

New England Software Symposium

Sheraton Framingham

September 12 - 14, 2008

<http://www.nofluffjuststuff.com/conference/boston/2008/09/index.html>

(event schedule as of September 13, 2008)

Fri, Sep. 12, 2008						
	Grand South	Grand North	Ashland	Middlesex	Carlisle	Boardroom
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	The Busy Java Developer's Guide to Debugging Ted Neward	Enterprise Messaging Using JMS (Part 1) Mark Richards	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	JavaServer Faces: A Whirlwind Tour David Geary	Architecture and Agility Are Not Mutually Exclusive David Hussman
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	The Busy Java Developer's Guide to Monitoring Ted Neward	Enterprise Messaging With JMS (Part 2) Mark Richards	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	Facelets explained David Geary	Agile Management & Managing Agility David Hussman
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Rapid Web Development with Grails and Ajax Scott Davis	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	SOA Unplugged Mark Richards	10 ways to use Hibernate effectively Brian Sam-Bodden	Using Ajax4jsf David Geary	Automating Customer Acceptance David Hussman
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Ted Neward					

Sat, Sep. 13, 2008						
	Grand South	Grand North	Ashland	Middlesex	Carlisle	Boardroom
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	The Productive Programmer: Practice (10 Ways to Improve Your Code) Neal Ford	YSlow: Building Your Website for Speed Scott Davis	Viva La Javolution! Brian Sletten	Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Spring 2.5 - Spring without XML Ken Sipe	Advanced Rules Programming with Drools Brian Sam-Bodden	Intro to Seam David Geary	Real World JSON Scott Davis	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Leading Agile Projects: Maintaining Sustainable Agility David Hussman
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	Agile Test Driven Development With Groovy Jeff Brown	Architecture and Scaling Ken Sipe	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Spring and JMS: Message-Driven POJOs Mark Richards	Give it a REST Brian Sletten	Code Metrics & Analysis for Agile Projects Neal Ford
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	"Design Patterns" in Dynamic Languages Neal Ford	The Java Memory Model Brian Goetz	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Java Persistence: Approaching the Silver Bullet Mark Richards	RESTlet for the Weary Brian Sletten	10 Tips for Getting Your Project Back on Track Jared Richardson
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION					

Sun, Sep. 14, 2008						
	Grand South	Grand North	Ashland	Middlesex	Carlisle	Boardroom
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Test Driven Design Neal Ford	Structuring concurrent applications in JDK 5.0 Brian Goetz	Distributed Teams: Remote Agility Jared Richardson	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Designing for Ajax, part 1 Nathaniel Schutta	Developing Web Services Quickly using GroovyWS Mark Johnson
10:30 - 11:00 AM	MORNING BREAK					
11:00 - 12:30 PM	SOAs Challenges Ken Sipe	Effective Concurrent Java Brian Goetz	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Advanced Web Development With Grails Jeff Brown	Designing for Ajax, part 2 Nathaniel Schutta	Getting Started with BPEL Mark Johnson
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	Squashing bugs with FindBugs Brian Goetz	The Busy Java Developer's Guide to Annotations Ted Neward	Techniques 2008 Jared Richardson	Promoted to Technical Lead - Now what do I do? Mark Johnson	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	Grails Plugins: A Free Library of Functionality Kenneth Kousen
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Hacking - The Dark Arts Ken Sipe	The Busy Developer's Guide to Scala Ted Neward	Agile Software Testing Strategies Jared Richardson	Software Development Risk Analysis techniques Mark Johnson	Pragmatic Usability (aka, Software Engineer's Guide to Usability) Nathaniel Schutta	How Groovy Helps Kenneth Kousen

New England Software Symposium

Sheraton Framingham

September 12 - 14, 2008

<http://www.nofluffjuststuff.com/conference/boston/2008/09/index.html>
(event schedule as of September 13, 2008)

The Java Memory Model by Brian Goetz

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

Structuring concurrent applications in JDK 5.0 by Brian Goetz

JDK 5.0 is a huge step forward in developing concurrent Java classes and applications, providing a rich set of high-level concurrency building blocks.

Effective Concurrent Java by Brian Goetz

The Java programming language has turned a generation of applications programmers into concurrent programmers through its direct support of multithreading. However, the Java concurrency primitives are just that: primitive. From them you can build many concurrency utilities, but doing so takes great care as concurrent programming poses many traps for the unwary.

Squashing bugs with FindBugs by Brian Goetz

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Are you using static analysis as part of your QA process? If not, you're probably missing out on some bugs that can be caught before they bite your customers.

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.

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.

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*

Viva La Javolution! by Brian Sletten

You're a good Java programmer. You understand the JDK libraries and how to use them. The problem is that many fundamental APIs don't take the bigger performance picture in mind. Garbage collection can end up killing your app if you aren't careful. Concurrency problems and contention can keep your well-intentioned software from leveraging modern hardware architecture that support multi-core and multi-cpu systems. Who knew that simply using the standard library code the way it was designed was opening you up for performance problems in your apps? Don't worry, Javolution has your back.

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.

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. **Prerequisite:** *Give it a REST (unless you are very comfortable with REST)*

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

Facelets explained by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet.

