

Pacific Northwest Software Symposium

Redmond Marriott Town Center

April 3 - 5, 2009

<http://www.nofluffjuststuff.com/conference/seattle/2009/04/index.html>

Fri, Apr. 03, 2009

	Salon 3&4	Bear River	Elk River	Lewis River
12:00 - 1:00 PM	REGISTRATION			
1:00 - 1:15 PM	WELCOME			
1:15 - 2:45 PM	Emergent Design & Evolutionary Architecture Neal Ford	The Busy Java Developer's Guide to Java7 Ted Neward	The Amazing Groovy Weight-loss Plan Scott Davis	JSF 2.0: An Introduction David Geary
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Real-world Refactoring Neal Ford	The Busy Java Developer's Guide to Collections Ted Neward	Groovy XML Ninja Skills Scott Davis	JSF 2.0: Advanced Topics David Geary
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Test Driven Design Neal Ford	The Busy Java Developer's Guide to the OpenJDK Ted Neward	DSLs in Groovy: Say What You Mean, Mean What You Say Scott Davis	Flex for Java Developers David Geary
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: by Neal Ford			

Sat, Apr. 04, 2009

	Salon 3&4	Bear River	Elk River	Lewis River	Alder Room
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	So you want to be an Architect Ken Sipe	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Lizard Brain Web Design Scott Davis	GWT: An Introduction David Geary	
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Architecture: Non-Functional Requirements Ken Sipe	The Busy Java Developer's Guide to Advanced Platform Security Ted Neward	Web 2.0 Checklist: Deconstructing Modern Websites Scott Davis	GWT: Advanced Topics David Geary	
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM		The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward	Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications Scott Davis	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	What's New in Spring 3 Ken Sipe
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM		The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward	Git Going with Distributed Version Control Matthew McCullough	Advanced Rules Programming with Drools Brian Sam-Bodden	Hacking - The Dark Arts Ken Sipe
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSION				

Sun, Apr. 05, 2009

	Salon 3&4	Bear River	Elk River	Lewis River
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	7 Habits of Highly Effective Developers Ken Sipe	Architect for Scale Michael Nygard	10 ways to use Hibernate effectively Brian Sam-Bodden	Mastering Maven 2.0 Matthew McCullough
10:30 - 11:00 AM	MORNING BREAK			
11:00 - 12:30 PM	Iteration 0 Ken Sipe	The 90-Minute Startup Michael Nygard	Enterprise Integration Patterns with Camel and Mule Brian Sam-Bodden	The Busy Java Developer's Guide to Hacking with the JDK Ted Neward
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	Java Memory, Performance and the Garbage Collector Ken Sipe	Clouds, Grids, and Fog Michael Nygard	Refactoring the Enterprise with JRuby Brian Sam-Bodden	Open Source Debugging Tools & Recipes Matthew McCullough
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	Debugging your Production JVM Ken Sipe	The Busy Java Developer's Guide to Scala: Scala + Services Ted Neward	Increasing your Eclipse Productivity Brian Sam-Bodden	iPhone Objective-C integration to Java Web Services Matthew McCullough

Pacific Northwest Software Symposium

Redmond Marriott Town Center

April 3 - 5, 2009

<http://www.nofluffjuststuff.com/conference/seattle/2009/04/index.html>

The Amazing Groovy Weight-loss Plan by Scott Davis

"The central enemy of reliability is complexity." (Dr. Daniel Geer) Java is a powerful programming language. A smart developer can do nearly anything with Java. So the next question is, "How quickly can it be done? How many lines of code does it take to do common tasks?" Groovy greases the wheels of Java by decreasing the complexity of the language while preserving the raw power. At first glance, you might think that this talk is simply about how Groovy drastically reduces the lines of code you need to write. What this talk is really about is bringing simplicity, clarity, readability, and yes, beauty to your source code.

Groovy XML Ninja Skills by Scott Davis

"XML is like violence: if it doesn't solve your problem, you aren't using enough of it." (Anonymous) XML is everywhere. Whether you are dealing with local configuration files (web.xml, struts-config.xml) or remote web services (SOAP, REST, RSS, Atom), the modern software developer needs to be able to request, slice, and dice XML with ease. That requires a set of razor-sharp tools that reduce the inherent complexity of the problem, not multiply it. Once you see XML tremble in fear at the awesome power of Groovy, you'll wonder what you ever did without it.

