

Research Triangle Software Symposium

Marriott RTP
August 27 - 29, 2010

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Research Triangle Software Symposium on August 27 - 29, 2010. RTSS2010 will be held at the Marriott RTP.

Since 2001, the No Fluff Just Stuff Java Symposium has been regarded as the premier Java/Agility event series anywhere serving over 29,500 attendees with some 190 events. The popularity of the NFJS symposium series can be traced to the following:

- 1). Exceptional Speakers
- 2). Limited Attendance - capped at 250 people
- 3). No Vendors, No Sales Pitches, No Marketecture
- 4). Excellent networking opportunity with speakers and fellow attendees because of small size.
- 5). The Best Value in the Java conferencing space period.

Topics for the 2010 NFJS Tour

Languages on the JVM: Groovy, JRuby, Clojure, Scala
HTML5
Enterprise Java
Core Java, JVM Internals
No SQL: MongoDB
JSF, GWT
Agility
Groovy, Grails, Gradle
REST, RDFa, Resource Oriented Architectures
GIT Version Control
jQuery, Ajax, Flex, RIA
Mobile Applications - iPhone and Android
More...

The Registration Fee Includes:

3 Day All Access Pass to RTSS2010
All Meals/Snacks - duration of the symposium
90 Day IntelliJ license compliments of JetBrains
Session Materials
Custom NFJS Binder
Great Giveaways @ NFJS

Early Bird Registration: \$825/person good thru 8/2/10 after \$925

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 8/2/10

Registration Fees

| Attendees | Before Aug. 2, 2010 | After Aug. 2, 2010 |
|-----------|---------------------|--------------------|
| 5-9 | \$725 | \$825 |
| 10-14 | \$700 | \$800 |
| 15-24 | \$675 | \$775 |
| 25+ | \$650 | \$750 |

Go to <http://www.nofluffjuststuff.com/conference/raleigh/2010/08/home> and register today!

Research Triangle Software Symposium

Marriott RTP
August 27 - 29, 2010

Fri, Aug. 27, 2010

| | Salon A-B | Salon C | Salon D | Raleigh | Durham |
|-----------------|--|---|---|--|--|
| 12:00 - 1:00 PM | REGISTRATION | | | | |
| 1:00 - 1:15 PM | WELCOME | | | | |
| 1:15 - 2:45 PM | Learning Open Source Business Intelligence Tim Berglund | Third time's a charm: What's new in Spring 3.0 Craig Walls | The Busy Java Developer's Guide to Collections Ted Neward | Grails - How to Build Enterprise Apps Jeff Brown | Encryption on the JVM: Boot Camp Matthew McCullough |
| 2:45 - 3:15 PM | BREAK | | | | |
| 3:15 - 4:45 PM | Open Source Business Intelligence Workshop Tim Berglund | Building RESTful Applications with Spring MVC Craig Walls | The Busy Java Developer's Guide to Functional Java Ted Neward | GORM Inside And Out Jeff Brown | Encryption on the JVM: Advanced Techniques Matthew McCullough |
| 4:45 - 5:00 PM | BREAK | | | | |
| 5:00 - 6:30 PM | Complexity Theory and Software Development Tim Berglund | Open Source Debugging Tools for Java Matthew McCullough | The Busy Java Developer's Guide to Advanced Collections Ted Neward | Building Twitter With Grails In 90 Minutes Jeff Brown | Modular Java: An Introduction to OSGi Craig Walls |
| 6:30 - 7:15 PM | DINNER | | | | |
| 7:15 - 8:00 PM | KEYNOTE: TED NEWARD - ICONOCLASM | | | | |

