

# Pacific Northwest Software Symposium

Redmond Marriott Town Center

September 19 - 21, 2008

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

(event schedule as of September 17, 2008)

Fri, Sep. 19, 2008					
	1	2	3	4	5
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	SOA Unplugged Mark Richards	Give it a REST Brian Sletten	JavaServer Faces: A Whirlwind Tour David Geary	A Thorough Introduction To Groovy Jeff Brown	10 Tips for Getting Your Project Back on Track Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Enterprise Messaging Using JMS (Part 1) Mark Richards	RESTlet for the Weary Brian Sletten	Facelets David Geary	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Techniques 2008 Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Enterprise Messaging With JMS (Part 2) Mark Richards	Resource-Oriented Computing w/ NetKernel : Software for the 21st Century Brian Sletten	Using Ajax4jsf David Geary	Advanced Web Development With Grails Jeff Brown	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Jared Richardson				

Sat, Sep. 20, 2008					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	EJB3 Core Specification (JSR-220) Mark Richards	Viva La Javolution! Brian Sletten	"Design Patterns" in Dynamic Languages Neal Ford	Agile Test Driven Development With Groovy Jeff Brown	Restoring Agility: Getting Your Team Back on Track Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Java Persistence: Approaching the Silver Bullet Mark Richards	Regular Expressions in Java Neal Ford	Intro to Seam David Geary	Powerful Metaprogramming Techniques With Groovy Jeff Brown	Build Teams, Not Products Jared Richardson
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Transaction Design Patterns Mark Richards	Towards an Evolutionary Design Venkat Subramaniam	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Code Metrics & Analysis for Agile Projects Neal Ford
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	The Busy Java Developer's Guide to Performance and Scalability Ted Neward	Know your Java? Venkat Subramaniam	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Garbage-collector-friendly programming Brian Goetz	Test Driven Design Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION				

Sun, Sep. 21, 2008					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	The Java Memory Model Brian Goetz	The Productive Programmer: Mechanics Neal Ford	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Essence of Agility Venkat Subramaniam
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Squashing bugs with FindBugs Brian Goetz	The Busy Java Developer's Guide to ClassLoaders Ted Neward	MOPping Up Groovy Venkat Subramaniam	Intermediate Maven David Bock	Agile Project Management (With Just a Bit About Mingle) Neal Ford
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	The Busy Java Developer's Guide to Annotations Ted Neward	Introduction to JRuby Neal Ford	Structuring concurrent applications in JDK 5.0 Brian Goetz	Caring about your Code Quality Venkat Subramaniam	Estimating vs. Guessing - How Agile Teams Estimate Their Work David Bock
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Meta-programming JRuby for Fun & Profit Neal Ford	Effective Concurrent Java Brian Goetz	FP on JVM Venkat Subramaniam	Surviving Middle Management David Bock

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

### Friday, Sep. 19

12:00 - 1:00 PM : REGISTRATION

1:00 - 1:15 PM : WELCOME

1:15 - 2:45 PM - Sessions

#### Session #1 : 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?

#### Session #2 : 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.

#### Session #3 : 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.*

#### Session #4 : 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.

#### Session #5 : 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.

2:45 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

#### Session #6 : 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.

#### Session #7 : 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)*

#### Session #8 : 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.*

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

### **Session #9 : 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.

### **Session #10 : 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.

4:45 - 5:00 PM : BREAK

5:00 - 6:30 PM - Sessions

### **Session #11 : 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*

### **Session #12 : Resource-Oriented Computing w/ NetKernel : Software for the 21st Century by Brian Sletten**

Imagine the simplicity of REST married to the power of Unix pipes with the benefits of a loosely-coupled, logically-layered architecture. If that is hard to imagine, it may be because the architectures available to you today are convoluted accretions of mismatched technologies, languages, abstractions and data models. NetKernel is a disruptive technology that changes the game. It has been quietly gaining mind share in the past several years; people who are exposed to it don't want to go back to the tired and blue conventions of J2EE and .NET. Not only does it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. Come see how this open source / commercial product can change the way you think about building software.

### **Session #13 : Using Ajax4jsf by David Geary**

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

### **Session #14 : 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*

### **Session #15 : 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.

6:30 - 7:15 PM : DINNER

Keynote: Career 2.0: Take Control of Your Life - Jared Richardson

## **Saturday, Sep. 20**

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### **Session #16 : EJB3 Core Specification (JSR-220) by Mark Richards**

EJB3 (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will explore in detail some of the new features of the core EJB 3 specification. Included in this session will be a hands-on discussion and demonstration of session beans, dependency injection, interceptors (aop), and Message-Driven Beans (MDB). For the interceptors discussion I will be showing how to define interceptors for enabling a method trace, mocking objects, and sending JMS message notifications to be later picked up by the MDBs I will be creating. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. Note: this session does not cover the new Java Persistence API (JPA) - only the core specification.