DSLs in Groovy: Say What You Mean, Mean What You Say by Scott Davis

"Simplicity is the ultimate sophistication." (Leonardo da Vinci) The history of computer programming has been bridging the gap between what the user says ("We need to add sales tax to each item in the order") and what the programming language requires you to say ("for Iterator i = orderList.iterator();"). Building Domain Specific Languages (DSLs) allow you to express the solution in the language of the domain user instead of the language of the programmer.

Lizard Brain Web Design by Scott Davis

"There's an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone." (Bjarne Stroustrup) The "lizard brain" is the oldest part of the human brain -- the part responsible for autonomic functions like breathing, heart rate, and navigating websites. OK, maybe not that last part, but your website should be easy to use. Stupid easy. Lizard brain easy. Any time your user spends figuring out how to do something -- even for a split second -- is wasted time due to poor design. Inspired by Steve Krug's book "Don't Make Me Think", this talk answers the question, "Why is that website so hard to use?"

Web 2.0 Checklist: Deconstructing Modern Websites by Scott Davis

"The challenge of modernity is to live without illusions and without becoming disillusioned." (Antonio Gramsci) There are plenty of sarcastic "Web 2.0" checklists out there -- be perpetually in BETA, when in doubt add rounded corners, etc. While we can all laugh at the superficial aspects of the Web 2.0 revolution, there are plenty of serious aspects to it as well. Is your website mash-up friendly or hostile? Do you tell your visitors when things change (via RSS or Atom syndication), or do you expect them to check in daily for updates? Is your website a silo or a part of a larger ecosystem?

Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications by Scott Davis

"The proof of the pudding is in the eating. By a small sample we may judge of the whole piece." (Miguel de Cervantes Saavedra) Most Grails tutorials demonstrate how easy it is to build simple CRUD (Create/Retrieve/Update/Delete) applications. While skinning a database with a web front-end is undeniably one beneficial aspect of Grails, it isn't the only thing Grails is good for. As you'll see here, Grails can be used to build a wide variety of web applications. You won't see a single HTML table with "edit" and "delete" links, I promise.

Emergent Design & Evolutionary Architecture by Neal Ford

Most of the software world has realized that BDUF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent design & evolutionary architecture, including both proactive (test-driven development) and reactive (refactoring, composed method) approaches to discovering design. The goal of

this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed, keeping you code in sync with the problem domain.

Real-world Refactoring by Neal Ford

Refactoring is a fine academic exercise in the perfect world, but we don't really live there. Even with the best intentions, projects build up technical debt and crufty bad things. This session covers refactoring in the real world, at both the atomic level (how to refactor towards composed method and the single level of abstraction principle) to larger project strategies for multi-day refactoring efforts. This talk provides practical strategies for real projects to effectively refactor your code.

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.

Keynote: On the Lam from the Furniture Police by Neal Ford

When you were hired by your current employer, you may think it's because of your winning personality, your dazzling smile, or your encyclopedic knowledge of [insert technology here]. But it's not. You were hired for your ability to sit and concentrate for long periods of time to solve problems, then placed in an environment where it's utterly impossible to do that! Who decides that, despite overwhelming evidence that it's bad for productivity and people hate it, that you must sit in a cubicle? The furniture police! This keynote describes the frustrations of modern knowledge workers in their quest to actually get some work done, and solutions for how to gird yourself against all those distractions. I talk about environments, coding, acceleration, automation, and avoiding repetition as ways to defeat the mid-guided attempts to sap your ability to produce good work. And I give you ways to go on the lam from the furniture police and ammunition to fight back!