Sat, Aug. 28, 2010

| | Salon A-B | Salon C | Salon D | Raleigh | Durham |
|------------------|--|--|---|---|---|
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | Hadoop: Divide and Conquer Gigantic Datasets (Intro) Matthew McCullough | Emergent Design Neal Ford | Introducing Spring Roo: From Zero to Working Spring Application in Record Time Craig Walls | Busy Java Developer's Guide to Parallelization Patterns Ted Neward | Gaelyk: Lightweight Groovy on the Google App Engine Tim Berglund |
| 10:30 - 11:00 AM | BREAK | | | | |
| 11:00 - 12:30 PM | Evolving towards REST-based Enterprise Integration Neal Ford | Decision Making in Software Teams Tim Berglund | Hadoop: Divide and Conquer Gigantic Datasets (Advanced) Matthew McCullough | The Busy Java Developer's Guide to Scala: Basics Ted Neward | Aspect Oriented Programming With Spring AOP Jeff Brown |
| 12:30 - 1:30 PM | LUNCH | | | | |
| 1:30 - 3:00 PM | Agile Engineering Practices Neal Ford | Java Memory, Performance and the Garbage Collector Ken Sipe | Pragmatic Architecture Ted Neward | Migrating to Maven 3.0 Matthew McCullough | Compile Time and Runtime Metaprogramming With Groovy Jeff Brown |
| 3:00 - 3:15 PM | BREAK | | | | |
| 3:15 - 4:45 PM | HTML 5 ... and the Kitchen Sink Brian Sletten | Testing the Entire Stack Neal Ford | Architectural Kata Workshop Part I Ted Neward | Enterprise JPA & Spring 3.0 - Tips and Tricks for JEE Persistence Pratik Patel | Debugging your Production JVM Ken Sipe |
| 4:45 - 5:45 PM | BIRDS OF A FEATHER SESSION | | | | |

Sun, Aug. 29, 2010

| | Salon A-B | Salon C | Salon D | Raleigh | Durham |
|------------------|---|---|---|--|--|
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | REST : Information-Driven Architectures for the 21st Century Brian Sletten | jQuery: Ajax Made Easy Nathaniel Schutta | Agile Velocity Ken Sipe | Virtualization for development Pratik Patel | Real Java EE testing with Arquillian and ShrinkWrap Dan Allen |
| 10:30 - 11:00 AM | MORNING BREAK | | | | |
| 11:00 - 12:30 PM | Real-world JEE performance troubleshooting & tuning: Tips n' Tricks Pratik Patel | JavaScript Beyond the Basics Nathaniel Schutta | RDFa : Weaving Richness and Meaning in the Web Brian Sletten | JRuby in Practice Aaron Bedra | Enter The Gradle Ken Sipe |
| 12:30 - 1:15 PM | LUNCH | | | | |
| 1:15 - 2:15 PM | EXPERT PANEL DISCUSSION | | | | |
| 2:15 - 3:45 PM | Testing the Web Layer Nathaniel Schutta | SPARQL: Querying the Data Web Brian Sletten | Programming Clojure Aaron Bedra | Enterprise Security API library from OWASP Ken Sipe | CDI (JSR-299), Weld and the future of Seam Dan Allen |
| 3:45 - 4:00 PM | BREAK | | | | |
| 4:00 - 5:30 PM | Hacking Your Brain for Fun and Profit Nathaniel Schutta | Semantic SOA : Meaningful Service Strategies Brian Sletten | Easy mobile development (iPhone, Android, Palm Pre, Blackberry) without native code Pratik Patel | Security Boundaries Ken Sipe | The Art of the Spike Aaron Bedra |

Research Triangle Software Symposium

Marriott RTP
August 27 - 29, 2010

Encryption on the JVM: Boot Camp by Matthew McCullough

Does your application transmit customer information? Are there fields of sensitive customer data stored in your DB? Can your application be used on insecure networks? If so, you need a working knowledge of encryption and how to leverage Open Source APIs and libraries to make securing your data as easy as possible. Encryption is quickly becoming a developer's new frontier of responsibility in many data-centric applications.

Encryption on the JVM: Advanced Techniques by Matthew McCullough

Now that you have the basics of encryption under your belt, we'll advance to talking about where it is sensible and performant to add this level of security to your application. Symmetric key and public key encryption have various levels of processing overhead, so you can't blindly just use the "best" encryption out there. What about password hashes? Did you know they are vulnerable with our "salt"? **Prerequisite:** *Encryption Bootcamp on the JVM*

Open Source Debugging Tools for Java by Matthew McCullough

This session will survey a wide range of tools across the Java space. We'll look at utilities such as VisualVM, jstatd, jps, jhat, jmap, Eclipse Memory Analyzer, jtracert, btrace and more. 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. The price of these tools eliminates barriers to their use and their open source nature allows you to mix and match them into compositions that work well for your application's unique debugging needs.