### **Session #17 : 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

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

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.

### **Session #18 : "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.

### **Session #19 : 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.

### **Session #20 : 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.

10:30 - 11:00 AM : BREAK

11:00 - 12:30 PM - Sessions

### **Session #21 : 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.

### **Session #22 : 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.

### **Session #23 : 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.

### **Session #24 : Powerful Metaprogramming Techniques With Groovy by Jeff Brown**

Metaprogramming is a key component in building truly dynamic and flexible applications with Groovy. Groovy's metaprogramming capabilities bring great new possibilities to the table that would be very difficult or just plain impossible to write with Java alone. This session will demystify a lot of the magic that seems to be going on inside of a Groovy application. **Prerequisite:** *A Thorough Introduction To Groovy*

### **Session #25 : Build Teams, Not Products by Jared Richardson**

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

12:30 - 1:30 PM : LUNCH

1:30 - 3:00 PM - Sessions

### **Session #26 : 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.

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

### **Session #27 : Towards an Evolutionary Design by Venkat Subramaniam**

A good design is critical for success with agile development. That does not mean a big up-front design. The design has to be evolutionary. However, the design you evolve must be extensible and maintainable. After all, you can't be agile if your design sucks.

### **Session #28 : 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.

### **Session #29 : 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.

### **Session #30 : 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.

3:00 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

### **Session #31 : The Busy Java Developer's Guide to Performance and Scalability by Ted Neward**

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

### **Session #32 : 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.

### **Session #33 : 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.

### **Session #34 : Garbage-collector-friendly programming by Brian Goetz**

To many developers, garbage collection is black magic. Accordingly, there are a lot of conflicting advice about what is good or bad for the garbage collector. In this talk, I look at how garbage collection is implemented in the HotSpot VM, and techniques for writing programs that exhibit good garbage collection behavior. Surprisingly, many of these techniques coincide with writing good, clean code.

### **Session #35 : 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.

4:45 - 5:30 PM : BIRDS OF A FEATHER SESSION

## **Sunday, Sep. 21**

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### **Session #36 : 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.

### **Session #37 : The Productive Programmer: Mechanics by Neal Ford**

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

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

### **Session #38 : The Busy Java Developer's Guide to Java Platform Security by Ted Neward**

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

### **Session #39 : 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.

### **Session #40 : Essence of Agility by Venkat Subramaniam**

Begin agile is more than saying your organization is committed to being agile or your team is agile.

10:30 - 11:00 AM : MORNING BREAK

11:00 - 12:30 PM - Sessions

### **Session #41 : 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.

### **Session #42 : The Busy Java Developer's Guide to ClassLoaders by Ted Neward**

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

### **Session #43 : MOPping Up Groovy by Venkat Subramaniam**

TBD

### **Session #44 : 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.

### **Session #45 : Agile Project Management (With Just a Bit About Mingle) by Neal Ford**

You can read books about Agile projects, but you must consult real-world experience to really understand the dynamics of agile project management. This session discusses agile management topics including estimation, project tracking, and useful metrics (and how to obtain them). And just a little about Mingle, the agile project tracking tool from ThoughtWorks.

12:30 - 1:15 PM : LUNCH

1:15 - 2:15 PM : EXPERT PANEL DISCUSSION

2:15 - 3:45 PM - Sessions

### **Session #46 : 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).

# Pacific Northwest Software Symposium

## -Session Schedule-

(event schedule as of September 17, 2008)

### **Session #47 : 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.

### **Session #48 : 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.

### **Session #49 : 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.

### **Session #50 : Estimating vs. Guessing - How Agile Teams Estimate Their Work by David Bock**

Estimating is regarded as little more than 'educating guessing', but so much can hang on the quality of those estimates. With good estimates we can set clear expectations for project delivery, but with bad estimates we can run over schedule and over budget, or worse. We often estimate when we know the least about the work that needs to get done - so how can we make the best of what is potentially a bad situation?

3:45 - 4:00 PM : BREAK

4:00 - 5:30 PM - Sessions

### **Session #51 : 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.

### **Session #52 : 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.

### **Session #53 : 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.

### **Session #54 : 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.

### **Session #55 : 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?