

Northern Virginia Software Symposium 2006

Sheraton Reston - Reston, VA

April 28 - 30, 2006

(session listing as of 4/26/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Northern Virginia Software Symposium coming to Reston on April 28 - 30, 2006. 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 75 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 Northern Virginia Software Symposium will offer 6 concurrent sessions over three days with over 65 sessions to choose from. The following topics will be featured:

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

Registration Fees

Attendees	Before 4/10/2006	After 4/10/2006
1-4	\$725	\$825
5-9	\$650	\$725
10-14	\$625	\$700
15-24	\$600	\$675
25+	\$575	\$650

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) Great NFJS Swag
- 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.

Northern Virginia Software Symposium 2006

Sheraton Reston - Reston, VA

April 28 - 30, 2006

(session listing as of 4/26/2006)

Practically Groovy: Real World Groovy for Thrill Seekers by Andrew Glover

The key to incorporating any tool into your development practice is knowing when to use it and when to leave it in the box. Dynamic languages can be an extremely powerful addition to your toolkit, but only when applied properly to appropriate scenarios.

Unit Testing Best Practices by Andrew Glover

In the years since JUnit's introduction, a number of frameworks have been built to enhance its utility for testing and validating XML, controlling the state of a database, testing legacy code, performance testing, and functional web testing.

Unit Testing Java Objects with Groovy by Andrew Glover

What makes Groovy particularly appealing with respect to other scripting platforms is its seamless integration with the Java platform. Because it's based on the Java language (unlike other alternate languages for the JRE, which tend to be based on earlier predecessors), Groovy presents an incredibly short learning curve for the Java developer. And once that learning curve has straightened out, Groovy can offer an unparalleled rapid development platform.

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.

Introducing the Semantic Web 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 the new vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. While there are many naysayers chiding such grand visions, there are also pragmatic and useful technologies emerging that can be applied today.

Experiencing the Semantic Web by Brian Sletten

If you attended the "Introducing the Semantic Web" talk or have been paying attention, you understand where much of the current Web falls down. By building on languages like RDF and OWL our software will be able to do more for us to improve search and knowledge discovery. This talk is a more hands on introduction to using the core technologies of the Semantic Web for managing, storing querying and inferencing over metadata.

NetKernel : XML Processing for the 21st Century by Brian Sletten

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

Herdng 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?

Testing your Rails Application by Dave Thomas

The Ruby on Rails framework has unit and functional testing baked right in. In this talk we'll see how easy it is to get started with testing in Rails, and we'll explore just how deep the testing support goes.

Using Ajax with Ruby on Rails by Dave Thomas

Ajax is becoming a requirement for new applications: it creates richer user experiences and more dynamic applications. However, doing Ajax by hand is difficult and error prone. The good news is that if you use Rails,

you don't have to do Ajax the hard way.

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...)

Losing Battles and Winning Wars: Adopting Agile by David Hussman

Adopting agile is different for each company, but most companies will go through some amount of change during the adoption of agile.

Creating, Telling, and Tracking User Stories by David Hussman

The participants of this session will become agile customers and product owners, creating stories for project, organizing them into themes, and using them during mock planning activities.

#Show Me the Numbers# - Agile Planning Tools and Techniques by David Hussman

This session will focus on tools and techniques for tracking an agile project plan from creation to project completion.

Automating Business Value with FIT and FitNesse by David Hussman

The presentation will briefly discuss stories, the origin and authoring of story tests, and a demonstration of how FIT and FitNesse (FIT living within a Wiki) can be used to automate acceptance tests.

The Agile Enterprise by David Hussman

As with many methodologies, moving agile into larger organizations poses larger challenges. There are many factors outside the developer world that can crash all the benefits of agile without regard to its success.

Hibernate by Example by Eitan Suez

This talk covers the core of the Hibernate Object/Relational Mapping framework by example; that is: in a hands-on manner.

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.

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.

