

Greater Florida Software Symposium

Sheraton Suites Westshore

April 16 - 18, 2010

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Greater Florida Software Symposium on April 16 - 18, 2010. GFSS2010 will be held at the Sheraton Suites Westshore.

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 GFSS2010

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: \$850/person good thru 3/22/10 after \$950

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 3/22/10

Registration Fees

Attendees	Before Mar. 22, 2010	After Mar. 22, 2010
5-9	\$750	\$850
10-14	\$725	\$825
15-24	\$700	\$800
25+	\$675	\$775

Go to <http://www.nofluffjuststuff.com/conference/tampa/2010/04/home> and register today!

Greater Florida Software Symposium

Sheraton Suites Westshore

April 16 - 18, 2010

Fri, Apr. 16, 2010

	Manatee/Orange	Pasco	Sarasota	Hillsborough
12:00 - 1:00 PM	REGISTRATION			
1:00 - 1:15 PM	WELCOME			
1:15 - 2:45 PM	Implementing Evolutionary Architecture Neal Ford	The Busy Java Developer's Guide to Collections Ted Neward	Intro to Messaging Using JMS and ActiveMQ Mark Richards	The Agile Mindset: Applying Agile in Non-Technical Areas of an Organization Tiffany Lentz
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Emergent Design Neal Ford	The Busy Java Developer's Guide to Functional Java Ted Neward	The Art of Messaging Mark Richards	Iteration Management: What's in your Toolkit? Tiffany Lentz
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Testing the Entire Stack Neal Ford	The Busy Java Developer's Guide to Advanced Collections Ted Neward	Common AntiPatterns and How To Avoid Them Mark Richards	Agile Project and Management Metrics: Measuring Success Downward and Upward Tiffany Lentz
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	KEYNOTE: Neal Ford - Smithying in the 21st Century			

Sat, Apr. 17, 2010

	Manatee/Orange	Pasco	Sarasota	Hillsborough
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Using Apache Camel Mark Richards	HTML 5 ... and the Kitchen Sink Brian Sletten	Test Driven Design Neal Ford	The Busy Java Developer's Guide to Scala: Basics Ted Neward
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Architecture and Scaling Ken Sipe	REST : Information-Driven Architectures for the 21st Century Brian Sletten	Grails - How to Build Enterprise Apps Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	Spring 3 into REST Ken Sipe	RDFA : Weaving Richness and Meaning in the Web Brian Sletten	GORM Inside And Out Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Debugging your Production JVM Ken Sipe	Semantic Web : The Future Now Brian Sletten	Polyglot Web Programming With Grails Jeff Brown	Enterprise JPA & Spring 3.0 - Tips and Tricks for JEE Persistence Pratik Patel
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSIONS			

Sun, Apr. 18, 2010

	Manatee/Orange	Pasco	Sarasota	Hillsborough
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Enterprise Security API library from OWASP Ken Sipe	jQuery: Ajax Made Easy Nathaniel Schutta	Aspect Oriented Programing With Spring AOP Jeff Brown	Automated Software Quality Control Tools Pratik Patel
10:30 - 11:00 AM	MORNING BREAK			
11:00 - 12:30 PM	Agile Velocity Ken Sipe	JavaScript Beyond the Basics Nathaniel Schutta	Compile Time and Runtime Metaprogramming With Groovy Jeff Brown	Real-world JEE performance troubleshooting & tuning: Tips n' Tricks Pratik Patel
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	Enter The Gradle Ken Sipe	Agile UI Nathaniel Schutta	Performance and Scalability Revisited: In-Memory Data Grids Aleksandar Seovic	Virtualization for development Pratik Patel
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	So you want to be an Architect Ken Sipe	Testing the Web Layer Nathaniel Schutta	In-Memory Data Grids: Not Your Mom's Cache Aleksandar Seovic	Easy mobile development (iPhone, Android, Palm Pre, Blackberry) without native code Pratik Patel

Greater Florida Software Symposium

Sheraton Suites Westshore

April 16 - 18, 2010

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.)

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.

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.

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*

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*

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

The Agile Mindset: Applying Agile in Non-Technical Areas of an Organization by Tiffany Lentz

Agile techniques are often pigeon-holed as just applying to software projects and IT organizations. Agile techniques are a mindset more than a list of rules to follow and can bring efficiency and improvements to all areas of an organization.

Iteration Management: What's in your Toolkit? by Tiffany Lentz

Is your Agile team running smoothly? How do you know? This answer is found in your iteration and your toolkit for constant team improvement! Comparing Iteration Management skills and tools to those of the Agile Project Manager, Scrum Master and Technical Leader roles, you will see that Iteration Management encompasses end to end activities of the iteration, which are crucial to unblocking your software production line and making your team a success.

Agile Project and Management Metrics: Measuring Success Downward and Upward by Tiffany Lentz

This session focuses on visibility of team progress and correcting team "bad smells" using Agile metrics. Since we use metrics to self-correct and sharpen the team, they are an integral part of each iteration and flow into release planning. We will discuss some of the How's and Why's of team metrics and review the risks that are inevitable when it comes to gathering metrics.

