

Greater Atlanta Software Symposium 2005

WHotel Atlanta Perimeter - Atlanta, GA

October 21 - 23, 2005

(session listing as of 10/18/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Greater Atlanta Software Symposium coming to Atlanta on October 21 - 23, 2005. You will have the opportunity to attend the best technically focused Java / Open Source event anywhere. We make this claim based on the following:

- 1) **Excellent Speakers with unparalleled access**
- 2) **Limited Attendance = 250 Registrants Max**
- 3) **No Vendors, No Sales Pitches, No Marketecture**
- 4) **Unmatched Value - less than 1/3 of the cost of a national conference**
- 5) **Since 2002, we have delivered over fifty(50) conferences throughout North America**

The No Fluff Just Stuff Software Symposium Series caters to individual developers, development teams, project managers, architects and independent consultants. The Greater Atlanta Software Symposium will offer 6 concurrent sessions over three days with over 65 sessions to choose from. The following topics will be featured:

- 1) Client Side Java
- 2) Core Java
- 3) .Net
- 4) ServerSide Java
- 5) Architecture
- 6) XML / Web Services

Registration Fees

Attendees	Before 10/3/2005	After 10/3/2005
1-4	\$675	\$775
5-9	\$600	\$675
10-14	\$575	\$650
15-24	\$550	\$625
25+	\$525	\$600

The Registration Fee includes the following:

- 1) All Access Pass to the three day symposium
- 2) Handouts from all sessions attended w/binder
- 3) CD with all presentational content @ registration
- 4) Custom NFJS Laptop/Backpack
- 5) Opportunity to win an iPod everyday during the symposium raffle

Go to <http://www.nofluffjuststuff.com> for more details.

Questions/Comments: Contact Jay Zimmerman: zimmerman@nofluffjuststuff.com or (303)469-0486.

Greater Atlanta Software Symposium 2005

WHotel Atlanta Perimeter - Atlanta, GA

October 21 - 23, 2005

(session listing as of 10/18/2005)

Using Code Metrics for Targeted Code Refactoring by Andrew Glover

The knowledge of how to effectively spot smelly code and replace it with proven patterns will ultimately lead to a more stable, maintainable and elegant code base.

Taking Quality to the Next Level through Code Coverage Analytics by Andrew Glover

Understanding what code coverage represents, how to effectively apply it, and how to avoid its pitfalls will give you an unprecedented understanding of how unit tests may or may not be covering you from sneaky defects.

Introduction to TestNG, the next generation testing framework for developers by Andrew Glover

No one will argue the positive affects JUnit has had on the quality of thousands of Java applications around the world. JUnit's simplicity and ease of use ushered in a whole new era of code quality; however, as many developers have found, its simplicity has also limited its use. TestNG was designed from the ground up to overcome some of JUnit limitations.

Introduction to Ajax by Ben Galbraith

Ajax -- called DHTML just a few months ago -- has revolutionized (or "radically iterated", if you like) web application development in the short few months since the term was coined. What is it all about? Why are we excited about a set of capabilities that have been sitting in our browser for years? What can you do with it? And, how can you do it?

Creating Killer Graphics and Professional PDFs with XML by Ben Galbraith

You can do some pretty cool things with XML these days (despite what some curmudgeons in the technology world may claim). In the past few years, XML has solidified its place as the lingua franca of data sharing and data manipulation. But XML as a data transfer language is only marginally interesting. Things get really exciting when XML is dynamically transformed into other formats. In this session, I focus on two XML formats which can be readily transformed into high-quality presentation-centric output formats. XSL-FO is a typesetting format for XML that can be readily converted into PDF (or Postscript and some other formats). SVG is a vector graphics language in XML -- a sort of open-source version of the popular Macromedia Flash format. SVG files can be converted into beautiful, completely scalable -- and interactive -- images.

