

Research Triangle Software Symposium

Marriott RTP

June 20 - 22, 2008

<http://www.nofluffjuststuff.com/conference/raleigh/2008/06/index.html>

(event schedule as of June 21, 2008)

Fri, Jun. 20, 2008					
	Salon A	Salon B	Salon C	Raleigh	Durham
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Structuring concurrent applications in JDK 5.0 Brian Goetz	JavaServer Faces: A Whirlwind Tour David Geary	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	10 Tips for Getting Your Project Back on Track Jared Richardson	Evolutionary SOA Neal Ford
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Effective Concurrent Java Brian Goetz	Facelets David Geary	Spring 2.5 - Spring without XML Ken Sipe	Techniques 2008 Jared Richardson	Test Driven Design Neal Ford
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Beyond ACID: transactions management, in theory and practice Brian Goetz	Know your Java? Venkat Subramaniam	Architecture and Scaling Ken Sipe	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Regular Expressions in Java Neal Ford
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, Jun. 21, 2008					
	Salon A	Salon B	Salon C	Raleigh	Durham
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	"Design Patterns" in Dynamic Languages Neal Ford	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Caring about your Code Quality Venkat Subramaniam	Restoring Agility: Getting Your Team Back on Track Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	The Java Memory Model Brian Goetz	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Give it a REST Brian Sletten	Build Teams, Not Products Jared Richardson
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Design Patterns in Java and Groovy Venkat Subramaniam	Real World JSON Scott Davis	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	RESTlet for the Weary Brian Sletten	Software Development Risk Analysis techniques Mark Johnson
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	DSL in Groovy Venkat Subramaniam	7 Habits of Highly Effective Developers Ken Sipe	Advanced Rules Programming with Drools Brian Sam-Bodden	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Code Metrics & Analysis for Agile Projects Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION				

Sun, Jun. 22, 2008					
	Salon A	Salon B	Salon C	Raleigh	Durham
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	FP on JVM Venkat Subramaniam	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	A Thorough Introduction To Groovy Jeff Brown	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Software Project Estimation Mark Johnson
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	SOAs Challenges Ken Sipe	10 ways to use Hibernate effectively Brian Sam-Bodden	Introduction to JRuby Neal Ford	YSlow: Building Your Website for Speed Scott Davis	Getting to Acceptance: Validating your requirements with FitNesse Mark Johnson
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Monitoring Software Quality with Continuous Integration Andrew Glover	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Intermediate Maven David Bock	Got Guice? Venkat Subramaniam	Promoted to Technical Lead. Now what do I do? Mark Johnson
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Tactical Continuous Integration with Hudson Andrew Glover	Advanced Web Development With Grails Jeff Brown	Hacking - The Dark Arts Ken Sipe	Practices of an Agile Developer Venkat Subramaniam	How to Fail with 100% Test Coverage Jason Rudolph

Research Triangle Software Symposium

Marriott RTP

June 20 - 22, 2008

<http://www.nofluffjuststuff.com/conference/raleigh/2008/06/index.html>
(event schedule as of June 21, 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.

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.

Beyond ACID: transactions management, in theory and practice by Brian Goetz

Transactions are the software building blocks of enterprise applications, but not all transactional systems are created equally. This talk covers the basics of what transactions are, why they are essential to building reliable enterprise software, the fundamental properties of transactions, and how transactions are supported and implemented in popular frameworks such as Java EE and Spring.

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.

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.

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.

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.

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 by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

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.

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.

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.

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.

Build Teams, Not Products by Jared Richardson

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

How to Fail with 100% Test Coverage by Jason Rudolph

With an expressive language such as Ruby and with modern test practices, 100% C0 test coverage is readily achievable. But 100% coverage is meaningless without other supporting habits and practices. Over the last few years, we have taken dozens of projects to 100% coverage, and there are still plenty of things that can go wrong.

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.

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?

7 Habits of Highly Effective Developers by Ken Sipe

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

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.

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.

Software Project Estimation by Mark Johnson

As developers we dread when management requests a project estimate. Typically, you do not have the opportunity to understand all the requirements, the team composition is unknown, and you have been given until tomorrow end of day to produce an estimate. Several months later everyone is yelling at you about the software estimation errors encountered during the project.

Getting to Acceptance: Validating your requirements with FitNesse by Mark Johnson

How do you know when you are ?Done? and the assignment is complete. Well of course you are done when your requirements are complete. But it always happens that your interpretation differs from the customer/management?s interpretation.

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

You have just received the much desired promotion to Technical? The team is waiting your direction? what should you do now?

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?

"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.

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.

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.

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.

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.

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.

Know your Java? by Venkat Subramaniam

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

Caring about your Code Quality by Venkat Subramaniam

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

Design Patterns in Java and Groovy by Venkat Subramaniam

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

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.

FP on JVM by Venkat Subramaniam

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

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.

Practices of an Agile Developer by Venkat Subramaniam

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speakers will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the 2007 Jolt productivity award winning book with the same title as the talk.