Hadoop: Divide and Conquer Gigantic Datasets (Intro) by Matthew McCullough

Moore's law has finally hit the wall and CPU speeds have actually decreased in the last few years. The industry is reacting with hardware with an ever-growing number of cores and software that can leverage "grids" of distributed, often commodity, computing resources. But how is a traditional Java developer supposed to easily take advantage of this revolution? The answer is the Apache Hadoop family of projects. Hadoop is a suite of Open Source APIs at the forefront of this grid computing revolution and is considered the absolute gold standard for the divide-and-conquer model of distributed problem crunching. The well-travelled Apache Hadoop framework is currently being leveraged in production by prominent names such as Yahoo, IBM, Amazon, Adobe, AOL, Facebook and Hulu just to name a few.

Hadoop: Divide and Conquer Gigantic Datasets (Advanced) by Matthew McCullough

With the basics of Hadoop under your belt, we'll dig into the depths of this amazing framework by writing our own reducer in Java and deploying it to the cluster. Next, we'll dig deeper into DSLs like Pig and its log-file processing cousin, Chukwa. Since grid topology is intentionally very opaque in Hadoop, we'll look at the benefits and how to achieve a properly tuned cluster with replication. Specific to HDFS, we'll tune the configurable parameters for storage redundancy and bucket sizes. **Prerequisite:** *Hadoop: Divide and Conquer Gigantic Datasets (Intro)*

Migrating to Maven 3.0 by Matthew McCullough

Explore what's new on the cutting edge release of Maven, version 3.0. We'll explore the performance improvements, features that make debugging Maven issues easier, and changes to POMs that may require modifications to your build, but will result in more determinate build outputs.

Real Java EE testing with Arquillian and ShrinkWrap by Dan Allen

Unit tests and mocks can only take you so far. Eventually, you'll want to see how your components behave in the real runtime environment, a style of testing referred to as integration testing. This talk introduces Arquillian, an extension for TestNG and JUnit that can be used to validate the behavior of managed and enterprise beans or POJOs that rely on enterprise services just as easily as writing a unit test. Attend this talk to learn how to do real Java EE testing.

CDI (JSR-299), Weld and the future of Seam by Dan Allen

This talk introduces JSR-299: Contexts and Dependency Injection for the Java EE platform (CDI), the new Java standard for dependency injection and contextual lifecycle management. The talk covers the core programming model, explains its relationship to EJB 3.1 and JSF 2.0, and clarifies how it unifies and enhances the Java EE platform as a whole (extending to JPA, JAX-RS and JMS). You are then introduced to Weld, the JSR-299 reference implementation, and its servlet container extension. Finally, we look ahead at how a modularized Seam 3 ties into this new foundation as a set of portable CDI extensions, previewing several examples.

JRuby in Practice by Aaron Bedra

Ruby has made an significant upward trend in the past few years. Alongside this trend Charles Nutter and the fantastic JRuby team have implemented a version of Ruby that runs on the JVM giving you the power of Ruby coupled with the advantages of running on the JVM. Come see for yourself how you can harness the power of rapid development in Ruby and still maintain all the Java interoperability you need to help you build on top of your existing systems.

Programming Clojure by Aaron Bedra

Find out why Clojure is Java.next * Clojure provides clean, fast access to all Java libraries * Clojure provides all the low-ceremony goodness you know and love from dynamic languages * Clojure includes Lisp's signature feature: Treating code as data through macros. * Clojure's emphasis on immutability and support for software transactional memory make it a viable option for taking advantage of massively parallel hardware.

The Art of the Spike by Aaron Bedra

Exploring new technologies can be both challenging and rewarding. A good spike can make or break a new feature for your application. Have you ever thought that a technology or practice your company isn't currently using is the perfect fit for your next iteration? This is your time to shine! In this session you will learn how to treat new technologies as first class citizens and prove that they fit your needs. You will also learn how to provide concrete evidence supporting your decision. By the end of this session your fear of introducing new technology will simply melt away.

