

Northern Virginia Software Symposium

Sheraton Reston

April 25 - 27, 2008

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

(event schedule as of April 26, 2008)

Fri, Apr. 25, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME - DIAMOND BALLROOM					
1:15 - 2:45 PM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Know your Java? Venkat Subramaniam	Give it a REST Brian Sletten	Agile, Smagile: What's Working? - What's Not? David Hussman	Evolutionary SOA Neal Ford	Monitoring Software Quality with Continuous Integration Andrew Glover
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Got Guice? Venkat Subramaniam	RESTlet for the Weary Brian Sletten	Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman	Test Driven Design Neal Ford	Tactical Continuous Integration with Hudson Andrew Glover
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Grails for (Recovering) Struts Developers: A Groovy Alternative Scott Davis	Caring about your Code Quality Venkat Subramaniam	REST - Live! Brian Sletten	Leading Agile Projects: Maintaining Sustainable Agility David Hussman	Regular Expressions in Java Neal Ford	Easy BDD with Groovy Andrew Glover
6:30 - 7:15 PM	DINNER - DIAMOND BALLROOM					
7:15 - 8:00 PM	Keynote: by Neal Ford					

Sat, Apr. 26, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
8:00 - 9:00 AM	BREAKFAST - DIAMOND BALLROOM					
9:00 - 10:30 AM	Simplifying Enterprise Applications with Spring, Part 1 Mark Fisher	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	Acceptance Testing Application Behavior Venkat Subramaniam	YSlow: Building Your Website for Speed Scott Davis	Introduction to JRuby Neal Ford	Architecture and Agility Are Not Mutually Exclusive David Hussman
10:30 - 11:00 AM	BREAK - DIAMOND BALLROOM FOYER					
11:00 - 12:30 PM	Simplifying Enterprise Applications with Spring, Part 2 Mark Fisher	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	DSL in Groovy Venkat Subramaniam	Real World JSON Scott Davis	Meta-programming JRuby for Fun & Profit Neal Ford	Test Driven Everything David Hussman
12:30 - 1:30 PM	LUNCH - DIAMOND BALLROOM					
1:30 - 3:00 PM	The Busy Java Developer's Guide to Debugging Ted Neward	Enterprise Integration Patterns with Spring - Part I Mark Fisher	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	10 Ways to Improve Your Code Neal Ford	Groovin' builds Gant get any easier Andrew Glover	Shippers Unite! Jared Richardson
3:00 - 3:15 PM	BREAK - DIAMOND BALLROOM					
3:15 - 4:45 PM	The Busy Java Developer's Guide to Monitoring Ted Neward	Enterprise Integration Patterns with Spring - Part II Mark Fisher	Advanced Rules Programming with Drools Brian Sam-Bodden	SOA Unplugged Mark Richards	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS					

Sun, Apr. 27, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
8:00 - 9:00 AM	BREAKFAST - DIAMOND BALLROOM					
9:00 - 10:30 AM	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	A Thorough Introduction To Groovy Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward	Enterprise Messaging Using JMS (Part 1) Mark Richards	Restoring Agility: Getting Your Team Back on Track Jared Richardson	Internationalization and Localization in Java David Bock
10:30 - 11:00 AM	BREAK - DIAMOND BALLROOM FOYER					
11:00 - 12:30 PM	10 ways to use Hibernate effectively Brian Sam-Bodden	Agile Test Driven Development With Groovy Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward	Enterprise Messaging With JMS (Part 2) Mark Richards	Techniques 2008 Jared Richardson	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock
12:30 - 1:15 PM	LUNCH - DIAMOND BALLROOM					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION - DIAMOND BALLROOM					
2:15 - 3:45 PM	Java Persistence: Approaching the Silver Bullet Mark Richards	Grails - Agile Web 2.0 The Easy Way Jeff Brown	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Rich Clients, Rich Data Part I : Linking Brian Sletten	Build Teams, Not Products Jared Richardson	Intermediate Maven David Bock
3:45 - 4:00 PM	BREAK - DIAMOND BALLROOM FOYER					
4:00 - 5:30 PM	Transaction Design Patterns Mark Richards	Advanced Web Development With Grails Jeff Brown	The Busy Developer's Guide to Scala Ted Neward	Rich Clients, Rich Data Part II : Consuming Brian Sletten	10 Tips for Getting Your Project Back on Track Jared Richardson	Surviving Middle Management David Bock

Northern Virginia Software Symposium

Sheraton Reston

April 25 - 27, 2008

<http://www.nofluffjuststuff.com/conference/reston/2008/04/index.html>
(event schedule as of April 26, 2008)

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.

