

Greater Atlanta Software Symposium

Atlanta Marriott Northwest
September 24 - 26, 2010

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Greater Atlanta Software Symposium on September 24 - 26, 2010. GASS2010 will be held at the Atlanta Marriott Northwest.

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 GASS2010
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 8/30/10 after \$925

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 8/30/10

Registration Fees

Attendees	Before Aug. 30, 2010	After Aug. 30, 2010
5-9	\$725	\$825
10-14	\$700	\$800
15-24	\$675	\$775
25+	\$650	\$750

Go to <http://www.nofluffjuststuff.com/conference/atlanta/2010/09/home> and register today!

Greater Atlanta Software Symposium

Atlanta Marriott Northwest

September 24 - 26, 2010

Fri, Sep. 24, 2010

	Cobb Ballroom	Salon A	Salon E	Marietta	Albany
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	HTML 5 ... and the Kitchen Sink Brian Sletten	Encryption on the JVM: Boot Camp Matthew McCullough	Transforming to Groovy Venkat Subramaniam	Agile Engineering Practices Neal Ford	Learning Open Source Business Intelligence Tim Berglund
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	REST : Information-Driven Architectures for the 21st Century Brian Sletten	Encryption on the JVM: Advanced Techniques Matthew McCullough	Unit Testing and Mocking your Java Code with Groovy Venkat Subramaniam	Visualizations for Code Metrics Neal Ford	Open Source Business Intelligence Workshop Tim Berglund
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Evolving towards REST-based Enterprise Integration Neal Ford	Cloud Computing Boot Camp on the Google App Engine Matthew McCullough	Thinking Functional Style Venkat Subramaniam	Decision Making in Software Teams Tim Berglund	The ABCs of RDF Vocabularies Brian Sletten
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Venkat Subramaniam				

Sat, Sep. 25, 2010

	Cobb Ballroom	Salon A	Salon E	Marietta	Albany
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Testing the Entire Stack Neal Ford	Hadoop: Divide and Conquer Gigantic Datasets (Intro) Matthew McCullough	Scala Tricks Venkat Subramaniam	Third time's a charm: What's new in Spring 3.0 Craig Walls	RDFa : Weaving Richness and Meaning in the Web Brian Sletten
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Emergent Design Neal Ford	Hadoop: Divide and Conquer Gigantic Datasets (Advanced) Matthew McCullough	Test Driving Multithreaded Code Venkat Subramaniam	Spring for XML-Haters Craig Walls	Complexity Theory and Software Development Tim Berglund
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Open Source Debugging Tools for Java Matthew McCullough	Introducing Spring Roo: From Zero to Working Spring Application in Record Time Craig Walls	Tackling Concurrency on the JVM Venkat Subramaniam	Gaelyk: Lightweight Groovy on the Google App Engine Tim Berglund	Virtualization for development Pratik Patel
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Git Going with Distributed Version Control Matthew McCullough	Modular Java: An Introduction to OSGi Craig Walls	How to Approach Refactoring Venkat Subramaniam	Practical Agile Database Development Tim Berglund	Aspect Oriented Programming With Spring AOP Jeff Brown
4:45 - 5:45 PM	BOF & Refactoring Workshop with Venkat Subramaniam				

Sun, Sep. 26, 2010

	Cobb Ballroom	Salon A	Salon E	Marietta	Albany
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	jQuery: Ajax Made Easy Nathaniel Schutta	What's Brewing in Java Venkat Subramaniam	The Agile Guerilla Matt Stine	Grails - How to Build Enterprise Apps Jeff Brown	Engineering your DSLs Peter Bell
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Testing the Web Layer Nathaniel Schutta	Enterprise JPA & Spring 3.0 - Tips and Tricks for JEE Persistence Pratik Patel	The Seven Wastes of Software Development Matt Stine	GORM Inside And Out Jeff Brown	Tooling for External DSLs Peter Bell
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Real-world JEE performance troubleshooting & tuning: Tips n' Tricks Pratik Patel	JavaScript Beyond the Basics Nathaniel Schutta	Agile Development with OSGi Matt Stine	Building Twitter With Grails In 90 Minutes Jeff Brown	DSL Evolution Peter Bell
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Easy mobile development (iPhone, Android, Palm Pre, Blackberry) without native code Pratik Patel	Agile UI Nathaniel Schutta	Yes You Kanban Matt Stine	Compile Time and Runtime Metaprogramming With Groovy Jeff Brown	Options for DSLs on the JVM Peter Bell

Greater Atlanta Software Symposium

Atlanta Marriott Northwest
September 24 - 26, 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*

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.

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)*

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.

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*

Engineering your DSLs by Peter Bell

The easy part of implementing Domain Specific Languages is coding them. The hard part comes when you have to think about testing, documenting, evolving and providing appropriate editing interfaces for them.

Tooling for External DSLs by Peter Bell

Eventually if you do a lot of work with DSLs, you'll need to consider using external DSLs to make your languages more flexible and powerful.

