

New England Software Symposium

Sheraton Ferncroft

March 09 - 11, 2007

<http://www.nofluffjuststuff.com/sh/2007-03-boston>

Fri, Mar. 09, 2007					
	North Shore B&C	North Shore A	Newburyport	Gloucester	Ipswich
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Java 6 Features, what's in it for you? Venkat Subramaniam	Enterprise JavaScript Greg Murray	10 Ways to Improve Your Code Neal Ford	Mocking Web Services Scott Davis	Gradual Agile: The Secret to Introducing Agile Practices Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Annotation Hammer Venkat Subramaniam	Enterprise Ajax with Java Greg Murray	Implementing SOA Neal Ford	The Zen of REST Scott Davis	Agile Software Testing Strategies Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Domain Driven Design Venkat Subramaniam	Project jMaki - Enabling Web 2.0 Application Developers Greg Murray	Power Regular Expressions in Java Neal Ford	The Enterprise Service Bus: Do We Really Need It? Mark Richards	Shippers Unite! Jared Richardson
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, Mar. 10, 2007					
	North Shore B&C	North Shore A	Newburyport	Gloucester	Ipswich
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Java Performance Myths Brian Goetz	Spring 2.0: New and Noteworthy Mark Fisher	Groovy for Java Programmers Venkat Subramaniam	Debugging and Testing the Web Tier Neal Ford	Distributed Teams: Remote Agility Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	The Java Memory Model Brian Goetz	Introduction to Spring Security Mark Fisher	Agile Web Development with Grails Venkat Subramaniam	Advanced Selenium Neal Ford	Build Teams, Not Products Jared Richardson
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford	Spring into Groovy Venkat Subramaniam	Using Aspects to Work with Annotations Ron Bodkin	Structuring concurrent applications in JDK 5.0 Brian Goetz	Continuous Integration with Cruise Control Jared Richardson
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Productive Programmer: Automation and Canonicity Neal Ford	Spring MVC Essentials Mark Fisher	The Busy Developer's Guide to Rules and Rules Engines Using JESS Ted Neward	Effective Concurrent Java Brian Goetz	Making Architecture Work Through Agility Mark Richards
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Mar. 11, 2007					
	North Shore B&C	North Shore A	Newburyport	Gloucester	Ipswich
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Groovy and Java: The Integration Story Scott Davis	Intro to Java Persistence API (JPA) Mark Richards	The Busy Java Developer's Guide to Reflection Ted Neward	OSGi: A Well Kept Secret Venkat Subramaniam	Squashing bugs with FindBugs Brian Goetz
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Ajax development with the Yahoo! UI Library and Grails Scott Davis	Advanced Java Persistence API (JPA) Mark Richards	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Drooling with Groovy and Rules Venkat Subramaniam	Effective AOP Ron Bodkin
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	BIRDS OF A FEATHER SESSIONS				
2:15 - 3:45 PM	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Message Driven POJOs Mark Fisher	EJB3 Core Specification (JSR-220) Mark Richards	AJAX Performance and Scalability Ron Bodkin	Continuous Integration Paul Duvall
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	The Role of Spring in an ESB Mark Fisher	Making The Right Persistence Framework Choice Mark Richards	Glassbox: Open Source Monitoring and Troubleshooting Ron Bodkin	Improving Code Quality Using Maven 2 Paul Duvall

New England Software Symposium

Sheraton Ferncroft

March 09 - 11, 2007

<http://www.nofluffjuststuff.com/sh/2007-03-boston>

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.

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.

Effective Concurrent Java by Brian Goetz

The Java programming language has turned a generation of applications programmers into concurrent programmers through its direct support of multithreading. However, the Java concurrency primitives are just that: primitive. From them you can build many concurrency utilities, but doing so takes great care as concurrent programming poses many traps for the unwary.

Squashing bugs with FindBugs 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.

Enterprise JavaScript by Greg Murray

The popularity of Ajax has given us many JavaScript toolkits over the past few years, each of which have their strengths and weaknesses. This talk will include a discussion of some of the more popular of these, such as Dojo, JQuery, Prototype, Spry, Script.aculo.us, and the Yahoo User Interface (YUI) library. In addition, the presentation will also discuss what you need to know about JavaScript to develop enterprise-grade applications, including topics such as emulating namespaces in JavaScript, object creation and inheritance using prototypes, associative arrays, closures, internationalization, and Ajax interaction techniques.

