

# Lone Star Software Symposium: Austin

Marriott Austin Airport Hotel

July 11 - 13, 2008

<http://www.nofluffjuststuff.com/conference/austin/2008/07/index.html>

(event schedule as of July 9, 2008)

Fri, Jul. 11, 2008						
	Salon A&B	Salon C	Salon D	Pecan	Bluebonnet	Limestone
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	A Thorough Introduction To Groovy Jeff Brown	Evolutionary SOA Neal Ford	Give it a REST Brian Sletten	Know your Java? Venkat Subramaniam	JavaServer Faces: A Whirlwind Tour David Geary	10 Tips for Getting Your Project Back on Track Jared Richardson
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Agile Test Driven Development With Groovy Jeff Brown	Productive Programmer: Acceleration & Automation Neal Ford	RESTlet for the Weary Brian Sletten	Got Guice? Venkat Subramaniam	Facelets David Geary	Shippers Unite! Jared Richardson
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Powerful Metaprogramming Techniques With Groovy Jeff Brown	Productive Programmer: Canonicity & Focus Neal Ford	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Acceptance Testing Application Behavior Venkat Subramaniam	Seam David Geary	Build Teams, Not Products Jared Richardson
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Neal Ford					

Sat, Jul. 12, 2008						
	Salon A&B	Salon C	Salon D	Pecan	Bluebonnet	Limestone
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Simplifying Enterprise Applications with Spring, Part 1 Ramnivas Laddad	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Tool support for Agile Databases: Introducing Liquibase John Heintz	Test Driven Design Neal Ford	Viva La Javolution! Brian Sletten	Restoring Agility: Getting Your Team Back on Track Jared Richardson
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Simplifying Enterprise Applications with Spring, Part 2 Ramnivas Laddad	Advanced Web Development With Grails Jeff Brown	Adding Behavior to Java Annotations John Heintz	Achieving greater code reuse through decoupling of feature implementations from the domain Eitan Suez	Java Concurrency Idioms Alex Miller	Techniques 2008 Jared Richardson
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	The Busy Java Developer's Guide to Performance and Scalability Ted Neward	Enterprise Security with Spring Ramnivas Laddad	DSL in Groovy Venkat Subramaniam	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Java Collections API Alex Miller	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Architecture Enforcement with AOP Ramnivas Laddad	BDD in Java and Groovy Venkat Subramaniam	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Exploring Terracotta Alex Miller	Code Metrics & Analysis for Agile Projects Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION					

Sun, Jul. 13, 2008						
	Salon A&B	Salon C	Salon D	Pecan	Bluebonnet	Limestone
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	"Design Patterns" in Dynamic Languages Neal Ford	New Features in Spring Web Keith Donald	Leveraging annotations with AOP Ramnivas Laddad	The Busy Java Developer's Guide to Debugging Ted Neward	Internationalization and Localization in Java David Bock	Caring about your Code Quality Venkat Subramaniam
10:30 - 11:00 AM	MORNING BREAK					
11:00 - 12:30 PM	Introduction to Hibernate Scott Leberknight	Decorating Web Pages with Ajax using Spring JavaScript Keith Donald	MOPping Up Groovy Venkat Subramaniam	The Busy Java Developer's Guide to Monitoring Ted Neward	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools 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	Real World Hibernate Tips Scott Leberknight	Spring Web Flow 2 Deep Dive Keith Donald	Introduction to JRuby Neal Ford	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Intermediate Maven David Bock	Towards an Evolutionary Design Venkat Subramaniam
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Surviving Middle Management David Bock	JavaServerFaces: The Biggest Loser of Java Web Frameworks? Keith Donald	Meta-programming JRuby for Fun & Profit Neal Ford	The Busy Developer's Guide to Scala Ted Neward	Google Your Domain Objects With Hibernate Search Scott Leberknight	Tools to facilitate Agile Development Venkat Subramaniam

# Lone Star Software Symposium: Austin

Marriott Austin Airport Hotel

July 11 - 13, 2008

<http://www.nofluffjuststuff.com/conference/austin/2008/07/index.html>

(event schedule as of July 9, 2008)

## Java Concurrency Idioms by Alex Miller

This presentation will look at the many new additions in Java 5 and 6 for concurrent programming such as `Atomic`s, `Lock`s, `synchronizers`, and `concurrent collections`. In particular, we will be looking at common concurrency idioms around locking and access to shared state, thread coordination, thread pooling, and work execution. Each of these topics will be presented with code examples demonstrating common idioms and the usage of these new concurrency primitives.

