

Twin Cities Software Symposium: Spring

Hilton Minneapolis/Bloomington Hotel

March 12 - 14, 2010

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Twin Cities Software Symposium: Spring on March 12 - 14, 2010. TCSS will be held at the Hilton Minneapolis/Bloomington Hotel.

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 TCSS

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 2/22/10 after \$925

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 2/22/10

Registration Fees

Attendees	Before Feb. 22, 2010	After Feb. 22, 2010
5-9	\$725	\$825
10-14	\$700	\$800
15-24	\$675	\$775
25+	\$650	\$750

Go to <http://www.nofluffjuststuff.com/conference/minneapolis/2010/03/home> and register today!

Twin Cities Software Symposium: Spring

Hilton Minneapolis/Bloomington Hotel

March 12 - 14, 2010

Fri, Mar. 12, 2010					
	Salon A	Salon B	Salon C	Jefferson	Washington
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Encryption on the JVM: Boot Camp Matthew McCullough	Slaying the Legacy Dragon: Practical Lessons in Replacing Old Software Tim Berglund	The Busy Java Developer's Guide to Collections Ted Neward	What Stops You From Delivering? : Flow and the Theory of Constraints David Hussman	jQuery: Ajax Made Easy Nathaniel Schutta
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Encryption on the JVM: Advanced Techniques Matthew McCullough	Open Source Business Intelligence Tim Berglund	The Busy Java Developer's Guide to Functional Java Ted Neward	Redesigning Agility: Incorporating Design Thinking David Hussman	Agile UI Nathaniel Schutta
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Hadoop: Divide and Conquer Gigantic Datasets (Intro) Matthew McCullough	Practical Agile Database Development Tim Berglund	The Busy Java Developer's Guide to Advanced Collections Ted Neward	Agility as a Tool: Getting Ready to Iterate David Hussman	JavaScript Beyond the Basics Nathaniel Schutta
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	KEYNOTE				

Sat, Mar. 13, 2010					
	Salon A	Salon B	Salon C	Jefferson	Washington
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Testing the Entire Stack Neal Ford	iBeans: The Simplest Service Integrations You've Ever Implemented Matthew McCullough	Grails - How to Build Enterprise Apps Jeff Brown	What's Brewing in Java Venkat Subramaniam	Decision Making in Software Teams Tim Berglund
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Implementing Evolutionary Architecture Neal Ford	Cloud Computing Boot Camp on the Google App Engine Matthew McCullough	GORM Inside And Out Jeff Brown	Scala Tricks Venkat Subramaniam	Acceptance Test Driven: Beyond the Geekdom David Hussman
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Implementing Emergent Design Neal Ford	Migrating to Maven 3.0 Matthew McCullough	Polyglot Web Programming With Grails Jeff Brown	Tackling Concurrency on the JVM Venkat Subramaniam	Coaching Agility into Organizations: Success over Dogma David Hussman
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Agile Engineering Practices Neal Ford	HTML 5 ... and the Kitchen Sink Brian Sletten	Aspect Oriented Programing With Spring AOP Jeff Brown	How to Approach Refactoring Venkat Subramaniam	MongoDB Ted Neward
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSION				

Sun, Mar. 14, 2010					
	Salon A	Salon B	Salon C	Jefferson	Washington
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Architecting Code for Concurrent Execution: Theory and Practice Robert Fischer	Third time's a charm: What's new in Spring 3.0 Craig Walls	Software Architecture for the Cloud Michael Nygard	XSS-Proof Ken Sipe	REST : Information-Driven Architectures for the 21st Century Brian Sletten
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	The Concurrency Toolset: JConch, Google Collections, and java.util.concurrent Robert Fischer	Introducing Spring Roo: From Zero to Working Spring Application in Record Time Craig Walls	Design for Operations Michael Nygard	Enter The Gradle Ken Sipe	RDFa : Weaving Richness and Meaning in the Web Brian Sletten
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Modular Java: An Introduction to OSGi Craig Walls	Agile Velocity Ken Sipe	DSLs in Scala: Internal and External Michael Nygard	Integrating Groovy Concurrency with Java Robert Fischer	SPARQL : Querying the Web of Data Brian Sletten
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Beyond JUnit: Powertools for Test-Driven Development Craig Walls	So you want to be an Architect Ken Sipe	High Performance Persistence with Redis Michael Nygard	Agile Practices Review: A Tactics Retrospective Robert Fischer	Semantic SOA : Meaningful Service Strategies Brian Sletten

Twin Cities Software Symposium: Spring

Hilton Minneapolis/Bloomington Hotel

March 12 - 14, 2010

Slaying the Legacy Dragon: Practical Lessons in Replacing Old Software by Tim Berglund

It's a given that everyone hates the legacy application and wants to replace it. You're tired of the brittle, untested code, the outdated frameworks, the platform nobody cares about anymore. You want to apply current practices and the productivity gains of today's tools. Usually this is just a frustrated dream, but every once in a while, you actually get to do it. That's great news, but it raises a question: how do you...do that?

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.