Enterprise Ajax with Java by Greg Murray

Ajax and Java are the perfect combination for creating Web 2.0 applications. This session will many of the key issues that Java developers may experience when developing Ajax clients and services. Topics that we will cover various topics including security, performance, usability, debugging, and working with JavaScript Object Notation (JSON). This session will conclude with a discussion of both Java-centric and JavaScript-centric approaches (i.e, light client-side logic versus heavy client-side logic) along with the tradeoffs to consider with both of these approaches.

Project jMaki - Enabling Web 2.0 Application Developers by Greg Murray

Creating a breathtaking user interface requires that you and your team are skilled in JavaScript, Cascading Style Sheets (CSS), and Ajax interactions. This session will discuss how you can use Project jMaki (or just the ideas behind it) to create JavaScript-centric user interfaces that are developer and designer friendly. Project jMaki is an open-source client/server framework hosted on java.net that is focused on creating JavaScript-centric web applications. jMaki promotes a clean separation of JavaScript, CSS, and template code following the web design approach widely known as "unobtrusive JavaScript?". This session will include a technical discussion of the architecture and features of jMaki including JavaScript toolkit interoperability, widget design, using layouts, wiring JavaScript functionality together, and service/data integration.

Gradual Agile: The Secret to Introducing Agile Practices by Jared Richardson

Agile practices are popular because they work, but getting people to take that first step can be tricky.

Agile Software Testing Strategies by Jared Richardson

Creating and maintaining a solid automated test suite is critical to an Agile strategy, but often we're just told to "Do it." In this talk we'll look at several pragmatic strategies for creating and building your suite.

Shippers Unite! by Jared Richardson

An overview of the Agile software approach from the book Ship It! A Practical Guide to Successful Software Projects.

Distributed Teams: Remote Agility by Jared Richardson

How do you keep a team scattered across time zones in sync?

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

Continuous Integration with Cruise Control by Jared Richardson

Continuous Integration is increasingly recognized as a vital practice in an Agile software shop. Traditionally it's been difficult to set up and administer. Today, that's no longer the case.

Spring 2.0: New and Noteworthy by Mark Fisher

Spring 2.0 has marked a major advance in the Spring Framework. While still maintaining backwards compatibility, this release adds quite a few new features. What are those features and how do they add value? Come by and see. In this session we'll provide a practical tour of what's new in Spring 2.0. Spring 1.x users who are looking to upgrade to Spring 2.0 will love this session. If you're not using Spring already, this talk will give a great overview of the things you're missing out by not using Spring 2.0.

Introduction to Spring Security by Mark Fisher

Spring Security (a.k.a. Acegi) enables self-contained, consistent solutions for securing your applications. The interceptor-based approach is non-invasive even when extended to accommodate domain-specific requirements. The two main security processes (authentication and authorization) are decoupled in order to provide flexibility across a wide variety of providers and strategies.

Spring MVC Essentials by Mark Fisher

Spring MVC is a powerful and flexible framework for building web applications. Its interface-based API promotes loose coupling, yet it also offers many convenient base classes for common functionality. Whether you use the base classes or roll your own, you will discover numerous strategies and extension points. This flexibility will be greatly appreciated once you are up to speed, but newcomers often wonder where to begin.

Message Driven POJOs by Mark Fisher

One of the most exciting new features of Spring 2.0 is its support for Message-Driven POJOs. It is now possible to receive JMS messages asynchronously and delegate the handling of those messages to simple objects. If your POJO has a return value, it will automatically be sent to a reply destination. Spring's messaging containers support configurable pooling of concurrent consumers and offer full integration with Spring's transaction management.

The Role of Spring in an ESB by Mark Fisher

An Enterprise Service Bus (ESB) brings flow-related concepts such as transformation and routing to a Service-Oriented Architecture. An ESB can also provide an abstraction for endpoints. This promotes flexibility in the transport layer and enables pluggability of POJO services.