Ajaxian JavaScript Frameworks by Ben Galbraith

In the "Introduction to Ajax" session, we discuss what Ajax is, how it works, and how others are using it. This session goes deeper into Ajax by reviewing the existing JavaScript frameworks that aim to make it easier.

Creating Polished Swing Applications by Ben Galbraith

Too often, Swing applications are slow, ugly, and hard-to-maintain. It turns out that it doesn't have to be this way. Swing can be used to create highly-responsive, beautiful applications that are very maintainable. If this isn't consistent with your own experience, don't feel bad; it's not very obvious how to make Swing sing.

Making the Most of XML by Ben Galbraith

For many of us, XML has become a ubiquitous presence in application development, whether parsing, validating, or manipulating it. For many of us, all that XML is coupled with pain, in the form of tedious APIs (like, say, the W3C DOM API) and confusing technologies (oh, I don't know, W3C XML Schema?).

Advanced Swing: Architecture and Frameworks by Ben Galbraith

Are you spending more time plumbing your Swing applications than solving business problems? Has your Swing application grown out of control? This session is for you.

Being Productive with Java in the Enterprise by Ben Galbraith

It sounded like such a good idea back in the mid-nineties: based the Java platform on a standards-based, open community, and let anyone participate. There is no question that Sun's strategy for Java's stewardship via the JCP and sponsored open-source has yielded some enormous benefits. However, these have not been enjoyed without tremendous cost.

Introduction to Hibernate by Bruce Tate by Bruce Tate

O/RM (Object/Relational Mapping) seeks to eliminate repetitive or tedious work enabling the CRUD (create, read, update, delete) that underlies most applications. Hibernate is a popular, open-source O/RM tool that uses reflection (instead of code generation, like EJB, or bytecode injection, like JDO) to manage your persistence layer.

Stretching Java by Bruce Tate

In Stretching Java, we'll look at some of Java's limitations, see how other programming languages solve those problems, and look at how Java developers can implement those ideas in Java using open source frameworks, design strategies, and tools.

Where Agile meets Argyle: New processes in established companies by Bruce Tate

Agile programming is a collection of core principles and techniques that allow software developers to create lighter, more responsive applications, and to have fun doing it. Many established organizations are either openly or sub-consciously hostile to many of the principles of Agile development.

Beyond Java by Bruce Tate

All programming languages have a limited life span, and Java is no different. This is a philosophical session rather than a programming session. Sooner or later, Java will lose its leadership position. This session will explore Java's strengths and weaknesses. We'll try to understand whether conditions are ripe for alternatives to emerge, and what those alternatives may be.

Introduction to Spring by Bruce Tate

This session, for the Spring beginner, helps you: # Understand dependency injection and inversion of control # Know the meaning of lightweight containers and Spring # Understand the basic pieces of Spring # See core Spring modules in action, including Persistence, AOP, transactions. Attendees need not know anything about Spring. This session does talk about integration with core J2EE frameworks like JDBC and transactions.

Ruby on Rails by Dave Thomas

The Ruby on Rails framework has exploded onto the scene over the last few months. Propelled by some genuine benefits, and fueled by a whole lot of controversy, Rails seems here to stay. So, is it a Java killer? (No.) Is it a great way to develop certain classes of web application? (Yes.) Does it really deliver the 10-fold increase in developer productivity that some have claimed? (It depends...)

Herdin Racehorses and Racing Sheep by Dave Thomas

Are you frustrated by experts who can't tell you what to do, or by junior team members who refuse to see the big picture? How can you best develop careers: both yours and those of your teammates and managers? How can we learn to apply experience more effectively, and why do the many approaches designed to tame complexity actually end up increasing it?

Ruby for Java Programmers by Dave Thomas

Ruby recently enjoyed its tenth birthday. Instead of cake and candles, the community celebrated by releasing a wave of new libraries and frameworks that make Ruby programming even easier. This talk features some of the best of these, as we explore Ruby.

