

New England Software Symposium 2006

Sheraton Ferncroft - Danvers, MA

March 10 - 12, 2006

(session listing as of 3/7/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the New England Software Symposium coming to Danvers on March 10 - 12, 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 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 New England 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) ServerSide Java
- 3) Core Java
- 4) Architecture
- 5) XML / Web Services

Registration Fees

Attendees	Before 2/20/2006	After 2/20/2006
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) 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.

New England Software Symposium 2006

Sheraton Ferncroft - Danvers, MA

March 10 - 12, 2006

(session listing as of 3/7/2006)

Improving Java code quality with code auditing tools by Brian Goetz

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Fortunately, there are some new code auditing tools that can help spot some bugs missed by other approaches.

Garbage Collection in the HotSpot JVM by Brian Goetz

Pop quiz: which is faster, Java or C++? If you are talking about allocation performance, the answer is Java, hands-down.

The Java Memory Model by Brian Goetz

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

Structuring concurrent applications in JDK 5.0 by Brian Goetz

JDK 5.0 is a huge step forward in developing concurrent Java classes and applications, providing a rich set of high-level concurrency building blocks.

Java Performance Myths by Brian Goetz

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

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...)

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.

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?

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.

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.

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.

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.

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.

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.

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 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.

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

#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.

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.

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.

Advanced JavaScript Part 2: Objects, Tools, and Techniques by Glenn Vanderburg

Building on the fundamentals from part 1, this talk covers JavaScript's strange but powerful object model, shows some useful JavaScript tools, and shows how to use JavaScript's power and steer clear of its weaknesses.

Modern Project Infrastructures by Glenn Vanderburg

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

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.

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.

Advanced JavaScript, Part 1: JavaScript Foundations 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. Part 1 focuses on JavaScript's procedural and functional features; the object model is covered in part 2.

Tapestry In Action by Howard Lewis Ship

An introduction to the Jakarta Tapestry web application framework, which will explain the concepts and features of the framework while live coding simple applications. Tapestry forms, request cycle, component object model, and the use of several important components (including the powerful Table data grid) will be featured.

Creating Tapestry Components by Howard Lewis Ship

In Tapestry, components are not an add-on; in fact, anything but! Tapestry components are integral to the entire framework # if something dynamic is going on in a page, there's a component involved.

Pragmatic Unit Testing with TestNG and EasyMock by Howard Lewis Ship

You've heard about unit testing but were daunted when it came time to put the pedal to the metal. That's because JUnit is just one tool and there's others you need to learn about, including the wonderful and wierd EasyMock and the easy and powerful TestNG.

Code Coverage: A Guardian of Quality by Ian Roughley

Code coverage is generally viewed as a metrics that managers use to chart progress, a number that has to be blindly attained. In this talk we discuss everything that you, the developer, need to know to make it more than a number and part of a process that will improve code quality.

Evaluating Open Source Solutions by Ian Roughley

Many companies and most, if not all, software today utilizes open source. Whether it is databases, application servers, frameworks or libraries, these projects are fast becoming a standard commodity for building business-related functionality upon and speeding up development time. Sometimes technology evaluations are done, but frequently the library is simply slipped into the code base to address an urgent requirement - often without evaluating the technology beyond the immediate need.

Improving Developer Productivity with WebWork 2.2/Struts Action Framework 2.0 by Ian Roughley

WebWork and Struts Action Framework 2.0 are merging to become the next action based framework from Apache. Learn what this means to you as a developer, and how to use the many features of WebWork.

Software Tools That Make Life Easier: Part One by Jared Richardson

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

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.

Software Tools That Make Life Easier: Part Two 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.

EJB 3.0 and New Java Persistence API by Mark Richards

The new EJB 3.0 spec (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will take a look at the new EJB 3.0 spec and the new Java Persistence API. Included in this session will be a discussion about Java metadata annotations, simplification of enterprise beans (session and message-driven beans), interceptors, changes in transaction processing, and how the new Java Persistence API works. During the session I will be demonstrating how the EJB 3.0 spec differs from the EJB 2.1 spec through code example comparisons. I will also be discussing how the new Java Persistence API compares to related Java persistence options and whether we should be excited about the new persistence API or (yawn) sticking with what we have.

J2EE Command Pattern Architecture by Mark Richards

Using 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. In this session we will explore the main issues facing a typical EJB architecture and learn how the use of the Command Pattern can increase testability, reduce complexity, increase performance, and increase maintainability of our J2EE applications. Through interactive coding you will learn what components make up the Command Pattern framework and what simple coding changes are required to convert a complex EJB-based application to a remotable POJO-based application using the J2EE Command Pattern.

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!

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.

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.

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.

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.

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."

Easing into Agile by Scott Davis

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

Real World Web Services by Scott Davis

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

Holistic Testing by Scott Davis

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

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.

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.

Advanced Hibernate by Stuart Halloway

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

Spring AOP by Stuart Halloway

Learn to use Spring AOP, aspect injection. and AspectJ integration

Spring Dependency Injection by Stuart Halloway

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.

Spring Fundamentals by Stuart Halloway

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.

Pragmatic AJAX (Intro) by Stuart Halloway

Web applications have traditionally been a sort of Faustian bargain, yielding the high-quality user experience that desktop applications can deliver in exchange for incredibly easy deployment and lower support costs. One of the arguments against creating rich HTML/JavaScript has been how difficult they can be to write and maintain. You will see how to adapt modern Java web frameworks such as JavaServer Faces and Tapestry to ease both the creation and maintenance of these types of applications. With Ajax we can get a lot of the best of both worlds. Now our web applications can be powerful, usable, and easily deployed! Web applications have traditionally been a sort of Faustian bargain, yielding the high-quality user experience that desktop applications can deliver in exchange for incredibly easy deployment and lower support costs (i.e., no client software to install, upgrade, and test on every conceivable desktop configuration).

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 significance 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 based on our specific needs.

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.

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?

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.

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 brings 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.

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.