Learning Open Source Business Intelligence by Tim Berglund

Traditionally, business intelligence tools have been a high-cost part of any enterprise's software inventory. Recently, options have emerged that allow architects to build a credible business intelligence stack out of entirely open-source components. In this brief overview, we will demonstrate ETL, reporting, and analytics tool that can be deployed free or at low cost. Learn how to turn your company's transactional database into a rich data asset with a business-friendly user interface that integrates into your existing software infrastructure.

Open Source Business Intelligence Workshop by Tim Berglund

Once you're familiar with the concepts of data warehousing, star schemas, cubes, and pivot tables, then it's time to dive in and look at how the tools really work. Continuing from the quick demos in Part I, this workshop session will have you building an actual ETL process with Talend Open Studio. This hands-on exercise will acquaint you with the tooling and solidify the concepts you've learned. **Prerequisite:** *Learning Open Source Business Intelligence (or a solid grasp of BI concepts)*

Complexity Theory and Software Development by Tim Berglund

Some systems are too large to be understood entirely by any one human mind. They are composed of a diverse array of individual components capable of interacting with each other and adapting to a changing environment. As systems, they produce behavior that differs in kind from the behavior of their components. Complexity Theory is an emerging discipline that seeks to describe such phenomena previously encountered in biology, sociology, economics, and other disciplines.

Gaelyk: Lightweight Groovy on the Google App Engine by Tim Berglund

You love Groovy and you're a believer in cloud computing. For a larger project you might choose Grails and hosting on Amazon EC2, but what if you want to take advantage of the nearly massless deployments of a cloud provider like the Google App Engine? You could make Grails work, but it's not always the best fit. Enter Gaelyk.

Decision Making in Software Teams by Tim Berglund

Alistair Cockburn has described software development as a game in which we choose among three moves: invent, decide, and communicate. Most of our time at No Fluff is spent learning how to be better at inventing. Beyond that, we understand the importance of good communication, and take steps to improve in that capacity. Rarely, however, do we acknowledge the role of decision making in the life of software teams, what can cause it to go wrong, and how to improve it.

Grails - How to Build Enterprise Apps by Jeff Brown

Grails represents technology that offers great flexibility and power without the complexity introduced by other Java web application frameworks. Custom tag libraries are a snap. GSP Templates provide a simple mechanism for reusing UI elements. Sitemesh is integrated to help provide a consistent presentation across the entire application. GORM is super powerful ORM. Grails provides simple mechanisms for leveraging the power of Ajax.

GORM Inside And Out by Jeff Brown

GORM is a super powerful ORM tool that makes ORM simple by leveraging the flexibility and expressiveness of a dynamic language like Groovy. With GORM developers get access to all of the power and flexibility of an ORM tool like Hibernate without any of the complexity. **Prerequisite:** *Advanced Grails*

Building Twitter With Grails In 90 Minutes by Jeff Brown

This live coding session will demonstrate how the basics of Twitter can be built using Grails and JMS in less than 90 minutes. The fast paced and code-driven presentation will build a Twitter-like application from scratch using Grails and its rapid application development capability.

Aspect Oriented Programming With Spring AOP by Jeff Brown

Aspect-Oriented Programming (AOP) complements Object-Oriented Programming (OOP) by providing another way of thinking about program structure. The key unit of modularity in OOP is the class, whereas in AOP the unit of modularity is the aspect. Aspects enable the modularization of concerns such as transaction management that cut across multiple types and objects. (Such concerns are often termed crosscutting concerns in AOP literature.)

Compile Time and Runtime Metaprogramming With Groovy by Jeff Brown

The dynamic nature of Groovy makes it a fantastic language for building dynamic applications for the Java Platform. The metaprogramming capabilities offered by the language provide everything that an application development team needs to build systems that are far more capable than their all Java counterparts. Taking advantage of Groovy's metaprogramming capabilities brings great new possibilities that would be very difficult or just plain impossible to write with Java alone. Building Domain Specific Languages in Groovy is easy to do once a team has a good understanding of the Metaobject-Protocol (MOP) and the method dispatch mechanisms used by the Groovy runtime environment.

Emergent Design by Neal Ford

Emergent design is a big topic in the agile architecture and design community. This session covers the theory behind emergent design and shows examples of how you can implement this important concept.