The State Machine Compiler by Eitan Suez

Classes will often bear various states. Examples include a user who may be "logged in" or "logged out," a bill that is "open" or "paid," or potentially a more complex situation where an object obeys a set of complex rules that determines which of a number of possible states that object is in. The Gang of Four gave us the State Pattern, a fairly straight-forward mechanism for developers to model and implement the behaviour of stateful objects. The State Pattern is only the beginning of the story. Robert Martin developed the State Machine Compiler and has taken the job of developing and maintaining stateful systems to a new level. Today, SMC is a well-maintained open source project hosted on sourceforge.net. Come learn about SMC, a fundamental tool for implementing stateful classes and systems that every software developer should have in his toolchest.

XML Data Binding with JiBX by Eitan Suez

JiBX is an open source XML data binding API for Java. JiBX is younger than most other APIs in this space (Castor XML, BEA XMLBeans, JAXB). JiBX's philosophy on data binding is that: [a] databinding should be fast, and [b] databinding frameworks should allow for the divergence and evolution of your codebase from its xml representation. JiBX excels on both counts and consequently is a practical tool for the purpose of data

binding. In this session, Eitan will be covering all aspects of Dennis Sosnoski's JiBX framework.

Cascading Style Sheets: a Programmer's Perspective by Eitan Suez

Today, the Cascading Style Sheets (CSS) specification is well supported by the major browsers (Mozilla, Safari, IE). CSS has become a practical tool for web content publishers that has helped turn heavy, buggy, and hard-to-maintain web sites into lean, clean, and stylish ones. CSS is sometimes stereotyped as a technology geared for graphic designers and artists. I beg to differ: I see CSS as a refactoring tool for content publishers and one that encourages content to become more strongly semantic. Come see a developer's perspective on CSS and how it can be applied to refactor your web content.

Naked Objects Applied by Eitan Suez

Join Eitan in this hands-on session on Naked Objects. This session uses the "learning by doing" approach to learning an API or framework. Naked Objects is a powerful tool that can give you a significant advantage in the development of business systems. It gives you the ability to prototype a software application so quickly that it can be performed during information gathering phases of a project. It gives you the power to codevelop the core business model of your application with a non-developer business expert at your side. No prerequisite knowledge of Naked Objects is required.

Software Development Techniques by Jared Richardson

Throughout our software careers we learn habits from our coworkers, from books we've read, and occasionally, from conferences we attend. Much of our competence comes from the tips and tricks we pick up as we go.

Pragmatic Tracer Bullets by Jared Richardson

Are your product designs hit or miss? Do you have trouble building a loosely coupled system? Is your code incestuous? Refactoring not an option with your code base? Tracer Bullets help keep your project out of the fire. Tracer Bullet Development: * helps you create great software * lends itself to an iterative cycle * can be used for demos early and often * is easily refactored * allows your teams to work in parallel * makes a very testable system

Software Tools That Make Life Easier by Jared Richardson

a.. Do you spend more time fighting your tools than writing code? b.. Do you avoid merging your code with your teammates because of #Integration Hell#? c.. Do the same bugs keep sneaking back into your product? d.. Do your builds depend on the roll of the dice? A good set of infrastructure tools can go a long way toward smoothing out these and other problems. Come see how to make your toolset work seamlessly in the background so you can Just Work. We'll cover source code management (SCM), build scripts, automated test harnesses, automatic builds, feature tracking and issue tracking.

Building Applications with the Spring Framework by Keith Donald

You'll see how to use Spring to assemble a complex system from a set of focused, loosely-coupled components. You'll experience through example how Spring enables agile development by allowing you to start simple, validate architectural choices early, and scale up as requirements demand.

Advanced Spring: What's New and What You Might Not Know About by Keith Donald

Spring 1.2 is out--Spring 1.3 is right on the horizon. As a broad, user-driven project with a large community, the newest releases offer a wealth of new features to be taken advantage of. This session focuses on demonstrating the most important, and how you can start leveraging them in your projects immediately.