Practical Agile Database Development by Tim Berglund

Do your team's agile practices extend to the database? Agile methods are fairly well-understood as they apply to code, but these principles are not commonly understood or practiced on the databases that typically accompany enterprise software projects. Learn the tools, techniques, and mindset your team needs to make incremental improvements to the database's design over time with confidence.

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*

Polyglot Web Programming With Grails by Jeff Brown

Grails is one of the most flexible and most powerful frameworks on The Java Platform. Grails leverages the flexibility offered by the platform in a way that other web frameworks do not. Grails is a fantastic platform for polyglot web programming. **Prerequisite:** *Advanced Grails*

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.)

Architecting Code for Concurrent Execution: Theory and Practice by Robert Fischer

The power of multicore machines and cloud computing is all dependent on an application's ability to successfully leverage concurrency. Although concurrency has traditionally been considered fatally difficult in

Java, a few simple architecture principles can make all the difference. This session will review some of those principles in both theory and practice.

The Concurrency Toolset: JConch, Google Collections, and java.util.concurrent by Robert Fischer

JConch is a library that provides a few high-level tools for high-concurrency environments on the JVM. The java.util.concurrent package in the Java standard library provides low-level structures for managing concurrent communication. Learn here how to use both of them in order to produce clean, highly-concurrent, and highly-tunable code.

Integrating Groovy Concurrency with Java by Robert Fischer

The Groovy language now provides substantial concurrency capabilities via the GPars library, including the ability to work with actors and dataflow concurrency. This talk shows how you can integrate these Groovy concurrency structures into your Java applications.

Agile Practices Review: A Tactics Retrospective by Robert Fischer

Increasingly, people are adopting Agile practices a la carte, and some are even talking about "post-Agile" methodologies. If things are going to be changing, let's take a moment to review Agile development practices, the problems they were trying to solve, what worked, and what difficulties these new methodologies are responding to. With this information in hand, we can make an intelligent decision about the development methodology for our team.

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*

Implementing Evolutionary Architecture 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.

Implementing Emergent Design by Neal Ford

Emergent design is a big topic in the agile architecture and design community. This session covers some of the theory behind emergent design, but spends most of its time showing examples of how you can implement this important concept. **Prerequisite:** *understanding of architectural and design concepts*

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*

What Stops You From Delivering? : Flow and the Theory of Constraints by David Hussman

What stops you from delivering to your customers and what truly prevents incremental learning in your project community? How much of it lies in coding issues? Do you even need to write code to learn where to go or how to get there? For many companies, being agile means producing working code from an iteration. While working code is a measure of progress, and provides a tool to validate success, it is not always your best investment. Your best investment lies in combining 1) what helps you learn about your product and your market and 2) what you need to do to deliver it, and what is constraining either of these.

Redesigning Agility: Incorporating Design Thinking by David Hussman

Design tends to mean one thing to developers and another thing to designers. The later group are product designers and are not aware of the structure of the code. The hipsters in the agile community are trying to blend product design into the process of coding and delivering software. This session discusses what to do after your agility is flowing, or post agilism: imagine you using agile methods successfully, what's next?

Agility as a Tool: Getting Ready to Iterate by David Hussman

Many people simplistically apply agile recipes, assuming a one size fits all approach. This may lead to naive use beliefs like collocation breeds instant success. While sitting together always helps, it does not mean that

people spontaneously collaborate to create sustainable value. Instead of approaching agile methods like a recipe, this session will teach you to design agility that is a useful tool for your project community. We will cover practice selection ideas, tools for creating healthy development eco-systems and product discover tools. If you would like to improve the stickiness of your agility, stop in learn a pile of techniques to use before holding your first planning session.

Acceptance Test Driven: Beyond the Geekdom by David Hussman

How testable are your requirements? If they are user stories, you have better odds than others, but you may still be struggling to increase your automated feature testing. Acceptance test driven (ATD) means understanding the needs or the value before you start coding, and it is often hard to make ATD sticky in organizations. While tools and geekology are needed, the upstream flow with the larger project community must exist to be successful. The session will teach you how to spark ATD from idea to code, connecting the dots and the players along the way. From product concept to automated tests, we will discuss and practice various ways to connect business, development and testing around the creating and getting value from ATD.

Coaching Agility into Organizations: Success over Dogma by David Hussman

If you want to introduce meaningful agility in your company, real change takes more than a few certification courses or one successful pilot project. Process that produces tangible results sticks while dogmatic preaching is short lived. If you are planning to grow agility that helps you produce better products and better code in meaningful timeframes, stop in and learn how to succeed and how to avoid common failures.

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*

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.

iBeans: The Simplest Service Integrations You've Ever Implemented by Matthew McCullough

No app is an island nowadays and your bleeding edge Java & JavaScript apps demand that you integrate with Facebook, Amazon, Gmail, Google Search, Twitter or S3 just to name a few. Make your next integration project a breeze by leveraging the successful work of others from the iBeans Central repository, or if necessary, simply author a new iBean and contribute it back for the benefit of all. iBeans a new ultra-light service integration framework written in Java, but targeting both Java and JavaScript. It provides a centralized mechanism for community contributions of beans to the most commonly used services such as Twitter, Flickr, Gmail and more.

