

Twin Cities Software Symposium 2006

Marriott Minneapolis Airport Hotel - Bloomington, MN

March 17 - 19, 2006

(session listing as of 3/14/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Twin Cities Software Symposium coming to Bloomington on March 17 - 19, 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 Twin Cities Software Symposium will offer 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

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

Registration Fees

Attendees	Before 2/27/2006	After 2/27/2006
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: jjzimmerman@nofluffjuststuff.com or (303)469-0486.

Twin Cities Software Symposium 2006

Marriott Minneapolis Airport Hotel - Bloomington, MN

March 17 - 19, 2006

(session listing as of 3/14/2006)

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 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.

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.

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.

Applied Object-Oriented Metrics by Brian Sletten

Object-oriented code metrics are a little like Artificial Intelligence: those who did it twenty years ago roll their eyes at the thought and prophesy the same ultimate failure at applicability now. Those who grew up with Java are approaching the topic with new eyes and are finding useful ways of incorporating metrics into their projects. Come hear about tools and ways to measure properties of software, how they might be beneficial and where you are likely to go astray with this approach.

Applied Design Patterns by Brian Sletten

Just about every modern software developer has a copy of the Gang of Four "Design Patterns" book sitting on a shelf; many of them have actually read it. The dark secret of the patterns community is that there is often a large gulf between whiteboard simplicity and real-world complexity. Language choice plays a part in the design (and even importance) of patterns. The situation is made even more confusing by the fact that many of the core patterns have now been "voted off the island" for one reason or another. This talk will give a pragmatic overview of the motivations behind design patterns and will focus on applying a handful of the GOF patterns to example scenarios in Java, Ruby and C#. A quick introduction to the role AOP plays in changing the patterns landscape will also be covered.

Applied AOP by Brian Sletten

Most people new to Aspect-Oriented Programming (AOP) are fed up with separation of concerns zealots explaining how great their techniques are at dealing with... logging. Ok, you get it. Logging is a cross-cutting concern that can be appropriately modularized. What else does AOP have to offer? A lot, it turns out. This talk will give an introduction to the motivations of AOP as well as a series of concrete examples drawn from enterprise and client side Java. Come learn how AspectJ-flavored AOP can begin to benefit you immediately either in development or production environments. Learn how to enforce architectural policies, find Swing threading issues, reduce the invasiveness of the Observer design pattern or even improve the reusability of your domain models.

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. This session will address how agile interacts with enterprise concepts like SOA, cross team collaboration / program management, and tracking at the enterprise level # and more. Various tools and techniques will be discussed, and at least part of the session will include Q/A for the presenter to field specific questions about your organization.

Ajaxian Faces by David Geary

JavaServer Faces is a perfect platform for implementing Web 2.0 interfaces with Ajax. This session explores how you can use these two potent technologies--JSF and Ajax--together to create applications that look and behave like desktop applications but run in the browser.

JSF: State of the Art by David Geary

In 2005, JSF hit its stride, as evidenced from overwhelming support from both vendors and the open-source community. JSF 1.0 had plenty of holes, but open-source projects have arisen to address those needs. This session takes a look at three of those projects:

- Tomahawk (MyFaces component library)
- Facelets
- Seam

Hands-on Rails by David Geary

Come to this exciting preview of one of the leading web application framework contenders with the potential to be the Next Big Thing: Ruby on Rails. An innovative framework with an eye-popping array of ultra-cool features such as active record and native support for Ajax, Rails greatly simplifies web application development and puts the joy back in software development. Rails is easy, fun, and very productive; in fact, in the throes of Rails-mania, some converts have claimed that developing with Rails is at least 10 times as fast as your favorite Java framework. Could that be? Come see for yourself.

JavaServer Faces: A Whirlwind Tour by David Geary

JavaServer Faces (JSF) has arrived. The standard Java-based web application framework based on Struts, JSF really took off in 2005. Embraced by developers, vendors, and open-source projects, JSF has started to hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

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. This session will discuss some of the most common difficulties for adopting agile and provide various plans of attack. The session will start with a listing of issues for the session participants, and some portion of the session will be dedicated to an open forum where the presenter will address the issues collected.

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

As agile grows, so too do the questions for how to track and communicate progress within the project community as well as to upper management and others interested in progress. This session will focus on tools and techniques for tracking an agile project plan from creation to project completion. We will create a simple plan in a planning tool, and run a mock project, showing how the plan addresses: communicating progress, addressing missed estimates, scope modifications, and more.

Creating, Telling, and Tracking User Stories by David Hussman

The questions around user stories are many, and the list is only growing larger as their popularity of increases. Many organizations are on their path to adopting stories as requirements vehicles, possibly struggling with story writing as well as finding a way to fit them into their organization. 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. We will also discuss how to connect to product owners outside the project community and briefly review several tools for tracking and managing user stories.

Automating Business Value with FIT and FitNesse by David Hussman

Agile communities consider stories #done# when the acceptance tests (also called story tests) are shown to the customer. Originally, this was a manual process, but in recent years, several frameworks have been created to automate this process, providing acceptance testing all the benefits of automated unit testing. One of the most popular of these is called FIT, created by Ward Cunningham. 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.

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: Turbo-charge your JSF Apps 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.

Java Open Source Development Survey by Jim White

Considering open source development? Chances are pretty good that even if you have not fully embraced the open source philosophy, you are at least using one open source product. There might be other tools, frameworks or APIs that you would use if you knew a little more about them. The fact is, Java open source development has exploded in recent years. There is an open source product to help you gather requirements, to test and deploy your application and everything in between.

