software development is full of unknowns which result in the high rate of project failure we see too often in industry. Reasons for a project failure will vary based on the stakeholder interviewed. This session will provide a software development risk framework and examples you can apply in your projects to reduce or at least soften the impact of failure.

### **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.

### **Agile Project Management with Mingle by Neal Ford**

Mingle is an innovative project management tool with "skinnable religion", virtual card walls, highly customizable workflow, and role-based views. This talk describes its setup, use, and some implementation details of how it was created.

### **Implementing SOA by Neal Ford**

This talk avoids SOA hype and gets to the meat of the matter: how do you implement a Service-Oriented Architecture, what are the technological pitfalls, how do you test it, and what traps should you avoid. No marketecture: just implementation details.

### **Productive Programmer: Acceleration, Focus, and Indirection by Neal Ford**

This session discusses how to use the Productive Programmer principles of acceleration, focus, and indirection to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

### **Productive Programmer: Automation and Canonicity by Neal Ford**

This session discusses how to use the Productive Programmer principles of automation and canonicity to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

### **Introduction to JRuby by Neal Ford**

This session describes JRuby, the 100% pure-Java implementation of the Ruby programming language. It covers the basics of programming with JRuby and examples of how to integrate it into existing Java projects.

### **Rails for JRuby by Neal Ford**

This session explains all the hype surrounding Ruby on Rails, in a context familiar to Java developers. It covers convention over configuration, ActiveRecord, controllers, views, Ajax, scaffolding, testing, and deployment...on the JVM, using JRuby.

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

No one writes perfect code, and every developer eventually falls into a slump where they just crank out the same code day after day. This session illustrates 10 different ways to improve your code, covering sacred cows, good citizens, smells, and more.

### **Debugging and Testing the Web Tier by Neal Ford**

As our applications have spilled from the server across the wire to the web tier, we increasingly must debug and test in the browser. This session covers debugging and testing tools for clients, JavaScript, and Ajax.

### **Advanced Selenium by Neal Ford**

This session discusses advanced Selenium techniques for testing web applications. It discusses techniques for both TestRunner and Remote Control Selenium, including data driven tests, creating branch points, testing Ajax applications, creating flexible tests, integration with continuous integration, and tons more.

### **Database Refactoring by Pramod Sadalage**

Evolve an existing database schema a small bit at a time to improve the quality of its design without changing its semantics.

### **Recipes for Continuous Database Integration by Pramod Sadalage**

is your database being tested and integrated like your application code? do you have a database integration strategy? do you have a process to upgrade your database as you make subsequent releases? Make the database part of your Continuous Integration cycle and benefit from it.

### **Enterprise Applications with Spring: Part 1 by Ramnivas Laddad**

This session (part 1 of the two-part session) shows the core concepts in the Spring Framework -- the most popular lightweight container that recently crossed 1 million downloads.

### **Enterprise Applications with Spring: Part 2 by Ramnivas Laddad**

This session (the second part of the 2-part session) will cover advanced concepts in the Spring framework. While the core concepts in the first session will get you started with Spring, the advanced concepts in this session will make you effective at developing Spring-based applications.

### **AspectJ for Spring Developers by Ramnivas Laddad**

If you are keeping up-to-date with all the cool features in Spring 2.0, you have surely heard about much improved integration with AspectJ. AspectJ is for real. Come to this session to understand the core concepts of this wonderful technology and how to use it to create even simpler Spring-based applications.

### **Leveraging annotations with AOP by Ramnivas Laddad**

Specifying metadata using annotations has gained huge popularity since its introduction in Java 5. However, the story on consuming annotations isn't as clear. Reading and processing annotation is still a complex process often requiring you to understand byte-code manipulation tools and their low-level API. As a result, most developers shy away from using custom annotations, limiting their usages of annotations only those prescribed by frameworks. The result is missed opportunities for programming simplification. In this session, we explore how AOP can make it a simple task to consume annotation in a powerful manner.

### **Groovy: Greasing the Wheels of Java by Scott Davis**

This is the year of the dynamic scripting language. Ruby (and Rails) has won the hearts and minds of many independent software developers. JavaScript is experiencing a renaissance thanks to the wild success of AJAX and websites like Google Maps. And Groovy (JSR-241) brings the same level of excitement and "scripting goodness" to the Java platform.