Using Ajax4jsf by David Geary

Ajax4jsf makes it very easy to add Ajax to your JSF applications. Come to this presentation to see how.

Intro to Seam by David Geary

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0.

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

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.

Agile Management & Managing Agility by David Hussman

Management and agility are not mutually exclusive. Many managers are already working in an agile manner as a means to improve, produce, or simply survive. Other managers hear about projects using agile methods and struggle to find a place in the project community. This session provides a new way to think about managing projects. Some managers will find that their existing practices and skills are supported and enhanced by the forums and metrics provided within an agile project while others will be challenged by some of the principles and practices.

Automating Customer Acceptance by David Hussman

Why should the value of test driven development (TDD) stay stuck in the realm of coding? The ideas behind TDD are now being successfully applied to the automation of business value. While this has been going on for some time within the agile community, it is not starting to spread to main stream development. There are more tools are coming available everyday which allow developers, testers, and customers (or product owners) to work together to automate acceptance tests. This process helps clarify the needs of the end user before development begins and removes more of the wasteful work based on incorrect assumptions from vague requirements.

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*

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.

Distributed Teams: Remote Agility by Jared Richardson

How do you keep a team scattered across time zones in sync?

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.

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.

Agile Software Testing Strategies by Jared Richardson

Creating and maintaining a solid automated test suite is critical to an Agile strategy, but often we're just told to "Do it." In this talk we'll look at several pragmatic strategies for creating and building your suite.

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*

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

SOAs Challenges by Ken Sipe

SOA... Is it hype? What's real... and what's not? What is the right abstraction level?

Hacking - The Dark Arts by Ken Sipe

A live Hacking demonstration exposing the tools and techniques used by Hackers.

Grails Plugins: A Free Library of Functionality by Kenneth Kousen

One of the best features of the Grails platform is that its modular architecture makes developing plugins very easy. As a result, many, many plugins have already been developed, with the number growing almost daily.

How Groovy Helps by Kenneth Kousen

A good way to understand how a language works is to collect examples where it is used in practice. That makes it easy to see which features of the language are likely to arise naturally to simplify your work.

Developing Web Services Quickly using GroovyWS by Mark Johnson

This session will explore GroovyWS as a tool to quickly produce and or consume a web service. Web Service testing becomes much easier without the need to purchase expensive testing tools using the GroovyWS framework.

Getting Started with BPEL by Mark Johnson

With all of these web services becoming available there is an increasing need for tools to pull together multiple web services into one composite service. BPEL is an up and coming approach to orchestrating a workflow consisting of Web Service calls.

Promoted to Technical Lead - Now what do I do? by Mark Johnson

The job of tech lead/development manager seemed so easy before. Now you are in the hot seat. You are responsible for getting the new product/project out the door on time and under budget. Your team, management, and finance departments are counting on you to make the right decisions. So what do you do?

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.

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 JMS providers, 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, load balancing, 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*

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?

Spring and JMS: Message-Driven POJOs by Mark Richards

The Java Message Service (JMS) provides an standard messaging API that allows you to send and receive messages using a variety of messaging providers (including Java EE application servers). The Spring Framework takes this abstraction one step further by providing an robust JMS messaging framework that greatly simplifies message processing. In this session we will see how to use the JMS Messaging Framework provided in Spring 2.5. I will start by describing Spring's overall messaging architecture and how to configure the various beans needed for messaging. Then, through interactive coding I will discuss and demonstrate Spring's JMS Template. which is used for sending messages and receiving messages synchronously. I will then discuss and demonstrate Message Driven POJOs, which are Spring's answer for

asynchronous message listeners. After attending this session you will have all the necessary knowledge and code examples to use JMS in your Spring applications. **Prerequisite:** *Knowledge of JMS and Spring*

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.

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 decide 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.

Pragmatic Usability (aka, Software Engineer's Guide to Usability) by Nathaniel Schutta

While some companies have the luxury of a full time usability team, most of us have to make do on our own. Sure, it might be easier (and more comfortable) to focus on all the hip back end goodness, but if your user interface makes users yack, your product is doomed.

The Productive Programmer: Practice (10 Ways to Improve Your Code) by Neal Ford

No one writes perfect code: even the best developers fall into bad habits and traps. These topics from The Productive Programmer illustrate blind spots and help you write better code.

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.

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

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.

Rapid Web Development with Grails and Ajax by Scott Davis

Grails is a Java- and Groovy-based web framework that is built for speed. First-time developers are amazed at how quickly you can get a page-centric MVC web site up and running thanks to the scaffolding and convention over configuration that Grails provides. Advanced web developers are often pleasantly surprised at how easy it is to break out of that coarse-grained navigation model using the native Ajax support baked into the framework.

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 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.

Why the Next Five Years Will Be About Languages by Ted Neward

Thanks to the plateau of per-chip performance increases and the resulting need to work better with multi-core CPUs, the relative difficulty of mapping user requirements to general-purpose programming languages, the emergence of language-agnostic "virtual machines" that abstract away the machine, the relative ceiling of functionality we're finding on the current crop of object-oriented languages, and the promise and power of productivity of dynamically-typed or more loosely-typed languages, we're about to experience a renaissance of innovation in programming languages.

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 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.