## Java Collections API by Alex Miller

Did you know that Java 5 and 6 added 8 new interfaces and 16 new collection implementations to the JDK, more than doubling the size of the collection API? `Collections 201` gives you an update on all of the interfaces, implementations, and utilities and gives you guidance on picking the perfect collection. In particular, Java 5 introduced a new major collection type `Queue` and a whole new `java.util.concurrent` package with data structures optimized for concurrent use.

## Exploring Terracotta by Alex Miller

Terracotta is an open-source Java clustering technology. It creates a virtual, durable Java heap that is shared across a cluster of Java Virtual Machines. This is done by dynamically instrumenting bytecode at load time to intercept calls to read and write fields, and also to enter and exit monitor locks. Information about these calls is then transmitted to the Terracotta Server (which can also be clustered) and out to other nodes in the cluster as needed. The advantage of this approach is that many Java programs can be clustered without code changes by providing just external Terracotta configuration. Many performance optimizations are performed to minimize communication and locking costs. Terracotta is commonly used for session sharing in web applications, distributed caching, and distributed workflow processing. This presentation will give an overview of the Terracotta technology, how it's implemented, and common use cases that can benefit from the technology. We will look at some code and cluster some Java applications during the presentation.

## 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.

## 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.

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

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

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

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

### **Intermediate Maven by David Bock**

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

### **Surviving Middle Management by David Bock**

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

### **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.*

### **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. Two

frameworks. And then you have to learn to use them together. Why do we have to learn two frameworks just to retrieve "Hello World" from a database and show it in a view. Isn't that crazy? Now you can use one framework, and use one component model. One. Isn't that nice? Seam, a framework built on JSF and EJB3.0, unifies the JSF and EJB component models. Seam is a steam roller, quickly gathering market share among JSF newbies and longtime believers alike. Come see what it's all about. **Prerequisite:** *Some knowledge of JSF is required. If you don't know what a managed bean is, for instance, then attend JSF Whirlwind before this session.*

### **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.

### **Achieving greater code reuse through decoupling of feature implementations from the domain by Eitan Suez**

Over the last few years, I have taken a different approach to building domain-driven software applications.

### **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.

### **Shippers Unite! by Jared Richardson**

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

### **Build Teams, Not Products by Jared Richardson**

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

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

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

### **Techniques 2008 by Jared Richardson**

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

### **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.

### **A Thorough Introduction To Groovy by Jeff Brown**

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

### **Agile Test Driven Development With Groovy by Jeff Brown**

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

### **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*

### **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*

### **Tool support for Agile Databases: Introducing Liquibase by John Heintz**

This presentation introduces and demonstrates Liquibase: a new Java tool to support automating database refactoring and deployment.

### **Adding Behavior to Java Annotations by John Heintz**

Java's Annotations provide a way to add data to program elements. Annotations are used to configure containers, describe persistence configuration, set security roles, and are defined by nearly every recent JSR standard. This presentation explains the processing options available for consuming Annotations and demonstrates the techniques with live code demonstrations.

### **New Features in Spring Web by Keith Donald**

Spring has a number of interesting modules for web application development, including Spring Web MVC, Spring Web Flow, Spring JavaScript, and Spring Faces. This session will provide an overview of these modules and show how they relate to one another. By the end of this session, you'll understand how Spring simplifies the development and deployment of rich web applications. You'll also gain a glimpse into the roadmap for Spring Web 3.0.

### **Decorating Web Pages with Ajax using Spring JavaScript by Keith Donald**

Spring JavaScript is a JavaScript abstraction framework that allows you to progressively enhance a web page with behavior. The framework consists of a public JavaScript API along with an implementation that builds on the Dojo Toolkit. Spring.js aims to simplify the use of Dojo for common enterprise scenarios while retaining its full-power for advanced use cases. Come to this session to learn to use Spring.js and Dojo to create compelling user interfaces.

### **Spring Web Flow 2 Deep Dive by Keith Donald**

Web Flow is a Spring Web MVC extension that allows you to define Controllers using a higher-order domain-specific-language. This language is designed to model user interactions that require several requests into the server to complete, or may be invoked from different contexts. This session dives deep into the features of the Web Flow definition language, and illustrates how to use it to create sophisticated controller modules.