Under the Hood of Java Memory Management by Glenn Vanderburg

Most of the time, Java's automatic memory management works really well#it's one of the things that makes programming in Java a pleasant and productive experience, and it's nice that we don't have to worry about managing memory manually. However, although it's usually nice to ignore memory management, occasionally we have to pay close attention. Sometimes we need to take control of certain aspects of memory management. Sometimes Java programs do exhibit memory leaks, or unacceptably long garbage

collection pauses, or very poor overall performance. But because Java's memory management is supposed to be "fully automatic," it can be difficult to find out what's really going on inside the VM.

JavaScript Exposed: There's a Real Programming Language in There! (Part 2) by Glenn Vanderburg

Building on part 1, this talk dives deep into JavaScript's object model. We'll see how it differs from more mainstream object-oriented languages, and why. We'll explore how to hide some of those differences, as well as the reasons you might not want to. Additionally, we'll cover useful tools for JavaScript testing, debugging, and profiling.

Java Performance Myths by Glenn Vanderburg

Performance myths about the Java platform abound, from the general "Java is slow", to the more specific "reflection is slow", "allocation is slow", "synchronization is slow", "garbage collection is slow", etc. Many of these myths have their root in fact (in JDK 1.0, everything was slow); today, not only are many of these statements not true, but Java performance has surpassed that of C in many areas, such as memory management.

JavaScript Exposed: There's a Real Programming Language in There! (Part 1) by Glenn Vanderburg

With the sudden importance of Ajax, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of just copying other people's poorly written examples.

Modern Project Infrastructures by Glenn Vanderburg

The support infrastructure for your software project is a crucial factor for success. A new generation of tools offers significant benefits over their predecessors. This talk discusses how to choose the right mix of tools for a top-shelf project infrastructure.

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: Part Two by Jared Richardson

This talk is a continuation of Part One of the Tools talk. During Part Two we'll cover Continuous Integration, automated testing, bug tracking, and feature tracking.

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.

It's All About the Platform by Jared Richardson

Validating a platform is hard work. We'll look at several approaches and strategies you can use to keep your platform solid.

Software Tools That Make Life Easier: Part One 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.

Unobtrusive JavaScript by Jason Harwig

With the recent popularity of AJAX, JavaScript has entered a renaissance in the last year, bringing with it both cool new features and the potential for a mess. An increased number of scripts executing in web pages can lead to embedded behaviors fighting with each other and crowding "onEvent" attributes. Most developers know that embedding script in html is not a best practice, but creating reusable and gracefully degrading libraries in JavaScript is not a widely known skill.

Advanced Hibernate by Justin Gehtland

Hibernate is easy to get started with, but can sometimes be hard to make efficient or secure. In fact, the default settings for Hibernate create applications that will run slowly, cause unwanted round trips to the database, and may be more restrictive and/or permissive from a security standpoint than you would otherwise want.

Spring Dependency Injection by Justin Gehtland

Dependency Injection (DI) is the cornerstone of Spring. The core concept is quite simple, but (surprise!) actual practice can become complex. To take full advantage of Spring DI, you need to understand not only the basics on configuration, but also the container lifecycle model and the various hooks provided by the framework.

Java Platform Security and JAAS by Justin Gehtland

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. You will learn the core APIs: SecurityManager, AccessController, Permissions and Policy JAAS Subjects, Principals, and LoginModules

Ajax Architecture by Justin Gehtland

Ajax applications have unique architectural challenges and opportunities. This presentation will show you how to take advantage of the Ajax's strengths, and work around its quirks.

Spring Intro by Justin Gehtland

The Spring framework is one of the fastest growing open source frameworks. New job postings are gaining rapidly, and many customers are adopting Spring instead of heavier alternatives. In this session, we'll introduce Spring. You'll see how Spring can give you much of the power of EJB, without the complexity or pain. Spring uses concepts like dependency injection and aspect oriented programming to ease standard enterprise development. Spring developers write plain, ordinary Java objects (POJOs), instead of sophisticated components. In this session, you'll see a basic Spring application. You'll also see some details about some of the enterprise integration strategies, including: # Spring AOP # Transactions # Persistence # Model/view/controller When the session is over, you won't be an expert, but you should have a much clearer understanding of what Spring does, what it doesn't do, and why it's growing so rapidly.