Design Guidelines for Large Java Message-based Enterprise Architecture Integration Systems by Jim White

Designing very large message-based EAI solutions is unlike developing Web or other distributed applications. It requires designers, architects and implementers to consider issues of timing, distributed process control, failure and volume that they are probably unaccustomed to addressing.

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.

Introduction to Hibernate by Justin Gehtland

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.

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.

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.

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.

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 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.

JMX and System Health by Kevin Pumarlo

Systems today are built with multiple components spread across many platforms and diverse technologies. The major focus of most development efforts is to #get it built # get it deployed#, after which most development teams move on. But what of the system? It's still there, out in the wilds of production, with the support team unaware of it # until it breaks. What if a system could communicate the state of its health, proactively working to help those supporting the application? JMX technology provides the mechanisms to see into the workings of a system and enable reporting of its vital statistics to the outside world, and more importantly, provide insights into things to come.

GOF Patterns Applied by Kirk Knoernschild

Design Patterns are proven and powerful techniques that can help improve the resiliency, maintainability, and extensibility of your applications. However, overusing or misapplying patterns is a common mistake often times resulting in applications that are over-architected, and resemble a tangled web of classes. How can patterns be applied to achieve the goal of better software?

Dependency Management Techniques by Kirk Knoernschild

Why is software so difficult to change? When you establish your initial vision for the software's design and architecture, you imagine a system that is easy to modify, extend, and maintain. Unfortunately, as time passes, changes trickle in that exercise your design in unexpected ways. Unlike what you had anticipated, each change begins to resemble nothing more than another hack, until finally the system becomes a tangled web of code that few developers care to venture through. Eventually, modifications to the software intended to improve the system have the opposite affect of breaking other parts of the system. The software is beginning to rot.

From Code to Architecture by Kirk Knoernschild

The code we write has a tremendous impact on our software architecture. In fact, the code is our architecture. Not all of it, of course, but a good share of it. The problem is that we often don't fully comprehend the architectural impact of our code at the time we create it. One poorly designed class or method can severely impact the resiliency, stability, extensibility, and maintainability of your software. There are ways we can create flexible architectures...but we have to emphasize loose coupling among system components.

Benefits of the Build - A Case Study in Continuous Integration by Kirk Knoernschild

Agile processes such as XP and RUP advocate continuous integration, where shorter iterations produce an incremental and functional growth of the system. The fundamental component of any Continuous Integration strategy is an automated and repeatable build. In addition to ensuring your application is always in a functional state, a robust build strategy enables a number of other important lifecycle activities.

The Productive Programmer by Neal Ford

This session shows you how to become a more productive programmer every day by using tools that you didn't know you already had.

Introduction to Tapestry by Neal Ford

This session delves into details about building web applications with Tapestry, covering configuration, templates, and separation of concerns.

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.

Pragmatic Extreme Programming Part 2: Architecture, Coding, and Testing by Neal Ford

Continues the discussion from Part 1, focusing on how to keep the benefits of XP without sacrificing its effectiveness. This session shows in-depth samples of XP in action.

Advanced Enterprise Debugging Techniques by Neal Ford

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

Pragmatic Extreme Programming Part 1: Planning & Design by Neal Ford

This session begins a detailed discussion about how to actually get XP done in the real world (and what to tell your boss). This session includes artifacts (project tracking sheets, code coverage reports, etc.) from real XP projects.

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

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

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.

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.

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.

Testing Strategies for Web Applications by Ramnivas Laddad

Ever wondered if you can automate testing of your web application, but couldn't produce a satisfactory solution? If so, this is the session for you! Attend this session to understand the alternatives you have for unit and functional testing of web 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.

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.

Better Software Through Refactoring by Rob Sanheim

Refactoring is the process of improving the design of code without changing the behavior. The practice of refactoring is increasingly becoming a required skill for any professional developer, particularly as agile methodologies and incremental development spread. This presentation will bring you up to speed in refactoring for Java but will be applicable for any OO language.

Effective AJAX by Rob Sanheim

You have probably heard all the hype surrounding Ajax by now. But how do you actually go about implementing the next Google Maps or Basecamp? In this presentation we'll learn the several open source frameworks that make developing Ajax easier, faster, and more fun.

J2EE Web Services @ Work by Tom Marrs

Have you tried to deploy J2EE Web Services and thrown up your hands in frustration at the lack of tool support? Have you been confused by the tangled web of new deployment descriptors? Do you want to know how to develop and deploy J2EE-compliant Web Services so that it works every time? If so, then this talk is for you. at the lack of tool support? If so, then this talk is for you.

J2EE Security @ Work: J2EE Meets JAAS by Tom Marrs

Have you wasted time writing lots of security-based code and ever wondered if there's a better way to add security to your application? Are you confused by declarative security? Have you read about JAAS (Java Authentication and Authorization Service) but wondered where it fits? Have you ever said, "Can I just see a working example"? If so, then this talk is for you. Have you ever said, "Can I just see a working example"? If so, then this talk is for you.

Java/J2EE Architecture @ Work: EJB 3 vs Spring and Hibernate by Tom Marrs

You've used EJB in the past and been disappointed - it was too heavy and difficult to use. Like Bruce Tate, maybe you've gone from "Bitter" to "Better, Faster, Lighter". With EJB 3 shipping in early 2006, maybe it's time to take another look. We'll compare EJB 3 with alternative frameworks - Spring and Hibernate - to see if EJB 3 has closed the gap.

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.

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.

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.

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?