

Lone Star Software Symposium: Austin 2005

Embassy Suites Austin North - Austin, TX

August 12 - 14, 2005

(session listing as of 8/11/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Lone Star Software Symposium: Austin coming to Austin on August 12 - 14, 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 Lone Star Software Symposium: Austin will offer 6 concurrent sessions over three days with over 70 sessions to choose from. The following topics will be featured:

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

Registration Fees

Attendees	Before 7/18/2005	After 7/18/2005
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: jjzimmerman@nofluffjuststuff.com or (303)469-0486.

Lone Star Software Symposium: Austin 2005

Embassy Suites Austin North - Austin, TX

August 12 - 14, 2005

(session listing as of 8/11/2005)

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.

AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript by Ben Galbraith

As recent high-profile web apps such as Google's GMail have shown, modern browsers are capable of natively rendering web apps with highly dynamic and compelling UIs - fetching server data without page refreshes, animating and manipulating page contents on-the-fly, even offline use. The line between web and "desktop" apps is blurring.

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.

SWT Fundamentals by Ben Galbraith

The Eclipse project's SWT GUI toolkit provides one of the only viable alternatives to Swing for creating so-called rich client applications in Java. Whereas Swing paints its own widgets and has distinguished itself with a complex (and often obtuse) API, SWT relies on the host operating system for widget rendering and sports a simple, clean API. If your goal is to create a Java application that "looks" like a normal Windows application (or OS X, or Linux), SWT will revolutionize your world. In this session, I introduce SWT from the ground up. I start at a high-level, but quickly move into the details of SWT's API. By the presentation's end, attendees will have a solid understanding of SWT.

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 SWT and JFace by Ben Galbraith

This session picks up where SWT Fundamentals leaves off. Among the advanced topics I discuss are creating custom SWT widgets and exploring tight native integration. I combine another compelling topic with the advanced SWT material: JFace. SWT is a more akin to AWT than Swing; it's concerned more with wrapping native functionality than providing any high-level abstractions. JFace is an API on top of SWT that provides such abstractions. The combination of SWT and JFace is comparable to Swing. My coverage of JFace includes an introduction to several of its frameworks, such as the Viewer and Window frameworks, along with many examples. Learning JFace will enable you to write complex SWT applications much faster.

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.

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.

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.

Stretching Java by Bruce Tate

As dynamic languages become more popular, it pays to look at frameworks based in other languages.

Politics of Persistence by Bruce Tate

This session will help a Java developer choose a persistence framework. After the session, you will # Understand the core strengths and weaknesses of the main persistence frameworks in the Java space # Understand where marketing influences can impact persistence # Know what's going on behind the scenes to impact the persistence picture # Answer questions about persistence frameworks that might not be mainstream

Thinking Inside the Box: Building Spring-Enabled Portlet Applications by Craig Walls

Windows changed everything. Back in the days of MS-DOS, you could only run one application at a time. Switching between writing a letter and balancing your checkbook involved closing a word processor and opening a spreadsheet. But now you can be running dozens of applications simultaneously, each inside its own window. And now switching from one application to another may be as simple as a shift of your eye or a click of the mouse button.

At Your Service: Service-Oriented Spring by Craig Walls

Where Spring promotes loose-coupling between your application objects, service-oriented architecture (SOA) encourages loose-coupling between applications that interact with each other.

Committing Acts of Subversion: The next generation of version control by Craig Walls

Mistakes happen. Bad ideas happen. Have you ever felt that sinking feeling when you accidentally erase an entire directory of source code? Or have you ever realized that the oh-so-clever refactoring you applied yesterday is causing performance issues today? Wouldn't it be great if you could turn back time?

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.

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 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...)

OpenSource Ecosystems by Dave Thomas

Open Source communities produce high quality software with little management and (typically) no pay. Most people looking at open source focus on using this software in their projects.

Killer Web UIs by David Geary

User interfaces are usually the most turbulent aspect of an application during development. Constant tinkering with the UI means constant changes to your code, so as a UI developer, you want to minimize the

scope and effects of those code changes. Open-source Java provides two powerful software packages that help you manage UI complexity: Tiles and Sitemesh. Tiles composes webpages from discrete regions of your user interface known as tiles. A tile contains a JSP page for layout and one or more JSP pages for content. Sitemesh decorates webpages with decorators that can be associated with URL patterns. Once you set up your decorators, you can decorate pages that match a decorator's URL pattern.

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.

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.

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.

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.

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.

Seaside: A Radical Web Framework by Glenn Vanderburg

We've been writing web applications now for 10 years, and they're still no fun. They're awkward and clumsy to write. Internally, they're overly complicated (which almost invariably means that they're buggy). Meanwhile, they're usually too primitive externally. To put it another way: the web programming model is so cumbersome for programmers that the users pay through reduced features, clumsy interaction, bugs, and poor performance. There's a better way. I know -- who needs another web framework? But Seaside makes even Rails look primitive.