Prerequisite: *understanding of architectural and design concepts*

Evolving towards REST-based Enterprise Integration by Neal Ford

This talk describes an agile approach to architecture, and merges the current state-of-the-art thinking in both service oriented architectures(SOA) and web-based architectures like HTTP, REST, and hypermedia.

Agile Engineering Practices by Neal Ford

Most of the time when people talk about agile software development, they talk about project and planning practices and never mention actual development practices. This talk delves into best development practices for agile projects, covering all of its aspects. **Prerequisite:** *Having worked in an organization that values bureaucracy more than individuals*

Testing the Entire Stack by Neal Ford

This talk covers testing the entire stack: unit, integration, functional, behavior-driven, databases, user acceptance, mocking & stubbing, and other topics and strategies. **Prerequisite:** *Confusion about what to test when and where*

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 Functional Java by Ted Neward

Much noise has been made in recent years about functional languages, like Scala or Haskell, and their benefits relative to object-oriented languages, most notably Java. Unfortunately, as wonderful as many of those benefits are, the fact remains that most Java developers will either not want or not be able to adopt those languages for writing day-to-day code. Which leaves us with a basic question: if I can't use these functional languages to write production code, is there any advantage to learning about them? The short answer is yes, for the fundamental premise--"I can't use functional code on my Java project"--is flawed. Java developers can, in fact, make use of functional ideas, and what's better, they don't even have to reinvent them for Java--thanks to the `FunctionalJava` library, many of the core primitives--interfaces that serve as base types for creating function values, for example--already exist, ready to be used.

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

Once you've learned the core Collections classes, you're done, right? You know everything there is to know about Collections, and you can "check that off" your list of Java packages you have to learn and know, right? **Prerequisite:** *Busy Java Developer's Guide to Collections*

Busy Java Developer's Guide to ParallelizationPatterns by Ted Neward

Getting the most out of your hardware historically was a question of optimizing native code to take advantage of CPU instructions, and setting the right optimization flags on the compiler. Then, when Java came around, it became a matter of simply allowing Hotspot to do its thing. But as the rise of multicore CPUs has made its presence felt in the developer mindspace, so has the need to start writing code in a more parallel/concurrent fashion. Unfortunately, this is an area that most Java developers have historically tried to avoid (like the Plague), so it represents a huge "black hole" for most of them. **Prerequisite:** *Busy Java Developer's Guide to Concurrency (Parts 1 and 2) or equivalent education/experience*

The Busy Java Developer's Guide to Scala: Basics 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.

Pragmatic Architecture by Ted Neward

Building an application is not the straightforward exercise it used to be. Decisions regarding which architectural approaches to take (n-tier, client/server), which user interface approaches to take (Smart/rich client, thin client, Ajax), even how to communicate between processes (Web services, distributed objects, REST)... it's enough to drive the most dedicated designer nuts. This talk discusses the goals of an application architecture and why developers should concern themselves with architecture in the first place. Then, it dives into the meat of the various architectural considerations available; the pros and cons of `JavaWebStart`, `ClickOnce`, `SWT`, `Swing`, `JavaFX`, `GWT`, `Ajax`, `RMI`, `JAX-WS`, `JMS`, `MSMQ`, transactional processing, and more.

Architectural Kata Workshop Part I by Ted Neward

Fred Brooks said, "How do we get great designers? Great designers design, of course." So how do we get great architects? Great architects architect. But architecting a software system is a rare opportunity for the non-architect. The kata is an ancient tradition, born of the martial arts, designed to give the student the opportunity to practice more than basics in a semi-realistic way. The coding kata, created by Dave Thomas,

is an opportunity for the developer to try a language or tool to solve a problem slightly more complex than "Hello world". The architectural kata, like the coding kata, is an opportunity for the student-architect to practice architecting a software system.

Enterprise JPA & Spring 3.0 - Tips and Tricks for JEE Persistence by Pratik Patel

As with many technologies, the basics are easy. The hard part comes when the developer needs to do sophisticated integration, development, and testing as part of an enterprise application. A large enterprise application requires the developer to think of issues that affect the development, scalability and robustness of the application. This presentation will cover the advanced topics described below with a focus on the new persistence features in Spring 3.0 and JPA 2.0.

