

Northern Virginia Software Symposium 2005

Sheraton Reston Hotel - Reston, VA

October 28 - 30, 2005

(session listing as of 10/26/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Northern Virginia Software Symposium coming to Reston on October 28 - 30, 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 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) .Net
- 2) Architecture
- 3) ServerSide Java
- 4) XML / Web Services
- 5) Core Java
- 6) Client Side Java

Registration Fees

| Attendees | Before 10/10/2005 | After 10/10/2005 |
|-----------|-------------------|------------------|
| 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.

Northern Virginia Software Symposium 2005

Sheraton Reston Hotel - Reston, VA

October 28 - 30, 2005

(session listing as of 10/26/2005)

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.

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?).

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.

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.

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.

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?

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 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.

JmDNS : Easy Service Discovery for the 21st Century by Brian Sletten

Configuration files are so 90's. Software of the 21st Century should be able to find related services and components without users having to specify particular configurations at start up. The IETF's ZeroConf multicast DNS protocol was designed to solve exactly this problem. jmDNS is Java-based open source implementation of this capability that allows local-link applications to find and use automagically discovered

capabilities. Apple's Rendezvous technology is another open-source ZeroConf implementation behind many of the exciting applications it is building for OS X these days. Come learn how you can interact with these or your own service discovery-savvy applications without even having to learn how to spell UDDI. Bring your wireless notebooks to participate in a service-oriented environment (please have a working Java environment as we won't have time to debug installation issues).

Exploring the Semantic Web with Kowari 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. The Kowari project is an open source triplestore designed to facilitate the storage, management and inference of RDF and OWL documents, the main formats used for the Semantic Web. Come learn about the technologies involved in this vision and how you can store and manipulate "knowledge" in this next-generation schemaless database.

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.

Introduction to Hibernate 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.

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.

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...)

Herding 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.

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.

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.

Choosing An Agile Path: Agile Case Studies by David Hussman

With more and more companies choosing to take an agile path, the bounds of agile ways and means are

expanding. Agile implementations differ a great deal depending on the company, the project or product, and the players.

Overview of Agile Estimating & Planning by David Hussman

Planning is important even for projects using agile processes such as XP, Scrum, or one of the many other agile processes. Unfortunately, we've all seen so many worthless plans that we'd like to throw them away altogether. The good news is that it is possible to create a project plan that is useful, accurate and communicative.

Writing and Telling User Stories by David Hussman

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 many projects, not just those using XP.

Ruby on Rails by David Geary

At about the same time Java was brewing, another language from the far east entered the landscape with hardly any notice. Carefully crafted by Japanese devotees, Ruby, a potent mix of SmallTalk, Python, and Perl, toiled in relative obscurity as the marketplace moved in droves to Java. Today we have J2EE, the 800-pound gorilla of enterprise development. That 800 pounds cuts both ways: J2EE is powerful, but it's a complicated beast with a long and steep learning curve that sports a dissing array of peripheral open-source software. And J2EE has many design compromises and idiosyncrasies that reflect its growth and evolution. Some J2EE developers have begun to wonder if there's a better way...

Transitioning to Agile # Keys to Success by David Hussman

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

Felix: A bag of Tricks for Java Server Faces by David Geary

Okay, so you know a little about JSF. You understand managed beans, action outcomes and how to attach standard JSF validators to components in a JSP page. But there is a great deal of functionality that the average web application supports that JSF doesn't provide out of the box. For example, wouldn't you like to have JSF automatically place asterisks in front of labels for required fields? You are going to implement client-side validation, which JSF does not support out of the box, aren't you? Of course, you're going to test your application, right? And don't forget to trap unauthorized use of the back button.

Shale: The Next Struts? by David Geary

Struts is the most popular Java-based Web application framework today, but that's rapidly changing. There's a newcomer on the block, a leaner, meaner, better-designed framework loosely based on Struts that's poised to dethrone Struts as the reigning king of Java-based web application frameworks. That framework, of course, is JavaServer Faces. Craig McClanahan, the father of Struts and the co-spec lead for JSF 1.0, has proposed reinventing Struts for Struts 2.0 as a set of services for JSF applications. That new framework, which has no direct ties to Struts as we know it, is called Shale.

An Introduction to JavaServer Faces by David Geary

There are a lot of Java-based web application frameworks, but how many of them are: 1. Based on the most popular open-source framework (Struts), and 2. The standard that must be supported by every J2EE 1.5 container? JavaServer Faces (JSF) debuted in the Spring of 2004. Throughout the rest of 2004, JSF gained momentum with a handful of books and a growing user community that includes the popular MyFaces open-source JSF implementation which has recently moved from SourceForge to Apache. Perhaps the most telling sign of the times is Craig McClanahan's proposal for Struts 2.0, code named Shale, which reinvents Struts as a set of services for JSF applications.

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.

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.

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.

J2EE Transaction Management Part 1 by Mark Richards

Although the EJB container isolates us from most of the complexities involving transaction management, there are still a number of things we need to be aware of when dealing with transactions within the J2EE container. Too often transaction management is an afterthought in the design and development process, which leads to applications that have problems with data integrity, data consistency, and overall stability and reliability. In this session we will explore the three transaction models that J2EE supports (Local, Programmatic, and Declarative), and discuss the advantages, disadvantages, and pitfalls within each of these models, when it makes sense to use each transaction model, and under what situations these models are appropriate and inappropriate. We will spend most of our time on the Declarative transaction model, otherwise known as container managed transactions (CMT). Within this model we will explore some common pitfalls and look at the best practices within this model. Through coding examples and real-world scenarios, you will learn how to properly handle exceptions, how to correctly use transaction attributes, and how the isolation level can affect transaction and application behavior. We will also discuss the problems encountered with Entity Bean data caching and data synchronization within the context of JTA transactions. This session is the first part of a 3 hour session.