Spring Security with ACEGI by Justin Gehtland

Spring offers developers a simpler, more robust method for configuring applications. These benefits extend to security through the ACEGI framework. ACEGI makes the otherwise daunting task of securing your application logical and straightforward. More importantly, through its support for single sign-on provision through Yale's CAS system and its ability to provide instance-level authorization, Spring extends the common security model of most J2EE apps beyond what they are traditionally capable of.

JavaScript for Ajax Programmers by Justin Gehtland

This presentation covers JavaScript from the perspective of an Ajax programmer. We assume that you may be using an Ajax toolkit, but still need to be able to read, modify, and test the JavaScript code in your application. You will learn the common idioms of JavaScript by looking at working code from the Ajax toolkits themselves.

J2EE Command Pattern Architecture by Mark Richards

Tired of dealing with EJBs but cannot use other frameworks like Spring? How would you like to replace all of your remote Stateless Session Beans with POJOs and still access them remotely within J2EE? By using the J2EE Command Pattern we can write EJBs as POJOs and solve many of the issues facing EJB, including testability, configuration complexity, and performance, and still remain within our behemoth J2EE environment. The J2EE Command Pattern is a simple pattern that can significantly reduce the complexity of large-scale J2EE enterprise applications. In this session we will explore the numerous issues facing a typical EJB architecture and learn how the use of the J2EE Command Pattern can solve these issues. We will walk through the different design alternatives and see how the command pattern is implemented. Through interactive coding examples you will learn what components make up the Command Pattern framework and what simple coding changes are required to convert a complex EJB-based application to a remotable POJO-based application.

EJB 3.0 and New Java Persistence API by Mark Richards

The new EJB 3.0 spec (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will take a look at the new EJB 3.0 spec and the

new Java Persistence API. Included in this session will be a discussion about Java metadata annotations, simplification of enterprise beans (session and message-driven beans), interceptors, changes in transaction processing, and how the new Java Persistence API works. During the session I will be demonstrating how the EJB 3.0 spec differs from the EJB 2.1 spec through code example comparisons. I will also be discussing how the new Java Persistence API compares to related Java persistence options and whether we should be excited about the new persistence API or (yawn) sticking with what we have.

Understanding the Role of an ESB by Mark Richards

The Enterprise Service Bus is an integral part of any Service-Oriented Architecture. It is the glue that binds the business services to the client applications. There are many ESB third-party products and solutions in the marketplace, but in most cases these products only serve to further confuse us in terms of what an ESB is, particularly when you consider that an ESB is really an architectural component that has many different implementations. In this session we will take a detailed, product-agnostic look at the role of an ESB and the capabilities an ESB must provide. Through this session you will learn what an ESB is, the role of an ESB, what capabilities it provides, and the various ways an ESB can be implemented. We will also take a close look at the Java Business Integration (JBI) specification (JSR-208) and see what impact it will have with the ESB world. With the information from this session you will learn how to determine your own specific requirements for an ESB and then match those requirements to the product space rather than having the tail wag the dog!

WebWork/Struts 2.0 & Spring by Matthew Porter

From its early pre-release, WebWork 2 (the core of Struts 2.0) has included Inversion of Control. This implementation, based upon interfaces, served the WebWork community well due to its simplicity; however, it lacked a number of features. Beginning in version 2.2 and continuing with Struts 2.0, the internal IoC container has been deprecated and replaced with the more powerful Spring IoC.

WebWork 2.2 / Struts 2.0 by Matthew Porter

In November 2005, WebWork and Struts developers announced a merger between the two framework to build Struts Action Framework 2.0. The resulting product will be built upon the foundation of WebWork 2.2 with a slew of new features, including a migration layer for Struts Classic. This session introduces the audience to this revolutionary framework and teaches some new concepts to those already using WebWork.

Language-oriented Programming and Language Workbenches: Building Domain Languages atop Java 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.

Real-world Agile Development by Neal Ford

Lots of developers want to use Agile development technique but don't know where to start. This session discusses how to get started with Agility, the key benefits you can expect, and the pitfalls to avoid.

Advanced Enterprise Debugging Techniques by Neal Ford

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

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.

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.