### **JavaServerFaces: The Biggest Loser of Java Web Frameworks? by Keith Donald**

The Biggest Loser is a popular fitness TV show where severely overweight participants undergo radical transformations to get their lives back on track. Some might say JavaServerFaces has the reputation of being the severely overweight web framework of Java. Is it possible for JSF to shed the pounds and transform itself in the eyes of Java developers? Come to this session to find out.

### **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.

#### **Productive Programmer: Acceleration & Automation 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!

#### **Productive Programmer: Canonicity & Focus by Neal Ford**

Getting work done in modern office environments is a daunting task. This session tackles 2 of the things that drag down developer productivity: lack of focus and creeping repetition.

#### **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?

#### **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.

#### **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.

#### **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.

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

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

#### **Meta-programming JRuby for Fun & Profit by Neal Ford**

Ruby is the revenge of the Smalltalkers. Not since Smalltalk has a language had such powerful meta-programming facilities. While this may seem like a minor feature, it turns out that surgical meta-programming allows solutions to problems that are clearer, more concise, more maintainable, and take orders of magnitudes fewer lines of code.

#### **Simplifying Enterprise Applications with Spring, Part 1 by Ramnivas Laddad**

Developing enterprise applications ain't easy. You not only have to worry about constantly evolving business logic, but also need to address infrastructure concerns ranging from transaction management and security to manageability and integration with diverse external applications. Spring, the most popular lightweight

enterprise application framework, comes to the rescue by simplifying the common needs of enterprise applications. This session (part 1 of 2) presents the core concepts of the Spring Framework.

### **Simplifying Enterprise Applications with Spring, Part 2 by Ramnivas Laddad**

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

### **Enterprise Security with Spring by Ramnivas Laddad**

Spring Security (formerly known as 'Acegi') enables self-contained, consistent, and extensible solutions for securing your applications. Version 2.0 provides major enhancements including a domain-specific XML namespace, convention-based defaulting, and annotation support. This provides a significantly simpler experience for developers while still supporting the same degree of flexibility.

### **Architecture Enforcement with AOP by Ramnivas Laddad**

Aspect-oriented programming (AOP) is a proven paradigm for enforcing broad organizational policies. In this session, we will explore the definition and enforcement of software architecture policies to help keep a code base clean. We will examine several reusable examples you can apply within your own organization to catch architectural violations.

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

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

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

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

### **Real World Hibernate Tips by Scott Leberknight**

Hibernate is a very powerful object/relational mapping framework. With the vast amount of power also comes the responsibility to choose which features of Hibernate to use and how to use them, as well as things to avoid. We'll look at some real world Hibernate tips and tricks in this session.

### **Google Your Domain Objects With Hibernate Search by Scott Leberknight**

Hibernate is one of the pre-eminent object/relational mapping technologies, but the Hibernate Search project adds full-text search capabilities to an already extremely capable tool to allow you to Google your domain objects.

### **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?

### **The Busy Java Developer's Guide to Hacking (on) the JDK by Ted Neward**

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

### **The Busy 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 ClassLoaders by Ted Neward**

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

### **The Busy Developer's Guide to Scala by Ted Neward**

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.

### **Know your Java? by Venkat Subramaniam**

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

### **Got Guice? by Venkat Subramaniam**

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

### **Acceptance Testing Application Behavior by Venkat Subramaniam**

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

### **DSL in Groovy by Venkat Subramaniam**

DSL or Domain Specific Languages focus on a domain or problem at hand. They're expressive, but their restricted scope keeps them simple and small from the user point of view. However, designing them is not easy. In this presentation we will explore the features of Groovy and show how they can be used to create DSLs.

### **BDD in Java and Groovy by Venkat Subramaniam**

In this presentation we will take a look at what BDD is and look at tools to create them in Java and Groovy.

### **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.

### **MOPping Up Groovy by Venkat Subramaniam**

Groovy's dynamic capabilities makes it an attractive language for Meta Programming. There are several facilities to intercept method calls to do AOP kind of operations. You can also inject methods dynamically.

You can also easily perform method synthesis as well. In this presentation we will take a look at techniques that make Groovy pretty Hip for MOP.

### **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.

### **Tools to facilitate Agile Development by Venkat Subramaniam**

The first item in the Agile Manifesto reads that we must prefer "people and interaction over process and tools." Given a choice between average people with superior tools and superior people with average tools, you are likely to achieve greater success with the latter. However, it is important to be continuous and not be episodic?so you want to get continuous feedback about the state, health, and quality of your code and application. Tools can help us a great deal to realize this and make us productive.