The Enterprise Service Bus: Do We Really Need It? by Mark Richards

There has been a significant amount of buzz in the community and industry about the definition and role of an Enterprise Service Bus (ESB), particularly within the area of Service-Oriented Architecture (SOA). In this product-agnostic high energy session we will take a step back and consider whether we really need an ESB. Through real-world application and architecture scenarios we will see where an ESB would be helpful and where it would be overkill. Then, using product-agnostic coding examples we will learn what an Enterprise Service Bus is supposed to do, then answer the question about whether the ESB is just a bunch of hype or if we really need it.

Making Architecture Work Through Agility by Mark Richards

As companies continue to change the way they do business, so must the IT systems that support the business. Changes due to regulatory requirements, competitive advantage, mergers, acquisitions, and

industry trends require flexible IT systems to meet the demands of the business. Software Architects must therefore make their architectures more agile to meet the flexible demands of today's business. Through real-world examples and scenarios we will explore some of the challenges facing Software Architecture and discuss several concrete techniques for applying agility to both the architecture process and the technical architecture itself. We will also look at various architecture refactoring techniques, and discuss the pros and cons of each. By attending this session you will learn how to apply various agile techniques to improve your architectures and overcome some of the challenges facing software architecture in today's ever-changing market.

Intro to Java Persistence API (JPA) by Mark Richards

In addition to providing a simplified API, the new EJB3 specification (JSR-220) defines a standard ORM Java Persistence API (JPA) that is rapidly gaining in popularity. As you will see in this session, JPA bears a striking resemblance to popular ORM solutions like Hibernate and Toplink. In this session we will explore in detail the new Java Persistence API offered by JSR-220. We will start by discussing the overall design and architecture of the JPA and how the major components within JPA interact. We will then look at defining mapping objects (entities) and how to use the EntityManager to manage these entities. Through interactive coding examples we will investigate the pros and cons of detached entities and merging, and finally see how to use XML mappings rather than annotations. More advanced features of JPA will be covered in a separate session.

Advanced Java Persistence API (JPA) by Mark Richards

This session picks up where the Intro to JPA session left off and covers some of the more advanced topics in the Java Persistence API. Some of the topics covered in this session include entity relationships (1-1, 1-n, n-n), compound keys, using multiple persistence providers, switching persistence providers, lazy loading, managing detached entities, and the JPA Query Language (JPQL). Some knowledge of JPA is recommended for this session as I will not be covering the basics of JPA (that is covered in a separate Intro to JPA session). Through a combination of slides and interactive coding I will demonstrate these advanced topics using both Hibernate and Toplink JPA.

EJB3 Core Specification (JSR-220) by Mark Richards

The new EJB 3 specification (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will discuss why EJB is still important, and explore in detail some of the new features of the core EJB 3 specification. Included in this session will be defining and accessing session beans, JTA transaction management, declarative security, and interceptors. For those of you who still like to code in XML, I will also discuss and show how we can use XML rather than annotations within EJB3. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. Note: this session does not cover the new Java Persistence API (JPA) - that topic is covered in separate JPA sessions.

Making The Right Persistence Framework Choice by Mark Richards

Java Persistence has come a long way in the past 4 years. We have many viable options available now, including JPA, Hibernate, iBATIS, Toplink, and OpenJPA. With so many options available now it is difficult to know when to use which framework. In this session we will focus on native Hibernate, JPA, and the iBATIS framework, and discuss the main strengths and weaknesses of each approach and what the decision criteria is for using each of these frameworks. Knowing that it is not a one-size-fits-all situation when it comes to Java Persistence, through interactive coding we will take a look at how to use iBATIS together with native Hibernate or JPA and when this makes sense. By attending this session you will gain the knowledge necessary to make informed decisions about which Java Persistence Framework to choose for your current or next Java-based application.

10 Ways to Improve Your Code by Neal Ford

No one writes perfect code, and every developer eventually falls into a slump where they just crank out the same code day after day. This session illustrates 10 different ways to improve your code, covering sacred cows, good citizens, smells, and more.

Implementing SOA by Neal Ford

This talk avoids SOA hype and gets to the meat of the matter: how do you implement a Service-Oriented Architecture, what are the technological pitfalls, how do you test it, and what traps should you avoid. No marketecture: just implementation details.

