

New England Software Symposium

Sheraton Framingham

September 14 - 16, 2007

<http://www.nofluffjuststuff.com/sh/2007-09-boston>

(event schedule as of September 12, 2007)

Fri, Sep. 14, 2007					
	Grand South	Ground Centre	Witherspoons	Sudbury	Commons
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Groovy: Greasing the Wheels of Java Scott Davis	JavaServer Faces: A Whirlwind Tour David Geary	Agile Project Management with Mingle Neal Ford	Annotation Hammer Venkat Subramaniam	The Busy Java Developer's Guide to java.util.concurrent Ted Neward
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Groovy and Java: The Integration Story Scott Davis	Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary	Metrics-driven Agile Development Neal Ford	OSGi: A Well Kept Secret Venkat Subramaniam	The Busy Java Developer's Guide to Threads and Synchronization Ted Neward
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Real World Grails Scott Davis	Ajaxian Faces David Geary	Programming with Mock objects Venkat Subramaniam	10 Ways to Improve Your Code Neal Ford	Enterprise Performance and Scalability Ted Neward
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Scott Davis				

Sat, Sep. 15, 2007					
	Grand South	Ground Centre	Witherspoons	Grand North	Sudbury
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	The Zen of REST Scott Davis	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Distributed Teams: Remote Agility Jared Richardson	Enterprise Applications with Spring: Part 1 Ramnivas Laddad	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Java6: Exploring Mustang Ted Neward	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Shippers Unite! Jared Richardson	Enterprise Applications with Spring: Part 2 Ramnivas Laddad	Productive Programmer: Automation and Canonicity Neal Ford
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	The Busy Java Developer's Guide to ClassLoaders Ted Neward	The Google Web Toolkit, Part One David Geary	Software Development Techniques Jared Richardson	Domain Driven Design: Simplifying Enterprise Architecture Ramnivas Laddad	Introduction to JRuby Neal Ford
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Monitoring Software Quality with Continuous Integration Andrew Glover	The Google Web Toolkit, Part Two David Geary	get Fit Venkat Subramaniam	Leveraging annotations with AOP Ramnivas Laddad	Rails for JRuby Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Sep. 16, 2007					
	Grand South	Ground Centre	Witherspoons	Grand North	Sudbury
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Groovy 101: putting Groovy to use quickly Andrew Glover	Intro to Java Persistence API (JPA) Mark Richards	Software Development Risk Analysis techniques Mark Johnson	Open Source SOA Burr Sutter	Internationalization and Localization in Java David Bock
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Getting Started with Grails Jason Rudolph	Advanced Java Persistence API (JPA) Mark Richards	Requirements Drive Design and Development (RDDD) Mark Johnson	JBoss ESB Deep Dive Burr Sutter	Adding Behavior to Java Annotations John Heintz
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Advanced Domain Models in Grails: Enterprise Integration Made Easy Jason Rudolph	EJB3 Core Specification (JSR-220) Mark Richards	Introducing Agility to Large Organizations David Bock	REST: The basics, and not so basic... John Heintz	Message Driven POJOs Mark Fisher
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense Andrew Glover	Making The Right Persistence Framework Choice Mark Richards	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Glassbox: Open Source Java Monitoring and Troubleshooting John Heintz	Introduction to Spring Security Mark Fisher

New England Software Symposium

Sheraton Framingham

September 14 - 16, 2007

<http://www.nofluffjuststuff.com/sh/2007-09-boston>
(event schedule as of September 12, 2007)

Behavior-driven development in Java by Andrew Glover

Behavior-driven development, or BDD, has attracted a lot of attention a la RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework. In this session, we'll look at what BDD is and how it shifts the traditional testing vocabulary from being test-based to behavior-based.

Monitoring Software Quality with Continuous Integration by Andrew Glover

The practice of continuous integration facilitates early visibility into the development process by regularly conducting software builds, thus integrating disparate software pieces earlier than later, which often times minimizes the interval between when a defect is coded and when it is discovered. Given the automated nature of continuous integration spawned builds, software teams can now start to look at their build process as something more useful than a simple compile and test process.

Groovy 101: putting Groovy to use quickly by Andrew Glover

It has been said that Grails is the addiction and Groovy is the drug. If you want to start building slick web applications rapidly with Grails it helps to start with a solid understanding of the Groovy language itself.

Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense by Andrew Glover

Are your Ant builds giant XML files that scream for attention? Why not enhance your build process to act like a quality gate, much like a test suite would?

Open Source SOA by Burr Sutter

At some point, code will be written, software tools will be acquired and systems will be built. Unfortunately the Java development world is a confused mess as it relates to a method of building a Service Oriented Architecture (SOA)-based anything. Our objective is to answer the following questions: Should I use Web Services everywhere? Is an Enterprise Services Bus (ESB) useful and required? Should I be programming in the XML-based syntax of BPEL instead of Java? Do I need JBI and/or SCA? What Open Source implementations are available to solve SOA related challenges?