### **Groovy and Java: The Integration Story by Scott Davis**

I'm attracted to Groovy because of its spirit of inclusiveness. Because it extends my platform of choice, not replaces it -- include a single JAR in your classpath and you are Groovy-enabled. Because it offers

full bidirectional integration with Java. Because it offers a nearly flat learning curve for experienced Java developers. Come see how you can use Groovy to augment your existing Java codebase.

### **Real World Grails by Scott Davis**

Scott Davis is the Editor in Chief of aboutGroovy.com. The website, in addition to being, umm, about Groovy, is implemented in Grails. This talk shows you how to get started with Grails, but also talks about the experience of using it in a live, production web site.

### **The Secrets of GORM by Scott Davis**

GORM (the Grails Object/Relational Mapper) is one of the many high points of the Grails web framework. GORM is a thin Groovy wrapper over Hibernate, but that doesn't begin to capture excitement of what GORM brings to the party. Imagine being able to call `book.save()` and `book.delete()` on your `Book` class; calling `Book.get(1)` to retrieve your book from the database by primary key; using `Book.list()` to pull an `ArrayList` of `Book` objects into your application. Now imagine getting all of that functionality (and more) for free with each new class you define. No interfaces to implement. No abstract classes to extend. Persistence that is transparent, automatic, and simple to use: GORM.

### **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.

### **Introduction to Hibernate by Scott Leberknight**

This session introduces the Hibernate Object/Relational Mapping (ORM) framework, showing the basics of persisting Java objects to relational databases. No prior knowledge of Hibernate or ORM is assumed.

### **Spring, Hibernate, and JDBC Integration Patterns by Scott Leberknight**

Spring's Hibernate integration simplifies applications that use Hibernate by removing tedious and repetitive infrastructure code that you need to write. Spring JDBC achieves a similar simplification for executing SQL statements using the lower level JDBC API. So when do you use one or the other, and what are the important usage guidelines? Come to this session to find out.

### **Enterprise Performance and Scalability by Ted Neward**

Wondering why your enterprise Java app just? sucks? Trying to figure out why you can't get more than 10 concurrent users online at the same time? Looking for ways to try and spot the slowdowns and ways to fix them?

### **Java Annotations: From Definition to Consumption 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 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 and Monitoring 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 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.

### **Java 6 Features, what's in it for you? by Venkat Subramaniam**

What benefit do new Java 6 features offer you. Are there issues with using these features. The objective of this presentation is not simply to introduce you to the features, but to the effective use of these as well.

### **OSGi: A Well Kept Secret by Venkat Subramaniam**

In this presentation we will introduce OSGi and discuss how it can help modularize and version your enterprise Java applications.

### **get Fit by Venkat Subramaniam**

Unit testing tells you, the programmer, that your code (and the change) meets your expectations. How do you know if you are meeting your customers' expectations? Agile development is all about feedback and doing what's relevant to the customers, isn't it? Framework for Integration testing or Fit helps you to automate tests for customer expectations.

### **Spring into Groovy by Venkat Subramaniam**

What do you get when you mix an agile, object-oriented, dynamic language with a lightweight, flexible, and extensible framework? You get a Groovier Spring. Spring allows you to develop using Groovy as much as Java. Groovy brings some neat concepts to the Java Platform that is hard to realize directly through the Java language. Using these capabilities can lead to elegant and easier Spring development.

### **Domain Specific Languages by Venkat Subramaniam**

Domain Specific Languages or DSLs are languages that target a specific kind of problem or domain. We've had various degree of success with DSLs, over the past several years, in narrow areas. However, DSLs are not widely used in general purpose application partly because the popular widely used languages today do not make it easy.

### **Drooling with Groovy and Rules by Venkat Subramaniam**

Rule based programming allows us to develop applications using declarative rules. These can simplify development in applications where such rules based knowledge is used for decision making.

### **FP for Java Programmers by Venkat Subramaniam**

Most interest around Functional Programming (FP) has been academic until recently. Recent commercial languages are beginning to exploit FP features. Knowing more about FP will not only help us make better use of these features, but to exploit those. In this session we will take a close look at FP.