Virtualization for development by Pratik Patel

We've all heard about virtualization for deploying applications. How about leveraging virtualization for development? In this session, we'll look at some time saving tips and build a virtual VM for development and testing.

Real-world JEE performance troubleshooting & tuning: Tips n' Tricks by Pratik Patel

Performance tuning any application is a black art that can consume much time. Fortunately, Java has many tools that can aid in this effort. There also are a number of basic tips that can help to analyze and fix performance problems. The Java memory model is usually something that you don't need to tune, but for high performance applications it is necessary to tweak. While there are a number of advanced things that can be done to performance tune an application, we'll discover that the simple, basic things are all that are usually needed to make your apps fly.

Easy mobile development (iPhone, Android, Palm Pre, Blackberry) without native code by Pratik Patel

So you have a great idea for an iPhone app, you've tried learning Objective-C, but it's just too hard. What about those other new platforms like Palm Pre and Android? Who wants to write the same app three times? Four times if you count Blackberry! Fear not, there is a much easier way for you to develop on the iPhone. Using a development style called "hybrid mobile applications" you can write apps for iPhone and other platforms using stuff you already know: HTML, CSS and Javascript. In this course, we'll go over the basics for hybrid development

jQuery: Ajax Made Easy by Nathaniel Schutta

Sure, Ajax might not be the hardest thing you'll have to do on your current project, but that doesn't mean we can't use a little help here and there. While there are a plethora of excellent choices in the Ajax library space, jQuery is fast becoming one of the most popular. In this talk, we'll see why. In addition to it's outstanding support for CSS selectors, dirt simple DOM manipulation, event handling and animations, jQuery also supports a rich ecosystem of plugins that provide an abundance of top notch widgets. Using various examples, this talk will help you understand what jQuery can do so you can see if it's right for your next project.

JavaScript Beyond the Basics by Nathaniel Schutta

JavaScript is one of the most widely used languages around and yet its also one of the most misunderstood. With Ajaxified UIs becoming the norm, this humble language is once again at the forefront.

Testing the Web Layer by Nathaniel Schutta

While your project might have nearly 100% code coverage on the server tier, many projects ignore testing the web layer. With more and more code being pushed to the browser, a lack of tests for the client code begs for trouble.

Hacking Your Brain for Fun and Profit by Nathaniel Schutta

The single most important tool in any developers toolbox isn't a fancy IDE or some spiffy new language - it's our brain. Despite ever faster processors with multiple cores and expanding amounts of RAM, we haven't yet created a computer to rival the ultra lightweight one we carry around in our skulls - in this session we'll learn how to make the most of it. We'll talk about why multitasking is a myth, the difference between the left

and the right side of your brain, the importance of flow and why exercise is good for more than just your waist line.

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 todays Java 6 world, with its superior runtime monitoring and management capabilities the reasons to the bounce the server have been greatly reduced.

Agile Velocity by Ken Sipe

The agile development process is all about early and often feedback. One aspect of feedback is how is the team doing... Are we accurate in our estimates? Are we consistent in our velocity? As velocity varies, what is it telling me?

Enter The Gradle by Ken Sipe

In the Java build space, first there was ANT, which provided a reliable way to build without an IDE. Then there was Maven, which provided standardization in build life cycles and dependency management. Now... Enter the Gradle, which provides convention over configuration approach to the build process and an approach at building that isn't based XML. *Prerequisite: Some Groovy helpful*

Enterprise Security API library from OWASP by Ken Sipe

When it comes to cross cutting software concerns, we expect to have or build a common framework or utility to solve this problem. This concept is represented well in the Java world with the loj4j framework, which abstracts the concern of logging, where it logs and the management of logging. The one cross cutting software concern which seems for most applications to be piecemeal is that of security. Security concerns include certification generation, SSL, protection from SQL Injection, protection from XSS, user authorization and authentication. Each of these separate concerns tend to have there own standards and libraries and leaves it as an exercise for the development team to cobble together a solution which includes multiple needs.... until now... Enterprise Security API library from OWASP.

Security Boundaries by Ken Sipe

Security is a large concern in today's world of software development. Security is a multi-dimensional problem requiring skills at a number of different levels. This session is a security overview of a typical Java web development stack.