JBoss ESB Deep Dive by Burr Sutter

This session will be a deep dive into the capabilities of the open source JBoss Enterprise Service Bus 4.2 GA. An ESB is primarily categorized by its capabilities in the areas of protocol mediation/abstraction, transformation, orchestration, routing, endpoint registry, etc. Numerous live demos of ESB functionality.

Internationalization and Localization in Java by David Bock

Internationalization and Localization in Java is easy, right? Everyone knows you just store your strings in some resource bundles, set the locale, wave your hands a little bit, and your application is good-to-go. Right? Maybe not... Java provides some great utilities to get started, but leaves you needing more when it comes to things like screen layout, cultural sensitivities, semantic differences in translation, use of color and iconography, and other issues.

Introducing Agility to Large Organizations by David Bock

For several years, I was a member of a team of people caught in the middle of a 200+ person software development company, with senior management wanting "buzzword compliant process improvement" such as CMMI, and engineers wanting more "agile" solutions (and people on both sides confusing Agile with ad-hoc). We were responsible for sorting it all out. Reconciling this was a herculean effort, and can be a source of lessons learned for your own process improvement efforts. Are you trying to be more agile in your organization? Are you expecting it to be harder than it needs to be because of political and bureaucratic forces beyond your control? Do you have to "educate" your senior management to protect them from buzzwords? Come learn from my successes... and mistakes.

Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools by David Bock

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? How many times have you been the "new guy" on an established project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good structural guidelines and several tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

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 2006. Embraced by developers, vendors, and open-source projects, JSF has hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico by David Geary

An introduction to the popular Prototype JavaScript framework, and two frameworks built on top of Prototype: Scriptaculous and Rico.

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.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part One by David Geary

In this session, see how you can get Ruby On Rails-like productivity on the Java side of the house with this compelling combination of technologies.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two by David Geary

A continuation of a 2-session presentation on Seam, Facelets, and Ajax4jsf.

The Google Web Toolkit, Part One by David Geary

Developing highly interactive web applications, for the most part requires knowledge of a wide array of technologies: HTML, CSS, JavaScript, XMLHttpRequest, JSP, JSF, etc. With the Google Web Toolkit (GWT), Google turns that notion of development on its head. Instead, you implement Ajax applications by writing almost entirely in Java. You use an AWT-like API, which the Google compiler compiles to JavaScript that runs on the client.

The Google Web Toolkit, Part Two by David Geary

The second part of a 2-session presentation on the Google Web Toolkit.

Distributed Teams: Remote Agility by Jared Richardson

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

Shippers Unite! by Jared Richardson

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

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.

Getting Started with Grails by Jason Rudolph

Grails is an open-source web application framework that's all about getting things done. Grails combines best-of-breed Java technologies (including Hibernate and Spring), convention over configuration, and the powerful and dynamic Groovy language. Together with these elements and Groovy's ability to seamlessly integrate with your existing Java code, Grails finally legitimizes rapid web application development for the Java platform.

Advanced Domain Models in Grails: Enterprise Integration Made Easy by Jason Rudolph

Have you seen someone develop a Rails or Grails application in a matter of minutes, only to later discover that their domain model and database schema followed conventions that are different from your existing systems? Or perhaps you're interested in using Grails, but you don't want to duplicate your existing Java domain classes in Groovy. In this session, we'll see how Grails makes it easy to hook into your pre-defined schemas or existing entity classes, while still getting all the rapid application development (RAD) goodness that Grails has to offer.

Adding Behavior to Java Annotations by John Heintz

Java's Annotations provide a way to add data to program elements. Annotations are used to configure containers, describe persistence configuration, set security roles, and are defined by nearly every recent JSR standard. This presentation explains the processing options available for consuming Annotations and demonstrates the techniques with live code demonstrations.

REST: The basics, and not so basic... by John Heintz

This presentation introduces REST, compares and contrasts REST with SOA and WS-*, and finally delves into some of the more subtle topics that affect RESTful systems.

Glassbox: Open Source Java Monitoring and Troubleshooting by John Heintz

In this session you will learn about the Glassbox open source troubleshooting and monitoring tool. Glassbox enable detection of common application problems such as database failures, slow operations, thread contention, and excessive distributed calls. Glassbox enables low overhead monitoring and troubleshooting without needing to "bake in" instrumentation up front.

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.

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.

Software Development Risk Analysis techniques by Mark Johnson

Once you leave academic "hello world" projects, software development is full of unknowns which result in the high rate of project failure we see too often in industry. Reasons for project failure vary based on the stakeholder interviewed. However, generally the project problems were caused by some unknown factor under someone else's control.

Requirements Drive Design and Development (RDDD) by Mark Johnson

Validate that requirements are not missed during the design and development process by creating Requirements document test fixtures to clarify and validate the requirements between the end users, business analysts, architects, and developers early in the project.

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, how to map and use entity relationships (1-1, 1-N, N-1, and N-N), discuss Lazy Loading, 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 switching persistence providers, versioning, compound keys, entity inheritance, and finally handling both simple and complex stored procedures. 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.

Agile Project Management with Mingle by Neal Ford

Mingle is an innovative project management tool with "skinnable religion", virtual card walls, highly customizable workflow, and role-based views. This talk describes its setup, use, and some implementation details of how it was created.