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*

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.

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

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

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

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using `Threads` still remain a dark and mysterious art, and synchronization beyond the use

of the "synchronized" keyword is almost unknown. **Prerequisite:** *The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

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.

Automated Software Quality Control Tools by Pratik Patel

This session is aimed at helping developers get started with automating the collection of software quality metrics. We'll cover continuous integration, automated code metrics gathering, and analysis of these metrics. The ability to be able to detect problems early, and also to write higher quality code early, helps avoid bugs and headache down the line. We'll cover some best practices around using and putting in tools to help achieve higher quality.

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.

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.

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

Intro to Messaging Using JMS and ActiveMQ by Mark Richards

More and more companies are using messaging as a means for heterogeneous communication, scalability, performance, and load balancing. Why? Because messaging provides asynchronous requests, guaranteed delivery, load balancing, and ease of development. In this session I will introduce some basic messaging fundamentals, then show how easy it is to send and receive messages using the JMS API. During this session I will also show how to setup and configure ActiveMQ, an open source enterprise-wide messaging provider. By attending this session you will see how easy messaging using JMS really is!

The Art of Messaging by Mark Richards

Messaging is both a science and an art. Messaging is a science with respect to the mechanics of the JMS API and the syntax for sending and receiving messages. However, messaging is also an art when it comes to applying the JMS API to solve real-world problems. In this session I will review some of the more common use cases for messaging and show techniques for significantly increasing both the performance and scalability of messaging-based applications. Using ActiveMQ, you will see how to create embedded brokers, solve internal application bottleneck issues, how to use asynchronous logging with Log4J and JMS, and how to significantly speed up your messaging applications. In this session I will also describe and demonstrate some emerging trends in RESTful JMS (that is, JMS over HTTP). Come to this session to find

out how much fun messaging can really be! **Prerequisite:** *Some knowledge of JMS and Messaging in general*

Common AntiPatterns and How To Avoid Them by Mark Richards

In the book "97 Things Every Software Architect Should Know" (O'Reilly, 2009) I wrote about the importance of design patterns as a useful means of communication between architects and developers. Equally important to patterns is an understanding of AntiPatterns - things that we repeatably do that produce negative results. AntiPatterns are used by developers, architects, and managers every day and are one of the main factors that prevent progress and success. In this session we will look at some of the more common and significant development and architecture antipatterns. Through coding and design examples, you will see how these antipatterns emerge, how to recognize when the antipattern is being used, and most importantly, how to avoid them. By attending this session, you will be part of a movement to reduce the AntiPattern catalog from hundreds of entries to only a few. **Prerequisite:** *None*

Using Apache Camel by Mark Richards

Apache Camel is a robust open source integration framework that handles routing and mediation tasks associated with enterprise integration. Camel allows you to quickly and easily route messages and integrate components in a distributed, decoupled manner. For example, using the Camel Java DSL, you can send and receive JMS messages in just a couple of lines of Java code. In this live coding session I will describe what Camel is, describe the overall architecture, show why it is useful, and demonstrate through live coding examples how to use the Camel Java DSL to write simple (and complex) routing logic. By attending this session you will learn Camel well enough to use it at work the next day.

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.

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.

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.

Performance and Scalability Revisited: In-Memory Data Grids by Aleksandar Seovic

Building scalable, highly-available applications that perform well is not an easy task. These features cannot be simply "bolted" onto an existing application – they have to be architected into it. Unfortunately, the things we need to do to achieve them are often in conflict with each other, and finding the right balance is crucial.

In-Memory Data Grids: Not Your Mom's Cache by Aleksandar Seovic

While many developers still think of in-memory data grids as clustered caches, in reality they are much more and provide a solid foundation for the next generation of scalable web and enterprise applications.

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications that are scalable. How do you know if the application scales?

Spring 3 into REST by Ken Sipe

REST as an architectural approach is greatly simplified through the selection of framework or tool to help with the tedious and repetitive template style that it demands. Until recently, some of the best approaches were through frameworks that required the use of languages other than Java, such as Rails or Grails. In the Java space, the choices were limited. The newly released Spring 3 changes that. One of the most significant changes in Spring 3 is it's support for REST, which includes client as well as server support. **Prerequisite:** *Java 5*

Debugging your Production JVM by Ken Sipe

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

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 their 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.

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*

So you want to be an Architect by Ken Sipe

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

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.

Semantic Web : The Future Now by Brian Sletten

Just as the world is feeling comfortable with the Web, Tim Berners-Lee et al inform us that what we have seen so far is just the beginning. His original plans at CERN were larger and grander. The Semantic Web is a vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. The only problem is that there is no such thing. There is no Semantic Web, just the Web we have that is increasingly semantics-enabled. Forget the hype. Come learn how the technologies of this vision are being used today on the Web and in the Enterprise by more people than you might think.