Architecting JavaServer Faces Applications by Kito Mann

Over the past year, a lot of time has been spent explaining what JSF is, and how different pieces of it work. However, little attention has been given to the process of architecting applications. This makes JSF architecture seem like a black art, since there are so many possible approaches to the application's architecture.

Migrating from Struts to JSF by Kito Mann

As JavaServer Faces (JSF), the new standard Java web application framework, grows in popularity, development teams are beginning to evaluate different strategies for migrating from Struts to JSF.

Intro to JavaServer Faces by Kito Mann

JavaServer Faces (JSF) is a standard web user interface framework, developed under the Java Community Process (JSR 127), and released in March, 2004. JSF specifies a web user interface component model, complete with server-side event handling, validation, internationalization, page navigation, and declarative

mapping between user interface components and Java objects.

Struts Shale: Struts 2.0? by Kito Mann

With the growing popularity of new Java web frameworks, such as JavaServer Faces, Tapestry, and WebWork, Struts 1.x has lost its competitive edge in the web framework landscape. Recently, Craig McClanahan, the founder of Struts, initiated Struts Shale, a proposed next-generation framework built on top of JavaServer Faces.

Introduction to Portlets by Kito Mann

In late 2003, the Java Community Process released the Portlet API, designed to ease the progress of writing portlets for different portal environments. Using the Portlet API, developers can build reusable application components that work with portal servers from IBM, BEA, Oracle, Vignette, Apache, and other companies and open source organizations.

Managing Agile Projects: Dispelling the Myths by Mike Cohn

There is a myth that agile projects do not need project management and that they cannot be estimated and planned. In this session we will dispel those rumors and learn why the job of the agile project manager is to do more than just buy pizza and get out of the way.

An Introduction to User Stories for Software Requirements by Mike Cohn

The technique of expressing requirements as user stories is one of the most broadly applicable techniques introduced by Extreme Programming. User stories are an effective approach on all time constrained projects, not just those using XP.

Salvaging Struggling Projects: Digging for Gold Instead of Digging a Deeper Hole by Mike Cohn

Projects struggle for many reasons#overly aggressive deadlines, unproven technologies, scope creep, team dynamics, communication problems, and inter-team coordination are just some of the reasons. If not given attention, these problems can ultimately cause a project to fail entirely. However, if you act early and in the right way, most struggling projects can be turned around.

Project Economics: Selecting and Prioritizing High Value Projects by Mike Cohn

Almost all of us have worked on too many projects that have failed because of economic reasons rather than technical reasons. Just as the technical team is required to estimate the effort that will go into a project, a marketing or product management team should estimate the benefits of doing the project. Benefits can come in the form of additional sales, increased customer retention, increased operating efficiencies, and so on.

Overview of Agile Estimating and Planning by Mike Cohn

Estimating and planning are key skills. A good plan helps both the organization and the developers working on the project. In this session you#ll learn how an easy and effective approach to estimating and planning that can help you create more realistic plans.

Transitioning to Agile: A Dozen Keys to Success by Mike Cohn

Transitioning to an agile process from a traditional process is fraught with potential dangers. Attend this class and learn the dozen things you absolutely must do in order to succeed.

Pragmatic Extreme Programming by Neal Ford

This session talks about how to actually get XP done in the real world (and what to tell your boss). This session includes artifacts (like project tracking sheets) from real XP projects.

SOA and ESB: Next Wave of Enterprise Development or Return of the Son of CORBA? by Neal Ford

Are Service Oriented Architecture and Enterprise Service Buses the next wave of distributed computing or just the same old crap in a shiny new package? This session provides an overview of what most people agree is the definition of SOA and some of the characteristics of ESBs. I talk about EAI, your MOM, SOA, ESB, and all the other acronyms I can come up with.

Power Regular Expressions in Java by Neal Ford

Regular expressions should be an integral part of every developer#s toolbox, but most don#t realize how important it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

Web Application Security Vulnerabilities by Neal Ford

This session highlights common mistakes made by web programmers, stating the problems and avoidance techniques.