JSF 2.0: An Introduction by David Geary

This session introduces JSF 2.0 fundamentals, with emphasis on new features in JSF 2.0. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks*

JSF 2.0: Advanced Topics by David Geary

This session covers advanced aspects of JSF 2.0. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks. Familiarity with Ajax. This session builds on demos shown in the JSF 2.0 Introduction talk, so it is helpful, although not required, if you attend the intro talk before coming to this session.*

Flex for Java Developers by David Geary

An introduction to Flex for Java developers. **Prerequisite:** *Familiarity with Flex and at least one other web application framework*

GWT: An Introduction by David Geary

An introduction to Google Web Toolkit. **Prerequisite:** *Familiarity with a component-based framework, preferably a desktop application framework*

GWT: Advanced Topics by David Geary

Learn to do really cool stuff with GWT. **Prerequisite:** *The GWT: Introduction session is not a prerequisite for this session, but it will help if you have some familiarity with GWT.*

Git Going with Distributed Version Control by Matthew McCullough

Many development shops have made the leap from RCS, Perforce, ClearCase, PVCS, CVS, BitKeeper or SourceSafe to the modern Subversion (SVN) version control system. But why not take the next massive stride in productivity and get on board with Git, a distributed version control system (DVCS). Jump ahead of the masses staying on Subversion, and increase your team's productivity, debugging effectiveness, flexibility in cutting releases, and repository redundancy at \$0 cost. Understand how distributed version control systems are game-changers and pick up the lingo that will become standard in the next few years. **Prerequisite:** *Basic understanding of Subversion or similar version control system*

Mastering Maven 2.0 by Matthew McCullough

Maven has been on the Java build tools scene for quite a number of years, but the adoption rate in enterprises is now going through the roof. Maven can seem daunting, but this presentation will equip existing Maven users with more efficient techniques and tools to overcome the biggest perceived Maven hurdles and build issues with ease. We'll examine tools to help you find artifacts in central repositories, manage your corporation's internal Maven artifacts with a proxy tool such as Nexus, view and override dependency graphs, dependency management and multi-module best practices, create OS specific profiles, and leverage the latest Maven plugins for the top Java IDEs. **Prerequisite:** *Basic Maven knowledge*

Open Source Debugging Tools & Recipes by Matthew McCullough

Open Source is not just a suite of libraries you consume within your application, but now reaches into the space of tools to help you troubleshoot and improve your applications. This session will quickly survey a wide range of tools across the Java, Networking, Filesystem, SOAP, REST, HTML, CSS and JavaScript realms. We'll look at applications such as VisualVM, which help you analyze your heap and garbage collection cycles of both local and remote applications. Performance and load testing tools such as JMeter will expose bottlenecks, threading, and scalability concerns of everything from Java modules to Web Apps (even ones that don't use any Java). **Prerequisite:** *Basic knowledge of web services, core Java, web application design.*

iPhone Objective-C integration to Java Web Services by Matthew McCullough

iPhone development is all the rage both in the mobile entertainment, social networking, and productivity application spaces. As a Java developer, prepare yourself to be a participant in aspects of this new breed and platform of development. Hop on board with a quick start to iPhone application coding in Objective C and integration with some of our favorite Java web service back-ends such as Axis, JSR311 Jersey, Grails, and more.

The Busy Java Developer's Guide to Java7 by Ted Neward

Even though the Java 7 JSR has yet to be formed, some interesting things are beginning to emerge from Sun about what Java7 may include when its formal release contents are finally made public.

The Busy Java Developer's Guide to Collections by Ted Neward

For so many Java developers, the `java.util.*` package consists of List, ArrayList, and maybe Map and HashMap. But the Collections classes are so much more powerful than many of us are led to believe, and all it requires is a small amount of digging and some simple exploration to begin to "get" the real power of the Collection classes.

The Busy Java Developer's Guide to the OpenJDK by Ted Neward

With the release of the OpenJDK source code, Java developers have been given a unique opportunity to peer inside the hood of the JVM, see what's there, and how Java code actually executes.

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.