Java Collections Power Techniques by Glenn Vanderburg

The Java Collections framework is a cornerstone of Java development. It's been a part of J2SE for six years now. Every Java developer knows it how to create Lists, Maps, and Sets, how to put things into them and take things out, and how to iterate over the contents. But there's a lot more to the collections framework than that -- and very few programmers really know how to exploit the power that's just under the surface.

JavaScript Exposed: There's a Real Programming Language In There! by Glenn Vanderburg

Now that web browsers are settling down and Ajax applications are on the rise, 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. This talk takes that approach to JavaScript. It's a "no web pages" introduction to the JavaScript language.

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.

New Features in Java 5 by Jason Hunter

The new Java 5 release introduces a number of significant Java language enhancements: generics, typesafe enums, autoboxing, an enhanced "for" loop, a static import facility, and a general-purpose metadata facility. This talk gives an overview of the changes and helps you understand what all the funny new syntax means.

Forgotten Algorithms by Jason Hunter

There are many interesting and useful algorithms that people just don't remember or never learned. The Boyer-Moore string search algorithm is one prime example. The randomized skip list is another. Both solve common problems with wonderful flair and finesse -- and performance-wise they blow the pants off brute force solutions. This session covers these two algorithms plus several others. It's like your college algorithms course but with a practical bent and absolutely zero proofs. Extra bonus: The Google PageRank algorithm.

Extreme Web Caching by Jason Hunter

Web Caching is very important for high traffic, high performance web site but few people know all the professional-level strategies. In this talk I'll share some of the tricks of the trade, including advanced tips from Yahoo's Mike Radwin. We'll start with the basics: using client-side caches, conditional get, and proxies. Then we'll talk about more advanced features: how best to handle personalized content, setting up an image caching server, using a cookie-free domain for static content, and using randomization in URLs for accurate hit metering or sensitive content.

An Introduction to XQuery by Jason Hunter

XQuery is a new language from the W3C that lets you query and manipulate XML -- or anything that can be represented as XML, such as relational databases. As a Java developer -- especially a server-side Java developer -- XQuery is key to searching and manipulating large XML repositories or performing any XML-centric task. This talk introduces XQuery. I'll explain the XQuery language; I'll show how to call XQuery from Java; and as the creator of JDOM, I'll also explain when to use XQuery instead of JDOM, and when to use both.

Java Metadata by Jason Hunter

Java's new Metadata facility introduced in J2SE 5.0 defines a way to attach decorations to classes, fields,

methods, and even packages that can be extracted by the compiler or runtime tools to provide advanced functionality. Think of metadata as an extended `@deprecated` flag, or think of `XDoclet++`. In this tutorial session you'll learn how Metadata fits in the Java platform (and how it compares to the C# platform). We'll cover how to use the metadata attributes provided in the core J2SE libraries and how to write your own. We'll also show a bit of what's coming in JSR-181, tasked to define standard metadata attributes for web services.

Spring Security with ACEGI by Justin Gehrtland

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.

Cryptography for Programmers by Justin Gehrtland

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.

Introduction to Hibernate by Justin Gehrtland

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. This session will introduce you to Hibernate. After an overview of common usage scenarios, including web and enterprise applications, we'll examine the basics of getting Hibernate running. We'll cover the mapping file format and syntax, including common relational mapping structures. Then, we'll examine the Hibernate API for interacting with the framework. Finally, we'll cover the common architectural decisions you'll have to make as you include this (or any other) O/RM framework.

Writing Secure Web Services (with Java and Axis) by Justin Gehrtland

Web Services are message-oriented. This means that any application intention (the need for security, for transactionality, for reliability, etc.) must be included in the message and not just assumed as external context. The WS-Security specifications are very advanced and currently being used in the wild to create robust, secure web services.

Applied Cryptography by Justin Gehrtland

Following directly on Cryptography for Programmers, Applied Crypto shows how the cryptographic primitives introduced in the previous session are combined into the identity, confidentiality and authorization systems we use today. We'll take a deep look at certificates, then examine their use in the Secure Socket Layer (SSL) protocol. From there, we'll examine the two competing dreams of distributed identity: PKI and Kerberos.

Spring MVC by Justin Gehrtland

The Spring team, as in all things they do, have learned the valuable lessons of the past when introducing a Spring solution. Spring MVC is everything Struts should be, and more besides.

Advanced Hibernate by Justin Gehrtland

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.