Tactical Continuous Integration with Hudson by Andrew Glover

This session will walk attendees through a series of iterations on a fictional Java project where an automated build system is created that facilitates compilation, testing, inspection, and deployment. This build system is then plugged into the Hudson CI server and as features are coded using Agile techniques like developer testing, attendees will ultimately see firsthand how a Continuous Integration process reduces risk and improves software quality.

Easy BDD with Groovy by Andrew Glover

Behavior-driven development, or BDD, has attracted a lot of attention via RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework modeled off of the xUnit paradigm. But JBehave isn't the only framework available for Java developers-- with the advent of Groovy, new options are available for embracing BDD in the spirit of RSpec's innovative behavior based DSL.

Groovin' builds Gant get any easier by Andrew Glover

There's no question that Ant is the de facto standard for building Java applications; however, even its creator has acknowledged an inherent limitation with Ant's expressiveness due to its reliance on XML. Recently, the popularity of Ruby and the Rails framework has brought to focus Ruby's de facto build platform: Rake. Rake's expressiveness comes from its reliance on Ruby itself to define a DSL for software assembly. While Rake's ultimate focus is Ruby, there are a number of interesting projects that utilize expressive DSLs for building Java including Gant, which uses Groovy as a DSL format and builds upon Ant's existing cornucopia of tasks.

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.

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*

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.

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.

REST - Live! by Brian Sletten

You've read the articles, the books, the Ph.D. thesis and all of the meta-commentary about building RESTful APIs, but you're still not sure where to begin. This is an interactive session and has almost no slides. You should come prepared to discuss ideas and maybe pair program with me and everyone else in the room. Bring your ideas for open source projects that we might want to expose through a resource-oriented model. Bring your concerns about your domains that you are convinced don't fit this model.

What's Going On? : Complex Event Processing w/ Esper by Brian Sletten

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

Rich Clients, Rich Data Part I : Linking by Brian Sletten

You hear a lot of talk about rich clients, but the richness they purport to provide is predicated on having access to rich data as well as a rich user interaction style. Without the right levels of abstraction, it is hard to address and link all of the data we have to care about these days. Additionally, the web sites that do support the notion of linking require you to publish your data into TheirSpace. Forget that. You want to be able to link publicly available information to sensitive information in YourSpace. Ever since we started doing relational joins, we've looked for ways to tie data together. The problem is, the relational model is a bit tired and doesn't move at the speed of the Net. We need schemes that integrate relational data, web pages, XML files, RSS feeds and various other sources of information.

Rich Clients, Rich Data Part II : Consuming by Brian Sletten

You hear a lot of talk about rich clients, but the richness they purport to provide is predicated on having access to rich data as well as a rich user interaction style. Without the right levels of abstraction, it is hard

to address and link all of the data we have to care about these days. Additionally, the web sites that do support the notion of linking require you to publish your data into TheirSpace. Forget that. You want to be able to link publicly available information to sensitive information in YourSpace. Ever since we started doing relational joins, we've looked for ways to tie data together. The problem is, the relational model is a bit tired and doesn't move at the speed of the Net. We need schemes that integrate relational data, web pages, XML files, RSS feeds and various other sources of information.

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.

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.

Intermediate Maven by David Bock

Maven is a build tool that does a lot, demos well, and leaves the build maintainers managing what seems like unbridled complexity. It doesn't have to be that way - Maven is driven by some strong 'build process methodology', and that complexity can become manageable by wrapping your head around it. Furthermore, you can migrate to Maven 'piecemeal', by mapping your existing ant build to the Maven Lifecycle and calling your existing Ant tasks - you can decide to sip the Maven kool-aid. Ideally, a build tool should be so simple and approachable that it fades into the project background and allows anyone to maintain it. Unfortunately, Maven's power comes at the expense of this ideal - Maven's philosophy is more like "the build process is so important that the people maintaining it should be steeped in the ways of Maven". This talk will give you the exposure you need without elevating The Maven Way to a religion.