HTML 5 ... and the Kitchen Sink by Brian Sletten

HTML 5 is an adventurous and confusing prospect that will help change the Web as we know it. It is being finalized as a standard but won't be fully supported by most browsers for quite some time. Companies like Apple and Google have already committed to it as the future of Web application development, however. There are a huge number of new features, updates and gotchas coming at us (including the proverbial kitchen sink!) so it is time to get prepared. This talk will walk you through the new bits and try to put it all into perspective.

REST : Information-Driven Architectures for the 21st Century by Brian Sletten

There is a shift going on in the Enterprise. While still used and useful, the promises of the SOAP/WSDL/UDDI Service-Oriented Architecture (SOA) stack have failed to live up to their promise. A new vision of linked information is enveloping online and Enterprise users. The REST architectural style is squarely behind this thinking as a way of achieving low-cost, flexible integration, increased data security,

greater scalability and long-term migration strategies. If you have dismissed REST as a toy or are unfamiliar with it, you owe it to yourself to see what is so interesting about this way of doing things.

RDFa : Weaving Richness and Meaning in the Web by Brian Sletten

The human web is reasonably well in hand by now. We are getting pretty good at building systems that people find valuable and entertaining. We have not spent as much time concerned about our software friends. There is a ton a rich content available on the web that is too difficult to extract in automated ways using just XHTML, the meta tag and microformats. This talk will introduce you to some emerging technologies from the Semantic Web camp to enrich your web pages with useful information for both automated extraction and improved browsing experiences.

SPARQL: Querying the Data Web by Brian Sletten

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There is an emerging Data Web that relies on technologies from the Semantic Web stack to link increasingly rich connections between various data sources. SPARQL and RDF are the main tools for expressing and using this connectivity. This talk will introduce you to one of the practical and accessible aspects of employing these ideas on the Web and in the Enterprise. **Prerequisite:** *The Semantic Web: The Future, Now and Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa would both be helpful but are not required*

Semantic SOA : Meaningful Service Strategies by Brian Sletten

The goal for web services was always to reduce our burden by increasing the potential for reuse of business functionality. Somehow, we got lost along the way in a morass of confusing, unfulfilling and downright broken technologies. While we are interested in pursuing REST-based systems for managing information, we need some strategies for tying it all together sensibly. If we abandon WSDL, SOAP and UDDI, what do we replace them with? This talk will walk you through combining resource-oriented strategies with technologies from the Semantic Web to describe, find, and bind to services in dynamic, flexible and extensible ways. We will start to blur the distinction between data, documents, services and focus on information and how it is connected to what we already know. **Prerequisite:** *The Semantic Web: The Future Now, Give it a REST and SPARQL : Querying the Data Web would all be helpful talks to have attended*

Third time's a charm: What's new in Spring 3.0 by Craig Walls

In this session, I'll lead a guided tour through the latest that Spring has to offer. Whether you're a Spring veteran or a Spring newbie, there will be something new for nearly everyone.

Building RESTful Applications with Spring MVC by Craig Walls

In this session, we'll look at the latest features of Spring 3.0's MVC framework, with an emphasis on building RESTful web applications. You'll see how Spring 3.0's new `@PathVariable` annotation can be used to develop controllers that respond to resource-oriented RESTful URLs and how `ContentNegotiatingViewResolver` can turn a user-facing web application into a powerful web-based API. We'll explore other ways Spring 3.0 supports working with REST, including how to write REST clients.

Modular Java: An Introduction to OSGi by Craig Walls

Contrary to what you may have heard, OSGi is neither complex, nor heavyweight. In this session, I'll show you how OSGi can actually simplify application development rather than complicate it. We'll look at the benefits of modularity, the fundamentals of OSGi, and see how to develop basic OSGi bundles. We'll also see how a few gadgets in the OSGi toolbox can ease the development of OSGi bundles.

Introducing Spring Roo: From Zero to Working Spring Application in Record Time by Craig Walls

In this example-driven session we'll see how to swiftly develop Spring applications using Spring Roo. We'll start with an empty directory and quickly work our way up to a fully functioning web application. You'll see how Roo handles a lot of heavy-lifting that you'd normally have to do yourself when working with Spring. And we'll stop at a few scenic points along the way to see how Roo accomplishes some of its magic.