Metrics-driven Agile Development by Neal Ford

Agile software development is a highly disciplined way to build software, and one of the side effects of this discipline is the ability to gather meaningful metrics. This session describes what makes agility perfectly suited to metrics gathering and what kinds of real-world metrics you can generate.

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.

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.

Introduction to JRuby by Neal Ford

This session describes JRuby, the 100% pure-Java implementation of the Ruby programming language. It covers the basics of programming with JRuby and examples of how to integrate it into existing Java projects.

Rails for JRuby by Neal Ford

This session explains all the hype surrounding Ruby on Rails, in a context familiar to Java developers. It covers convention over configuration, ActiveRecord, controllers, views, Ajax, scaffolding, testing, and deployment...on the JVM, using JRuby.

Enterprise Applications with Spring: Part 1 by Ramnivas Laddad

This session (part 1 of the two-part session) shows the core concepts in the Spring Framework -- the most popular lightweight container that recently crossed 1 million downloads.

Enterprise Applications with Spring: Part 2 by Ramnivas Laddad

This session (the second part of the 2-part session) will cover advanced concepts in the Spring framework. While the core concepts in the first session will get you started with Spring, the advanced concepts in this session will make you effective at developing Spring-based applications.

Domain Driven Design: Simplifying Enterprise Architecture by Ramnivas Laddad

Domain-Driven Design (DDD) recommends dealing with complex software system using a domain model and preserving the model through implementation. A direct mapping between domain model and software artifacts creates simple to understand, inexpensive to implement, and easy to evolve systems. The DDD approach suggests ways to distill domain knowledge into a model and offers patterns to design and implement that model.

Leveraging annotations with AOP by Ramnivas Laddad

Specifying metadata using annotations has gained huge popularity since its introduction in Java 5. However, the story on consuming annotations isn't as clear. Reading and processing annotation is still a complex process often requiring you to understand byte-code manipulation tools and their low-level API. As a result, most developers shy away from using custom annotations, limiting their usages of annotations only those prescribed by frameworks. The result is missed opportunities for programming simplification. In this session, we explore how AOP can make it a simple task to consume annotation in a powerful manner.

Groovy: Greasing the Wheels of Java by Scott Davis

This is the year of the dynamic scripting language. Ruby (and Rails) has won the hearts and minds of many independent software developers. JavaScript is experiencing a renaissance thanks to the wild success of AJAX and websites like Google Maps. And Groovy (JSR-241) brings the same level of excitement and "scripting goodness" to the Java platform.

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.

KEYNOTE: No, I Won't Tell You Which Web Framework to Use: or The Truth (with Jokes) by Scott Davis

"Which framework should I use?" is the question most often heard on the No Fluff, Just Stuff tour. It's well worth asking. Unfortunately, there is no simple answer. After years on the tour, most speakers have crafted a response that would make any Washington politician proud -- long on style, but essentially, "Well, it depends..."

Real World Grails by Scott Davis

Scott Davis is the Editor in Chief of aboutGroovy.com. The website, in addition to being, umm, about Groovy, is implemented in Grails. This talk shows you how to get started with Grails, but also talks about the experience of using it in a live, production web site.

KEYNOTE: No, I Won't Tell You Which Web Framework to Use: or The Truth (with Jokes) by Scott Davis

"Which framework should I use?" is the question most often heard on the No Fluff, Just Stuff tour. It's well worth asking. Unfortunately, there is no simple answer. After years on the tour, most speakers have crafted

a response that would make any Washington politician proud -- long on style, but essentially, "Well, it depends..."

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.

The Busy Java Developer's Guide to java.util.concurrent by Ted Neward

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the java.util.concurrent packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

The Busy Java Developer's Guide to Threads and Synchronization by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown.

Enterprise Performance and Scalability by Ted Neward

Wondering why your enterprise Java app just? sucks? Trying to figure out why you can't get more than 10 concurrent users online at the same time? Looking for ways to try and spot the slowdowns and ways to fix them?

Java6: Exploring Mustang by Ted Neward

Mustang, the Java6 release, is out, and even if you're not looking to adopt the new platform right away, it's important to know what's there so you can start to plan for it. In this presentation, we'll go over the major new features of the Java6 platform, including the new integrated XML services capabilities (JAX-WS and JAXB), dynamic/scripting language support (javax.script), new JVM "attach" capabilities, new annotations supported by the javac compiler, and more.

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.

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.

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.

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.

Programming with Mock objects by Venkat Subramaniam

You are convinced that Test Driven Development is good for you and your project. You realize the benefits it has to offer. What's holding you back? All the code and components that your code so heavily depends on is most likely making you wonder if TDD is really for you. We will start out by looking at dependency

and dependency inversion. Then we will discuss how mock objects can help separate our code from its dependencies.

get Fit by Venkat Subramaniam

Unit testing tells you, the programmer, that your code (and the change) meets your expectations. How do you know if you are meeting your customers' expectations? Agile development is all about feedback and doing what's relevant to the customers, isn't it? Framework for Integration testing or Fit helps you to automate tests for customer expectations.