Cloud Computing Boot Camp on the Google App Engine by Matthew McCullough

Cloud this, cloud that. It's all we are hearing about these days. And whether buzz-worthy or not, you need to get in-the-know so that you can talk effectively about how this could fit into the application strategy on your next project. Leverage 100s of hours of research distilled into a 90 minute presentation. Get bootstrapped

with what cloud computing is and isn't, who the players are in this space, what unique features each offers, and then how Google is completely changing the game.

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.

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*

MongoDB by Ted Neward

MongoDB is designed for problems without heavy transactional requirements that aren't easily solved by traditional RDBMSs, including problems which require the database to span many servers. Like other document-oriented database systems such as CouchDB, MongoDB is not a relational database management system. The database manages collections of JSON-like documents which are stored in a binary format referred to as BSON.

Software Architecture for the Cloud 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. In this session, Michael will discuss the ingredients of real cloud computing and how you can apply it to your applications. He will show several architectures and discuss applications that fit each of these models. Finally, he will also talk about some of the pitfalls and problems that cloud computing customers can encounter.

Design for Operations by Michael Nygard

If your software fails in production, nobody will care how great the development project was, or how well the system passed QA. Production operations, the domain of your systems' least-appreciated stakeholders, is where the rubber meets the road. Come learn how to build your systems to thrive in Operations.

DSLs in Scala: Internal and External by Michael Nygard

We're no longer working in a single language. Programming today is about both consuming and creating languages. We've all heard a lot about domain specific languages (DSLs). So much so, in fact, that it seems like more people are talking about it than doing it. In this session, Michael will present a real domain with familiar problems. He'll then demonstrate both an internal DSL and an external DSL that solve the same problem. Along the way, we'll cover fluent interfaces, composing multiple DSLs, and the very cool parser

combinators. If you've ever left a DSL talk wondering when we're ever going to get past the "coffee ordering DSL" or the "Waffle House breakfast DSL", then you will want to see this session. **Prerequisite:** *None.*

High Performance Persistence with Redis by Michael Nygard

Redis is one of the fresh crop of "NoSQL" storage solutions. It's a distributed key-value store that knows how to deal with data structures. Oh, and it happens to also be incredibly fast. Like, microseconds per write.

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.

Agile UI by Nathaniel Schutta

Some developers assume that agility and usability are mutually exclusive - in reality, they are extremely complimentary; if you squint, you might have a hard time telling the difference between agile practices and good user interface design. This usability talk is aimed squarely at developers giving you the tools you need to develop UIs that won't make your users yack. We'll discuss the importance of observation, personas, paper prototyping, usability testing and the importance of good moderators. In addition, we'll map the various aspects of user interface design to a typical agile iteration.

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.

XSS-Proof by Ken Sipe

Companies have focused for years to solidify the back-end infrastructure in defense against hacking attempts. Most companies however are forced to open up many ports including port 80 (http) for users to access web applications among other resources. This has lead to web attacks growing to be the #1 classification of hacker attacks today. In this space Cross Site Scripting (XSS) is the #1 ranked vulnerability affecting a large number of sites. This evolution requires that the understanding of securing an application move beyond sys admins and incorporate all aspects of system delivery for the protection of a system and system resources.

Enter The Gradle by Ken Sipe

First in the Java Build space there was ANT, and there was 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*

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?

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. Whither 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.

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 Web of Data by Brian Sletten

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There are emerging Data Webs (both public and private) that rely 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 these topics and the practical and accessible aspects of employing them on the Web and in the Enterprise. Getting people to come to consensus on common models and schemas is usually the hardest part of any data integration strategies. These technologies help lower the bar on both the technical and social costs of stepping up your integration strategies.

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*

What's Brewing in Java by Venkat Subramaniam

Java has come a long way, and yet there is so much that's happening in this space. In this presentation we will take a look at the exciting additions and changes coming up in the next version of Java. **Prerequisite:** *Good programming knowledge of Java*

Scala Tricks by Venkat Subramaniam

Scala is a very powerful hybrid functional pure object oriented language on the JVM. Scala is known for its conciseness and expressiveness. In this presentation we will look at some common tasks you do everyday in developing applications and see how they manifest in Scala.

Tackling Concurrency on the JVM by Venkat Subramaniam

In this presentation we will take a quick walk though the issues with concurrency and how the solutions provided in Scala and Clojure help address those.

How to Approach Refactoring by Venkat Subramaniam

You can't be agile if your code sucks. You know that you have to constantly refactor your code and design. But the questions is how? In this presentation, instead of looking at a laundry list of refactoring techniques,

we will instead look at how to effectively approach refactoring and along the way discuss some core principles to look for.

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.

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.

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.

Beyond JUnit: Powertools for Test-Driven Development by Craig Walls

Writing tests is more than just writing JUnit test cases and hoping that they'll pass when your project is built. If you want assurance that your code is sound and provides the desired functionality, then you'll want to test it from every angle and run those tests as frequently as possible.