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.

Keynote: Polyglot Programming by Neal Ford

Java is dead; long live Java!

Debugging and Testing the Web Tier by Neal Ford

As our applications have spilled from the server across the wire to the web tier, we increasingly must debug and test in the browser. This session covers debugging and testing tools for clients, JavaScript, and Ajax.

Advanced Selenium by Neal Ford

This session discusses advanced Selenium techniques for testing web applications. It discusses techniques for both TestRunner and Remote Control Selenium, including data driven tests, creating branch points, testing Ajax applications, creating flexible tests, integration with continuous integration, and tons more.

Productive Programmer: Acceleration, Focus, and Indirection by Neal Ford

This session discusses how to use the Productive Programmer principles of acceleration, focus, and indirection to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

Productive Programmer: Automation and Canonicity by Neal Ford

This session discusses how to use the Productive Programmer principles of automation and canonicity to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

Continuous Integration by Paul Duvall

Increase feedback on your project by building your software with every change applied to your source code repository. The practice of Continuous Integration (CI) can be used to decrease the time between when a defect is introduced and when it is fixed.

Improving Code Quality Using Maven 2 by Paul Duvall

Learn how to use Maven 2 while using plugins that help analyze your code using measures such as cyclomatic complexity, code duplication, code coverage, coding standards and dependency analysis

Using Aspects to Work with Annotations by Ron Bodkin

In this session, you will learn how to use Aspect-Oriented Programming (AOP) as a tool to avoid annotation hell by working effectively with Java 5 annotations (such as @Remote). You will see simple and more advanced techniques to process custom annotations in a higher-level Java-like language, and how this compares to lower-level approaches like the Java Annotation Processing Tool. You will also see techniques for simplifying annotations, by providing application-specific default values and by deriving standard annotations used by frameworks like EJB 3, JAX-WS, and the Spring Framework from higher-level domain-specific annotations using AOP.

Effective AOP by Ron Bodkin

Aspect-Oriented Programming (AOP) allows better application architectures by centralizing scattered and tangled code for problems like security, error handling, testing, policy enforcement and feature variations across a product line. The recent releases of Spring 2.0 and AspectJ 5 illustrate the maturing options for Java projects.

AJAX Performance and Scalability by Ron Bodkin

You know AJAX can improve Web application usability, but only if designed properly. How do you deploy and manage responsive rich client networked applications that scale? If you are using services from multiple sources, how can you identify bottlenecks?

Glassbox: Open Source Monitoring and Troubleshooting by Ron Bodkin

In this session, you will learn how the Glassbox open source troubleshooting and monitoring agent supports low overhead monitoring and troubleshooting without needing to "bake in" instrumentation up front. Glassbox provides an easy to use AJAX interface, an automated installer, and concise summaries of common problems such as database failures, and slow operations caused by thread contention and excessive distributed calls. Glassbox also supports customization and detailed analysis for deeper investigation. Under

the covers, Glassbox uses JMX and aspect-oriented programming to discover applications, track performance, and automatically diagnose common problems in Java applications. You will see how Glassbox can be extended easily with XML, AspectJ, and Spring AOP, providing a useful foundation for customized application monitoring. See also <http://www.glassbox.com/> for more information.

Atom: From Blogging to Data Syndication by Scott Davis

You've probably read a blog or two, but have you ever looked at the underlying protocol that makes the blogosphere work? RSS kick-started the phenomenon, but Atom is the IETF specification that codifies it. Atom as a blogging dialect is interesting, but Atom as a data syndication format is something that is on the rise. Google recently deprecated its SOAP API and is aggressively moving towards Atom as its preferred way to interact with its services.

Mocking 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 can be easily mocked up for testing purposes and to aid offline development. You'll see working examples of RESTful, SOAP, and JSON web services, as well as strategies for unit and functional testing your asynchronous, service-oriented architecture.

The Zen of REST by Scott Davis

Google quietly deprecated their SOAP search API at the end of 2006. While this doesn't mean that you should abandon SOAP, it does reflect a growing trend towards simpler dialects of web services. Google joins a number of popular websites (Yahoo, Flickr, YouTube, del.icio.us) that offer all of the benefits of web services without all of the complexity of SOAP.