DSL Evolution by Peter Bell

Oh no, your DSL is actually popular. And of course, the feature requests keep come in. But what do you do to evolve your DSLs without breaking the existing models as your understanding of the domain changes radically over time?

Options for DSLs on the JVM by Peter Bell

What are the various options for building DSLs on the JVM and how can you choose the best one for your next project?

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)*

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.

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.

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.

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.)

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.

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.

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*

Visualizations for Code Metrics by Neal Ford

Judicious use of metrics improves the quality of your code. But interpreting metrics presents a challenge. You have a list of numbers for a project - what does it mean? And what does it tell me about the health of the project overall? This sessions shows how to produce visualizations for software metrics, making them easier to understand and more valuable. It covers metrics at the individual method level all the way up to the overall architecture of the application. This isn't just a talk about how some tools produce visualizations: this session shows you how to generate your own visualizations, allowing you to customize it to the level in information density that shows real value on your project. I show how to produce projected graphs from dependencies, heat-maps for cyclomatic complexity and code coverage, using XSLT to extract visual information from XML configuration documents, and others. Metrics can't help you if you can't understand them. By creating visualizations, it helps leverage metrics to make your code better.

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.

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*

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*

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.

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.

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.

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.

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.

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.

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.

The ABCs of RDF Vocabularies by Brian Sletten

RDF is one of the lower level technologies of the Semantic Web. We use it to describe and connect information on both our public and private Webs. Everyone enjoys using Dublin Core, FOAF, Creative Commons and other existing RDF vocabularies, but how do you go about writing your own? And why would you want to?

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.

The Agile Guerilla by Matt Stine

So you discovered agile software development this weekend. You've finally found the tools that you're going to use to fix your team. Do you rush in to work Monday morning with a slide deck in one hand and a baseball bat in the other, ready to bludgeon the first person who checks in untested code? How do you think that's going to work out for you? I can tell you from personal experience that it doesn't play out too well. There is a better way.

The Seven Wastes of Software Development by Matt Stine

One of the first principles of lean software development is the elimination of waste. Shigeo Shingo identified seven types of manufacturing waste in his "A Study of the Toyota Production System." Later, the Poppendieck's translated these to seven wastes of software development.

Agile Development with OSGi by Matt Stine

There isn't much said in typical Agile conversation about architecture and modularity. We will attempt to redress this omission by examining an agile approach to logical system architecture coupled with a potential implementation for the Java platform. Tracer Bullet Development (TBD) is a technique that allows you to prove out the proposed architecture of your system by firing a "tracer bullet" through a vertical slice of your system that exercises all of its horizontal components. It has multiple benefits, including encapsulation, decoupled code, parallel code development, and more. OSGi is a specification for a dynamic module system for Java with multiple open source implementations. It allows you to modularize your system into "bundles" which essentially firewall their own classloader space. Objects running within a bundle can only see types that they explicitly import and only expose types that they explicitly export. They interact with other bundles

by expose and consuming services which are registered under a public interface. At face value it seems that Tracer Bullet Development and OSGi are a match made in heaven!

Yes You Kanban by Matt Stine

Kanban is sweeping through the agile software development space. Is it hype? Or is it a useful tool to add to our belt?

Transforming to Groovy by Venkat Subramaniam

Groovy is a elegant, dynamic, agile, OO language. I like to program in Groovy because it is fun and the code is concise and highly expressive. Writing code in a language is hardly about using its syntax, however. It is about using the right idioms. Come to this section to pick up some nice Groovy idioms. **Prerequisite:** *Some knowledge of Groovy is helpful but not required.*

Unit Testing and Mocking your Java Code with Groovy by Venkat Subramaniam

One of the best ways to introduce Groovy to new projects and teams is to use it for Unit testing Java code. Using Groovy to unit test Java code has several advantages. You can take advantage of its concise syntax for writing tests. Groovy's dynamic and metaprogramming capabilities can be exploited for mocking purposes.

Thinking Functional Style by Venkat Subramaniam

Functional Programming has been around for a while, but it is gaining popularity, especially due to direct support in languages on the JVM. Writing code in functional style is not about syntax. It is a paradigm shift. In this presentation, using examples from Java, Groovy, and Scala, you will learn how to write code in functional style.

Keynote: It could be heaven or hell (being a polyglot programmer) by Venkat Subramaniam

Keynote discussing the benefits and challenges of being a polyglot programmer.

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.

Test Driving Multithreaded Code by Venkat Subramaniam

Once I got convinced about the benefits of TDD, I used it pretty extensively and consistently to drive the design and development of my code. So, it came as a surprise when I was trying to convince myself that those practices do not apply on a highly multithreaded code I was creating on a project. Thankfully, I set out to prove that TDD does not apply, but ended up proving myself wrong.

Tackling Concurrency on the JVM by Venkat Subramaniam

In this presentation we will take a quick walk through 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 question 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.

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*

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.

Spring for XML-Haters by Craig Walls

In this presentation, we'll explore all of the ways to do bean wiring in Spring. We'll take a pragmatic view of each style, evaluating their strengths, weaknesses, and applicability to varying circumstances.

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.