Using the J2EE Command Pattern to Simplify Client-Server Communications by Mark Richards

The latest buzz in the industry relating to EJB#s is to avoid them at all costs. However, there are many times application architectures require the use of remote services using EJB#s. Using EJB#s (specifically Stateless Session Beans) can sometimes lead to configuration complexity, transactional complexity, performance issues, and testing complexity. In this session we will see how the use of the J2EE Command Pattern can address these issues. The J2EE Command Pattern is an important design pattern that every developer and architect should be familiar with. We will first take a look at how the Command Pattern can help resolve some of the complexity and performance issues, and then look at two different implementations of the Command Pattern; Dynamic Commands and Static Mapped Commands. Through this session you will learn when the J2EE Command Pattern is appropriate, the different ways of implementing the pattern, and the implications of each approach.

EJB 3.0 and Java Persistence API Review 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.

J2EE Transaction Management Part 2 by Mark Richards

This session is the second part of a 3 hour transaction management session. In this session we will explore some of the more advanced features of J2EE transaction management. We will pick up where we left off from the first session by taking a detailed look at XA and J2EE distributed transaction processing, and how

to coordinate multiple resources within a single business transaction. Within the XA discussion you will learn what XA is, what the relationship is between JTA and XA, when you should use XA within J2EE applications, and how to enable JMS and DBMS resources to run under XA. We will also explore XA Drivers, and discuss the many issues surrounding XA Drivers within J2EE. We will then look at how to build an effective transaction design strategy by looking at three primary transaction strategy design patterns. We will review sample transaction code and settings within each pattern, and see how these patterns fit into different J2EE application architectures. We will end by looking at the steps for migrating from from a Local Transaction Model to a Declarative Transaction Model.

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!

A Pragmatic Look at Agile Architecture by Mark Richards

Designing application and enterprise architectures is a complex process. We follow defined processes, create lots of attractive architecture diagrams, kill lots of trees producing hundreds of pages of architecture documentation, and yet we find that in many cases the architectures we design are not followed by developers or simply not understood by the stakeholders or development community. As a result, either the delivered software does not match the original architecture, or the architect works 25 hours a day to ensure that the software is in compliance with the original architecture. This is a case where Agile Architecture can help. Agile architecture is an architecture process that leverages Agile principles and applies them to the design of application and enterprise architectures. During this session you will learn the shortcomings of standard architecture processes and why they sometimes go wrong. We will then look at some Agile principles and see how we can apply these methods to simplify the architecture process and produce better architectures. We will also look at some pragmatic techniques for architecture design and diagramming that support Agile Architecture, and learn how ATAM (Architecture Trade-off Analysis Method) can be applied to the concept of Agile Architecture. Of course, no pragmatic session would be complete without a discussion of the issues and shortcomings of the Agile Architecture process itself, so we will explore where the Agile Architecture process sometimes fails to live up to its promise. There will be plenty of room for discussion in this session, so please bring your architecture pains (and successes) with you so we can engage in a lively discussion.

Development Infrastructure Patterns by Paul Duvall

Design Patterns became part of the software development industry mainstream in the mid-1990s with the release of the Go4 Design Patterns book. Since then, architecture, design, and more recently, organizational patterns have become a part of our nomenclature. But, what about the software that helps us develop and deliver the software to our users: the software development infrastructure?

Continuous Integration using CruiseControl and CVS by Paul Duvall

Continuous Integration (CI) is the process of continually building and testing your software under development. It is identified as a core XP practice, although it works with many software development processes.

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.

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.

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.

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.

Scaling Agility by Ryan Shriver

Scaling Agility covers specific principles and practices for implementing an agile process for teams working on large-scale, complex software projects. Throughout the presentation I'll reference my direct experience with implementing these practices on my current engagement, a multi-million dollar project developing enterprise lending software for the banking and automotive industries.

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.

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."

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?

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.

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.

Simpler Builds and Deployment with Ruby Rake and RubyGems by Stuart Halloway

Have fun automating your builds and deployment!

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.

Web Services on Rails by Stuart Halloway

Using the ActionWebService pack that ships with Rails, you can rapidly create robust web services with the same productivity you see when developing classic web applications with Rails.

Programming XML in Ruby by Stuart Halloway

Ruby may not need XML, but XML sure benefits from Ruby!

Effective Enterprise Java: Security by Ted Neward

Security's become a hot topic among enterprise developers in recent years, but to many developers, security is still the white elephant in the middle of the room. Discussions about security usually begin with, "Uh, we'll worry about that later", or, "Start with two really large prime numbers.....". Security isn't as hard as developers make it out to be, but it is something that developers need to face and recognize.

Passing Messages: A Flexible, Powerful and Extensible Communication Model by Ted Neward

Over the last decade, focus in inter-process communication has centered on Remote Procedure Calls (RPC) and its object-oriented equivalents.

Introduction to Web services, 2005 edition by Ted Neward

WSDL, and Schema and SOAP, oh my! It's 2005, and the Web services landscape looks even more confusing than it did two years ago, despite all sorts of promises to the contrary. What's it all mean, and how the heck did we get here when the original goal was to try and keep it all simple?

Effective Enterprise Java: State Management by Ted Neward

Persisting data at first seems like such a simple idea, and yet thanks to issues like the object/relational impedance mismatch, it represents one of the most difficult parts of the enterprise system to get right. What's a poor enterprise Java developer to do?

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.

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.

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.

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.

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.

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.