Surviving Middle Management by David Bock

Most good developers eventually have the opportunity to be managers. Whether they call you the "project manager", "Technical Lead", "Lead Developer", or some other classic middle-management title, you become the 'goto' guy between management and developers. You're the guy who is expected to keep the project in-line, track a schedule, and occasionally answer the question "How's it going?", and perhaps still contribute at a technical level. So how do you do that?

Agile, Smagile: What's Working? - What's Not? by David Hussman

With the growth of agile comes the need to add a new line to the Agile Manifesto: Success over Dogma. The number of people who can say agile is growing faster than the number of people benefiting from agile practices. There are now many successful agile projects, yet there are also a growing number of projects claiming to be agile but not seeing any of the benefits agile methods provide. This session will discuss successful adoptions of agile, dumb things you can do to muck it up, and more.

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

Although 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. **Prerequisite:** *Leading Agile Projects: Finding Your Groove in the First 4 Iterations*

Architecture and Agility Are Not Mutually Exclusive by David Hussman

Being agile does not mean living life one iteration at a time. Agile projects without a long view can run into the common design problems of the past. Planning iteration by iteration is often foolish and feeds the myth that agile projects do not think beyond a few weeks. Successful agile projects plan within iterations and across iterations. The later planning is called release planning and it is the forum where agility first engages architecture and other cross cutting concerns.

Test Driven Everything by David Hussman

Why do we wait to test? Of course when you read this your thoughts went to testing code. While we still wait to test code and products early, we also wait to test ideas, projects, product direction, meetings and more. This session will show you (or challenge you) to think about test driven beyond the coding realm. You will be doing some thinking and talking and other things that involve more than just listening to someone blather with slides for 90 minutes.

Shippers Unite! by Jared Richardson

An overview of the Agile software approach from the book Ship It! A Practical Guide to Successful Software Projects.

Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

Restoring Agility: Getting Your Team Back on Track by Jared Richardson

An agile team is first and foremost "a team". When that gets lost in the rush to get a product out the door, the people suffer as well as the products. It's bad for the company, but even worse for the team members. We'll learn how to defuse some of the more common problems you'll run into on dysfunctional teams.

Techniques 2008 by Jared Richardson

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

10 Tips for Getting Your Project Back on Track by Jared Richardson

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Grails - Agile Web 2.0 The Easy Way by Jeff Brown

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

Advanced Web Development With Grails by Jeff Brown

Grails makes web application development both fun and easy. This session dives beyond the basics to cover advanced details of Grails that bring the really exciting features to your applications. **Prerequisite:** *Grails - Agile Web 2.0 The Easy Way*

Simplifying Enterprise Applications with Spring, Part 1 by Mark Fisher

Developing enterprise applications isn't easy. You not only have to worry about constantly evolving business logic, but also need to address infrastructure concerns ranging from transaction management and security to manageability and integration with diverse external applications. Spring, the most popular lightweight enterprise application framework, comes to the rescue by simplifying the common needs of enterprise applications. This session (part 1 of 2) presents the core concepts of the Spring Framework.

Simplifying Enterprise Applications with Spring, Part 2 by Mark Fisher

This session (part 2 of 2) 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 help you be more effective at developing Spring-based applications.

Enterprise Integration Patterns with Spring - Part I by Mark Fisher

In the first-part of this two-part workshop, Mark will focus on the essentials of Enterprise Integration with Spring. First, he will take a whirlwind tour of Spring's enterprise integration support libraries. Next, he will discuss the "big picture" of an event-driven architecture based on messaging with an overview of key enterprise integration patterns. Attendees will leave with a clear understanding of Spring's integration capabilities and an appreciation for the benefits of message-driven architecture, ready to put that into practice in Part II.

Enterprise Integration Patterns with Spring - Part II by Mark Fisher

Building on Part I, Part II of this workshop will demo a series of messaging systems built on Spring. The samples will exercise event-driven scenarios involving distributed architectures with messaging and remoting. Each sample will highlight a variety of important enterprise integration patterns.