Testing with Selenium by Neal Ford

This session describes the use and workings of Selenium, the open source web user interface testing tool.

Spring AOP in Depth by Ramnivas Laddad

Support for aspect-oriented programming is an important part of the Spring framework. It is the AOP support

that allows keeping implementation of functionality such as transaction management and security out of your POJOs. While many developers only use aspects provided with Spring, once you understand how it all works, you can make a better use of those aspects, extend them, and write brand new aspects.

Performance Monitoring in J2EE Applications by Ramnivas Laddad

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

Enterprise AOP with AspectJ by Ramnivas Laddad

Enterprise application development is a gold mine for applications of AOP. There are many crosscutting concerns found in a typical enterprise application, ranging from well-known security and transaction management to application- and technology-specific concerns. Using AOP leads to implementations that are easy to understand and easy to change.

The State of 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, 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.

Introducing the Eclipse Rich Client Platform by Scott Delap

Rich client application development using Java can be intimidating giving the vast flexibility in application design and structure. It also can be frustrating to create the large number of support services (persistence, menus, event and job frameworks) that a large scale rich client applications needs. The Eclipse Rich Client Platform is one project attempting to solve these issues by providing a core infrastructure that not only provides the day to day services a rich client application developer needs, but also providing a suggested path to guide you down the road of designing your application. This presentation introduces both the Eclipse RCP and the tools provided by the Eclipse IDE that assist developers in writing RCP apps.

Real World Web Services by Scott Davis

In this talk, we'll survey the web services exposed by leading websites (Google, Yahoo, Amazon, eBay) and discuss how they are driving the AJAX revolution. You'll see examples of RESTful, SOAP, and JSON web services, as well as the strengths and weaknesses of each.

Creating Polished Swing Applications by Scott Delap

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.

Holistic Testing by Scott Davis

Mark Twain once said, "Everyone talks about the weather, but nobody does anything about it." Do you feel the same way about Unit Testing? Are you actively testing your code, or are you just thinking about testing your code... some day... once you get some more free time...

Easing into Agile by Scott Davis

How do you get started with an Agile development methodology? Everyone has been talking about eXtreme Programming for years, but how do you get it introduced to your team? Many times, you're not simply transitioning from from one methodology to another -- you're introducing a methodology for the first time. Adding structure to a previously unstructured endeavor. Adding a touch of discipline where programmers once roamed free.

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."

Ajax, Flash, and Java - Choosing The Right Rich Client Technology for Your Next Project by Scott Delap

Today's users are beginning to demand richer and richer application experiences. Plain html pages simply don't cut it anymore. Applications like Google Maps (Ajax) and Yahoo Maps (Flash) show how the UI experience can be pushed to the next level. As an IT manager, how do you decide which route to take however? Should you use Ajax because it is the new "it" technology. Is Flash a viable option with its 95%+ browser availability? Perhaps Java deployed through web start is really the best choice in contrast to what the buzz would lead you to believe. This presentation takes a look at these three core rich client technologies from both deployment/user experience and ease of development perspectives.

Working with Rules Engines by Venkat Subramaniam

Rule based programming allows us to develop applications using declarative rules. These can simplify development in applications where such rules based knowledge is used for decision making.

Open Source Tools for Agile Development by Venkat Subramaniam

As a Java developer, you have taken the time to learn the basics of the language and relevant parts of its rich API. However, you need more than that to develop serious industrial strength applications. In this presentation, the speaker will introduce you to a number of open source tools which you can use to improve your application quality and your development process.

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.

Portal Standards and implementation by Venkat Subramaniam

Portals and Portlets allow you to personalize your web application. However, developing and deploying portlets across different portals can be a challenge. What is WSRP and JSR-168. How are these related and how are these different? Are these competing technologies or do they work with each other?

Practices of an Agile Developer by Venkat Subramaniam

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speaker will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the book with the same title as the talk.

Refactoring your code - a key step in agility by Venkat Subramaniam

Refactoring is one of the core practices in Agile Software Development. Refactoring is based on some core principles that apply to more than writing good code. But, what's refactoring? Why should you do it? How do you go about doing that? What tools are available to successfully refactor your App?