Lightweight POJO development with JBoss and EJB 3 by Michael Yuan

While EJB and J2EE are widely adopted standards in enterprise applications, they are also notorious for their complexity, excessive footprint, and low developer productivity. As enterprise Java evolve, many alternative frameworks have emerged, especially from the Open Source community, to promote simpler and POJO-based programming models for enterprise applications. They have met with great success. JBoss leverages the success of those Open Source projects and provide pre-integrated POJO services in a small footprint and modular container. In this presentation, I will discuss lightweight and POJO development options in the JBoss Application Server. The most important option is EJB 3.0, which standardizes many of the best ideas in existing Open Source POJO frameworks (e.g., POJO persistence, dependency injection, annotation-based configuration).

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.

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.

Design Pattern Modularization with AOP by Ramnivas Laddad

Design patterns # object oriented, concurrency control, and J2EE # all have certain crosscutting elements present. The obvious result of conventional implementation is unclear implementation that is tedious to implement and tough to change. Aspect-oriented programming (AOP) offers a way to simplify implementation of these design patterns. Further, AOP offers new design patterns of its own that allow for new ways of implementing functionalities.

Aspect-oriented refactoring: Taking refactoring to a new level by Ramnivas Laddad

Refactoring allows reorganizing code while preserving the external behavior, while AOP facilitates modularizing crosscutting concerns in a system through use of a new unit of modularity called aspect. Aspect-oriented refactoring synergistically combines these two techniques to refactor crosscutting elements. Individually, refactoring and AOP both share the high-level goal of creating systems that are easier to understand and maintain without requiring huge upfront design effort. A combination of the two # aspect-oriented refactoring # helps in reorganizing code corresponding to crosscutting concerns to further improve modularization that is easy to understand, highly consistent, and simple to change.

Advanced Groovy by Rod Cope

It's easy to start using Groovy, but there are lots of goodies that aren't so obvious in the beginning. In this session, we'll cover things like currying, single object iteration, dynamic language extensions, enhancing the JDK, default parameters, advanced closures, active proxies, and more.

Easy Enterprise Applications with JBoss, Hibernate, AspectJ, and XDoclet by Rod Cope

This session demonstrates how to make J2EE(TM) development faster and easier by integrating powerful Open Source tools to produce a rapid application development and deployment infrastructure. See how to harness the power of AOP (Aspect-Oriented Programming) in a practical EJB application to reduce clutter, improve readability, and remove drudgery.

Welcome to Hibernate: Object-Relational Mapping for Java by Rod Cope

Hibernate is an increasingly popular Open Source persistence framework for Java developers. It provides transparent persistence, works with all your favorite databases, and plays well both inside and outside of application servers. In this session, we'll learn the basics of Hibernate while exploring some mapping files and code.

Groovy = Java + Ruby + Python for the JVM by Rod Cope

Groovy is a new dynamic, object-oriented scripting language for the Java Virtual Machine. It has the expressive power of Ruby, the simplicity of Python, and can use all existing Java code. See how to build a Swing GUI, execute Ant scripts in-line, access a database, read and write XML, and more in a few lines of code.

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.

Unit Testing Java with Jython by Stuart Halloway

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

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.

Introduction to Java Reflection by Stuart Halloway

Reflection is writing code that manipulates itself. Well-written reflective code automates a broad class of repetitive, error-prone programming tasks. Poorly-written reflective code obfuscates programs and destroys the benefits of the type system. We'll focus on the former.

Working with Java Metadata by Ted Neward

As part of JDK 1.5, Java has introduced a facility for developers to create and use custom metadata annotations, as developed by the JSR 175 committee. This represents a radical new shift for the Java programming language, quite possibly larger and farther-reaching than generics or any other language feature.

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.

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?

The Fallacies of Enterprise Systems (Architecture) by Ted Neward

There's a set of fallacies that every enterprise developer has fallen for at some point in their enterprise development lives, and unless they've come to realize it early enough, all cause big trouble and painful learning experiences in the long run.

Effective Enterprise Architecture by Ted Neward

Bring all of your enterprise Java questions to this open forum discussion hosted by the author of #Effective Enterprise Java#, Ted Neward.

Test First Development by Venkat Subramaniam

Do you know that unit testing is more of an act of design than verification? What are its benefits? How do we write effective tests? How does unit testing relate to evolutionary design? How does it help you with refactoring? When should you write your tests? What are the types of tests you could write? These are some of the questions that you would ask if you are interested in Unit Testing. What is a better way to learn than practicing it? In this session the attendees will participate in designing and developing a small yet full application. Instead of PowerPoint slides, you will learn from example. The code you help develop will be available for free download on the speaker's web site.

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.

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.

Agile Software Development by Venkat Subramaniam

You have probably worked on a few projects that have succeeded and then a few that have failed. What were the factors that influenced the success or failure of those projects? You want to develop a system that is robust, maintainable, within budget, of high quality and with fewer defects. How can you realize those goals? What steps, process, tools you can use or follow to achieve this. In this session, the speaker will present a number of approaches that lead to successful development. He will also present his personal experience with those in implementing software projects. Attendees are encourage and expected to present their views on what has or has not worked for them.