Groovy and Java: The Integration Story by Scott Davis

I'm attracted to Groovy because of its spirit of inclusiveness. Because it extends my platform of choice, not replaces it -- include a single JAR in your classpath and you are Groovy-enabled. Because it offers full bidirectional integration with Java. Because it offers a nearly flat learning curve for experienced Java developers. Come see how you can use Groovy to augment your existing Java codebase.

Ajax development with the Yahoo! UI Library and Grails by Scott Davis

Yahoo! is a company that eats its own dog food. They open sourced the Ajax code that drives many of their own websites, including their eponymous homepage, Yahoo! Mail, and Yahoo! News. Come see first hand how the various pieces of the library work together as a seamless whole.

The Busy Developer's Guide to Rules and Rules Engines Using JESS by Ted Neward

If you've been keeping your ear to the ground, you may have heard some talk recently about "rules", "business rules" and "rules engines", but not necessarily any clear discussion on what they are, how to use or design them, or why they might be useful or important.

The Busy Java Developer's Guide to Reflection by Ted Neward

If you've never used Reflection (`java.lang.reflect`), you don't know what you're missing. In this presentation, we'll take a code-first, soup-to-nuts look at the Java Reflection APIs, from how to examine the class metadata that Reflection provides, to using annotations to enhance that metadata with your own information, even through the use of Java Dynamic Proxies to create flexible object "interceptors" that can layer services in front of ordinary method calls with nothing more complicated than an interface and a factory.

The Busy Java Developer's Guide to ClassLoaders by Ted Neward

If you've ever gotten a `ClassCastException` and just knew the runtime was wrong about it, or found yourself copying `.jar` files all over your production server just to get your code to run, then you probably find the Java `ClassLoader` mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--`ClassLoaders` aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose `ClassLoader` problems. And once you've got that, and hear about `ClassLoaders`' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like `ClassLoaders` after all... maybe.

The Busy Java Developer's Guide to Java Platform Security by Ted Neward

Permissions, policy, `SecurityExceptions`, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic,

code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

The Busy Java Developer's Guide to Debugging and Monitoring by Ted Neward

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to make use of the wealth of tools that the Java Standard Platform makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

Java 6 Features, what's in it for you? by Venkat Subramaniam

What benefit do new Java 6 features offer you. Are there issues with using these features. The objective of this presentation is not simply to introduce you to the features, but to the effective use of these as well.

Annotation Hammer by Venkat Subramaniam

Annotation is an interesting feature in Java. However, like any features, there are good uses and bad uses. When should you use Annotation? This presentation will answer that question for you.

Domain Driven Design by Venkat Subramaniam

Domain Driven Design (DDD) is an approach that places emphasis on the domain model and carrying it into implementation. DDD is mostly repackaging of fundamental OO Design. It brings new emphasis to what we should be already doing, but often find it hard and confusing given the realities and complexities of our real world. In this presentation we will take a close look at what DDD is and how to use it for agile development. We will discuss several design options, and also look at some examples of good modeling and layering.

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.

Agile Web Development with Grails by Venkat Subramaniam

Agile development is all about developing code and seeking feedback from your users to make sure you're developing what's relevant. When they suggest changes, those must be affordable and reliable. Grails, along with its facility to develop test driven, is a killer combination for rapidly developing web applications. In this ZePo (Zero PowerPoint) presentation, we will take a test driven approach to developing a small but fully functional web application in Grails. We will cover the fundamental features of Grails along with utilizing other capabilities like Ajax. At the end of this presentation, you not only be confident, but eager to roll your own web application using Grails.

Spring into Groovy by Venkat Subramaniam

What do you get when you mix an agile, object-oriented, dynamic language with a lightweight, flexible, and extensible framework? You get a Groovier Spring. Spring allows you to develop using Groovy as much as Java. Groovy brings some neat concepts to the Java Platform that is hard to realize directly through the Java language. Using these capabilities can lead to elegant and easier Spring development.

OSGi: A Well Kept Secret by Venkat Subramaniam

In this presentation we will introduce OSGi and discuss how it can help modularize and version your enterprise Java applications.

Drooling with Groovy and Rules 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.