Language Oriented Programming Part 1: Theory by Neal Ford

This session shows how to use Java as the building block for domain-specific languages. It discusses the next revolution in programming: language-oriented programming and the nascent tools that support it.

Language Oriented Programming Part 2: Practice by Neal Ford

This session builds on the theory presented in part 1 and shows how to use the new breed of tools, Language Workbenches, to build your own domain languages.

Advanced Enterprise Debugging Techniques by Neal Ford

This session discusses techniques and tools for debugging enterprise applications (without using `System.out.println()`!)

Performance monitoring in J2EE applications by Ramnivas Laddad

J2EE has become the main new platform for enterprise application deployment. Good performance is an important requirement from the business viewpoint. Supporting this requirement needs application profiling during the development phases and performance monitoring after deploying the application. Come to this session to understand challenges and choice in monitoring J2EE applications.

What's new in AOP by Ramnivas Laddad

A lot is happening in the field of Aspect-oriented programming (AOP). AspectJ and AspectWerkz, the two leading AOP implementations, have merged bringing in their respective strengths. The merged version (AspectJ 5, currently in a milestone release) adds many new features aimed at simplifying writing and deploying aspects. The new features include an annotation-based and XML-based syntax to define aspects, support for new Java 5 concepts, and load-time weaving. The tools support for AOP continues to improve, as well. Further, the most popular IOC framework # Spring # enables integrating aspects written in AspectJ. There is also serious discussion and preliminary work going on to support AOP right into the VM itself. All in all, there is a lot to learn about the changes in the exciting field of AOP. This session is designed to help you get up to date with all these changes.

Introduction to Aspect-oriented programming with AspectJ by Ramnivas Laddad

Aspect Oriented Programming (AOP) enables modularizing implementation of crosscutting concerns that abound in practice: logging, tracing, dynamic profiling, error handling, service-level agreement, policy enforcement, pooling, caching, concurrency control, security, transaction management, business rules, and so forth. Traditional implementation of these concerns requires you to fuse their implementation with the core concern of a module. With AOP, you can implement each of the concerns in a separate module called aspect. The result of such modular implementation is simplified design, improved understandability, improved quality, reduced time to market, and expedited response to system requirement changes. Come to this session and learn all about how AOP can help you simplify developing complex systems.

Aspect-oriented programming: Myths and realities by Ramnivas Laddad

Aspect-oriented programming (AOP) promises to modularize crosscutting concerns. Like all new technologies, AOP has its share of over zealotry and unjustified criticism, neither of which is useful to developers deciding if they should use AOP in their applications. Attend this talk to understand the real deal behind AOP and change your perspective of AOP forever.

Guerrilla Web Techniques by Scott Davis

Frameworks? We don't need no stinkin' web frameworks. OK, so maybe that's overstating the case. Web frameworks do plenty of good things, but sometimes they can also be golden handcuffs. Too many web developers fall into the trap of thinking, "If it can't be done by my web framework, then it simply can't be done."

Real World Web Mapping by Scott Davis

In this presentation, we'll explore the top four mapping sites and show you how to take advantage of their free services. MapQuest, Yahoo Maps, Google Maps, and MSN Virtual Earth all bring slightly different capabilities to the table. These sites allow you to create your own interactive maps with minimum effort and no previous mapping experience. They take care of hosting the mapping data and making it easy to manipulate -- all you have to do is bring a little bit of know-how to the party.

Testing the Web Tier by Scott Davis

Hopefully your test plan involves more than, "Well, it compiled..." JUnit is fast becoming a required part of the modern Java developer's toolkit. Unit testing your Java classes is a great start, but your test plan shouldn't stop there. This talk will introduce several additional testing tools for the web developer -- HttpUnit, Canoo WebTest, and JMeter. These tools allow you to test a live website with no changes to the production code. Even better, you can test sites that have been implemented in technologies other than Java.