The Busy Java Developer's Guide to Advanced Platform Security by Ted Neward

So you know the platform security model, and now you want to use it in new and interesting ways, like creating a custom Policy implementation, a custom Permission, or create a custom security context in which code will execute. Perhaps you even wish to make certain objects accessible only to those with the right permissions, or cryptographic key. Nothing could be easier, despite Java security's reputation as a dark and arcane place. **Prerequisite:** *The Busy Java Developer's Guide to Platform Security*

The Busy Java Developer's Guide to Concurrency (Part 1: Threads) by Ted Neward

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the `java.util.concurrent` packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown. **Prerequisite:** *The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

The Busy Java Developer's Guide to Hacking with 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 Scala: Scala + Services 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. Sporting the usual object-oriented concepts as classes and inheritance, Scala also offers a number of powerful functional features, such as algebraic data types, immutable objects by default, pattern matching, closures, anonymous functions and currying, and more. Combined with some deep support for XML generation and consumption, Scala offers Java programmers an opportunity to write powerful programs with concise syntax for a new decade of Java programming.

Architect for Scale by Michael Nygard

Is your system small, medium, large, or super-size? Is traffic on it's way up? Architecture patterns and structures that work at one scale seldom work across all of them. A communication style that's appropriate for small websites will probably fail badly if you apply it to world-wide networks of computers. Likewise, structures that work for large-scale systems are probably too complex and expensive to be worth it for small sites.

The 90-Minute Startup by Michael Nygard

Cloud computing is taking the world by storm. Amazon's Web Services, EC2, and S3 provide completely virtual infrastructure, letting startup and existing companies create sites and web applications faster than ever before. In this session, Michael will use cloud computing to create and deploy a fully-functional web site. You will learn how to create and run your own virtual infrastructure in the clouds.

Clouds, Grids, and Fog by Michael Nygard

Servers, storage, networking, backups... they're all vanishing into the "clouds". Cloud Computing is the emerging architecture for massive, scalable infrastructure that your company doesn't have to own or operate. From the "zero servers" web startup to the corporate IT department battling server-sprawl, cloud computing has many manifestations. This session will differentiate among the various types of cloud computing and describe applicable use cases.

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*

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.

Enterprise Integration Patterns with Camel and Mule by Brian Sam-Bodden

Integration Patterns Implementations in Camel and Mule

Refactoring the Enterprise with JRuby by Brian Sam-Bodden

Learn how JRuby can bring simplicity to the complex and rich APIs available in the Java platform. In this session you'll learn how to use JRuby to tackle some common tasks in Java SE and Java EE as well as how to abstract and simplify complex APIs.

Increasing your Eclipse Productivity by Brian Sam-Bodden

Mylyn, Eclipse Plugins, Tips and Tricks

So you want to be an Architect by Ken Sipe

This session is a quick look at all aspects of being a corporate software architect. Whether you are a developer looking to move into the role of architect, needing to have an understanding of what is expected or already in the role of software architect looking for new and interesting ideas, this session is for you.

Architecture: Non-Functional Requirements by Ken Sipe

The agile focus of software development puts heavy focus on user requirements through user stories. However we can not lose sight of the non-functional requirements as well. The software could be written to the exact specification and desire of the user, however if it takes 5 minutes for a request response, or it only supports 2 users or it isn't secure, then we still haven't done our jobs as developers.

What's New in Spring 3 by Ken Sipe

The Spring Framework has led the industry in innovation for years. Starting with dependency injection and promoting testing through removal of framework dependencies. Spring 3.0 continues that innovation in a way that takes full advantage of the Java 5 platform. There are a number of significant changes to the framework. So whether you are new to the framework or an experience Spring developer, this is a great session to come up to speed on the latest from SpringSource. **Prerequisite:** Java 5

Hacking - The Dark Arts by Ken Sipe

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

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.

Iteration 0 by Ken Sipe

The success of an Agile / SCRUM project is a successful start. The first interaction is often referred to as iteration 0. Other iterations have a set of stories with clear acceptance criteria which establishes the velocity of the team and its effort. What then is accomplished in iteration 0? How do we get an Agile process started.

Java Memory, Performance and the Garbage Collector by Ken Sipe

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

Debugging your Production JVM by Ken Sipe

So your server is having issues? memory? Connections? Limited response? Is the first solution to bounce the server? Perhaps change some VM flags or add some logging? In today's Java 6 world, with its superior runtime monitoring and management capabilities the reasons to bounce the server have been greatly reduced.