SOA Unplugged by Mark Richards

Awareness about Service Oriented Architecture (SOA) has grown significantly in the past several years. Unfortunately, along with that growth has come a significant amount of confusion about what SOA really is. SOA has become such a ubiquitous buzzword that it now has many faces and means different things to different people. CIO's, managers, vendors, business users, architects, and developers all see SOA differently which creates a sea of confusion about what is and isn't SOA. In this highly interactive and thought provoking session we will look beyond the hype and marketure of SOA and explore SOA from an architecture and development point of view - in other words, SOA as an architecture pattern. During this session we will look at SOA use cases, services, integration, implementation, guiding architecture principles of SOA, and attempt to answer the following question: What is and isn't SOA?

Enterprise Messaging Using JMS (Part 1) by Mark Richards

The chances are good that at some point in your career you will need to use messaging to pass information between applications, subsystems, or external systems, particularly with service-oriented architecture on the rise. The Java Messaging Service (JMS) allows Java applications to implement messaging using a standard API, thereby removing the dependency on any particular messaging provider. In Part 1 of this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, and request/reply messaging. I will begin the session by going over the basics of messaging and the JMS API. Then, through interactive coding using OpenJMS I will demonstrate how to connect to the JMS provider, send messages, receive messages, and use message properties. Please note that this is a two part session.

Enterprise Messaging With JMS (Part 2) by Mark Richards

In Part 1 of the JMS session I covered messaging models, messaging basics, the JMS API, and point-to-point messaging. In this interactive code-intensive session I will cover some additional JMS topics such as browsing queues, publishing and subscribing to messages within the pub/sub model, durable and

non-durable subscribers, message selectors, and message filtering. I will also discuss and demonstrate message prioritization, persistent and non-persistent messages, and finally message expiration (expiry). Note that this is Part 2 of a two-part JMS session. **Prerequisite:** *Enterprise Messaging With JMS (Part 1)* or some knowledge of JMS

Java Persistence: Approaching the Silver Bullet by Mark Richards

Java Persistence has come along way since the days of straight JDBC coding and custom framework development. We have at our disposal several outstanding open source frameworks such as Hibernate, Toplink, iBatis, and OpenJPA (just to name a few), and we now have a promising and emerging standards-based solution called Java Persistence API (JPA). However, all too often we find in the Java persistence space that it is a world of one-size-does-not-fit-all. We continually struggle with traditional ORM solutions like Hibernate when it comes to reporting queries, complex queries, complex relationships, and stored procedures, and we also struggle with managing the enormous amount of SQL required for solutions such as iBATIS or JDBC-based frameworks. In this coding-intensive session we will take a detailed look at identifying and overcoming the challenges we face when using frameworks such as Hibernate, iBATIS, and JPA, and how to combine the various persistence frameworks to create an effective Java persistence solution that approaches (but of course does not reach) the silver bullet.

Transaction Design Patterns by Mark Richards

Most web-based applications rely solely on the database to manage transactions, thereby freeing the developer from having to worry about transaction management. While this works in some circumstances, there are times when the use of transactions is vital to the integrity and operations of an application and its corresponding data. In this session I will demonstrate through real-world coding examples why transactions are such a critical part of the application development process. I will review the basics of both programmatic and declarative transactions, then introduce three transaction design patterns and explain when they should be applied, how to use them, and what problems they solve. By the end of this session you will see that by using transaction design patterns you can build an effective transaction management strategy for your application with very little effort.

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.

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.

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?

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.

Meta-programming JRuby for Fun & Profit by Neal Ford

Ruby is the revenge of the Smalltalkers. Not since Smalltalk has a language had such powerful meta-programming facilities. While this may seem like a minor feature, it turns out that surgical

meta-programming allows solutions to problems that are clearer, more concise, more maintainable, and take orders of magnitudes fewer lines of code.

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.

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 (Recovering) Struts Developers: A Groovy Alternative by Scott Davis

Struts enjoys an unprecedented marketshare in the Java web development space -- 60%-70% according to most surveys. As newer, modern web frameworks come to the scene, very little attention is paid to the real costs of migrating an existing Struts application. This talk shows you ways to mix Groovy into a legacy Struts application, dramatically reducing both the lines of code and the complexity. We'll also introduce you to Grails (a Groovy-based web framework) whose URL-mapping capabilities allow it to replace your Struts application without breaking legacy URLs.

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.

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 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 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.

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.

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.

Got Guice? by Venkat Subramaniam

In this presentation we will take a look at Google's dependency injection framework, discuss its features, capabilities, strengths, and weakness. We will then discuss where it stands in comparison to Spring.

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.

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.

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.