Pair Programming for the Single Programmer by Scott Davis

The full title of this talk is, "The Sound of One Hand Clapping, or How to Pair Program with a Single Programmer -- Scaling XP to Small Projects." Everyone talks about using J2EE for massive projects, but what about the lone wolf developer? Can they still apply the lessons learned from agile development methodologies to their everyday work?

Testing the Web Tier, Part 2 by Scott Davis

JUnit is more than a Java testing tool -- it is a testing framework that can be extended to test non-Java resources as well. In the first presentation in this series, we examined three JUnit extensions that allow you to functionally test your website. In this talk, we'll look at three more tools that web developers should have in their toolkit: JsUnit, DbUnit, and the W3C Markup Validation Service.

Programming Java Concurrency by Stuart Halloway

Java has always provided a model for concurrency and threads. With Java 1.5, this model received a major facelift. Learn how to use the new concurrency utilities to build responsive, scalable, and correct concurrent applications.

Unit Testing Java with Jython and JRuby by Stuart Halloway

JUnit is great. Jython and JRuby are even better. Unit testing libraries look the same everywhere, so why not use the one that lets you get your job done faster?

Cryptography for Programmers by Stuart Halloway

For centuries people have used crypto to build (and break) secure systems. Computers have only raised the pitch of conflict, providing enormous cryptographic power at commodity prices. Most programmers do not write their own crypto libraries, instead relying on the services of an operating system or virtual machine. But even with all this support, building secure systems is a daunting task.

Java Platform Security and JAAS by Stuart Halloway

The Java platform is built from the ground up with security in mind. This talk will introduce the security features of the J2SE, building quickly from the basic classes to realistic examples.

Agile Methodologies by Venkat Subramaniam

Agile development is picking up steam. You have heard about eXtreme Programming(XP). What other Agile methodologies are you familiar with and what do they bring of interest or significant to the table of Agility? More important, why should you learn about these different methodologies instead of simply focusing on one? There is no one shoe that fits all. Any methodology that requires you to follow it in totality and not let you adapt is rather dogmatic, not pragmatic. To be effective we have to take the best of different approaches and apply to our projects base on our specific needs.

Programming with Mock objects by Venkat Subramaniam

You are convinced that Test Driven Development is good for you and your project. You realize the benefits it has to offer. What's holding you back? All the code and components that your code so heavily depends on is most likely making you wonder if TDD is really for you. We will start out by looking at dependency and dependency inversion. Then we will discuss how mock objects can help separate our code from its dependencies.

Groovy for Java Programmers by Venkat Subramaniam

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Groovy bring this excitement to the Java platform with its ability to generate byte code. You can use Groovy instead of Java for some parts of your application. By learning it, you can switch between the languages where you consider fit.

Java 5 Features, What's in it for you? by Venkat Subramaniam

A number of new features have been introduced in Java. What benefit do these features offer you. Are there issues with using these features. For instance, when should you use annotation? The objective of this

presentation is not simply to introduce you to the features, but to the effective use of these as well.

Prudent OO Design by Venkat Subramaniam

Is your code object-oriented? Developing with objects involves more than using languages like Java, C#, C++ or Smalltalk for that matter. From time to time, the OO paradigm stumps even expert developers. Agile programming becomes a mere act of hack if we code without knowing the OO principles. What are these principles # the ones that influence your design? In this presentation the speaker will present some of the challenges that are fundamental in nature. Then he will present OO Design principles and good practices for prudent development of OO code.

Good, Bad and Ugly of Java Generics by Venkat Subramaniam

Java introduced Generics in the 1.5 version (Java 5). What are the capabilities of Generics? How do you use it? Are there some gotchas in using it? In this example driven presentation, we will start at the basics of generics and look at its capabilities. We will then look at some of the under the hood details on generics implementation. We will then delve into the details of some of the changes to Java libraries to accommodate generics. Finally we will take a look at some restrictions and pitfalls that we need to be familiar with when it comes to practical and prudent use of generics.