

Central Ohio Software Symposium

Crowne Plaza Dublin

June 08 - 10, 2007

<http://www.nofluffjuststuff.com/sh/2007-05-columbus>

| Fri, Jun. 08, 2007 | | | | | |
|--------------------|---|--|---|---|--|
| | Salon C-E | Salon A | Salon B | Waterford | Limerick |
| 12:00 - 1:00 PM | REGISTRATION | | | | |
| 1:00 - 1:15 PM | WELCOME | | | | |
| 1:15 - 2:45 PM | JavaServer Faces: A Whirlwind Tour David Geary | Groovy: Greasing the Wheels of Java Scott Davis | Gradual Agile: The Secret to Introducing Agile Practices Jared Richardson | 10 Ways to Improve Your Code Neal Ford | The Busy Java Developer's Guide to ClassLoaders Ted Neward |
| 2:45 - 3:15 PM | BREAK | | | | |
| 3:15 - 4:45 PM | Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary | Groovy and Java: The Integration Story Scott Davis | Distributed Teams: Remote Agility Jared Richardson | Power Regular Expressions in Java Neal Ford | Java Annotations: From Definition to Consumption Ted Neward |
| 4:45 - 5:00 PM | BREAK | | | | |
| 5:00 - 6:30 PM | Ajaxian Faces David Geary | Real World Grails Scott Davis | Build Teams, Not Products Jared Richardson | Implementing SOA Neal Ford | The Busy Java Developer's Guide to Java Platform Security Ted Neward |
| 6:30 - 7:15 PM | DINNER | | | | |
| 7:15 - 8:00 PM | KEYNOTE | | | | |

| Sat, Jun. 09, 2007 | | | | | |
|--------------------|---|--|---|---|---|
| | Salon C-E | Salon A | Salon B | Waterford | Limerick |
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary | The Zen of REST Scott Davis | Metrics-driven Agile Development Neal Ford | Domain Driven Design Venkat Subramaniam | Continuous Integration with Cruise Control Jared Richardson |
| 10:30 - 11:00 AM | BREAK | | | | |
| 11:00 - 12:30 PM | RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary | Spring 2.0: New and Noteworthy Ben Hale | Building DSLs in Static and Dynamic Languages Neal Ford | Practices of an Agile Developer Venkat Subramaniam | Mocking Web Services Scott Davis |
| 12:30 - 1:30 PM | LUNCH | | | | |
| 1:30 - 3:00 PM | The Google Web Toolkit, Part One David Geary | Spring and Hibernate in the Middle Tier Ben Hale | Shippers Unite! Jared Richardson | Introduction to JRuby Neal Ford | Drooling with Groovy and Rules Venkat Subramaniam |
| 3:00 - 3:15 PM | BREAK | | | | |
| 3:15 - 4:45 PM | The Google Web Toolkit, Part Two David Geary | AOP and JMX: A match made in heaven Ben Hale | Rails for JRuby Neal Ford | The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward | get Fit Venkat Subramaniam |
| 4:45 - 5:30 PM | BIRDS OF A FEATHER SESSIONS | | | | |

| Sun, Jun. 10, 2007 | | | | | |
|--------------------|--|--|--|--|---|
| | Salon C-E | Salon A | Salon B | Waterford | Limerick |
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | Productive Programmer: Acceleration, Focus, and Indirection Neal Ford | Java 6 Features, what's in it for you? Venkat Subramaniam | Developing Web Applications with Spring MVC Ryan Breidenbach | The Busy Java Developer's Guide to Reflection Ted Neward | Agile Software Testing Strategies Jared Richardson |
| 10:30 - 11:00 AM | BREAK | | | | |
| 11:00 - 12:30 PM | Productive Programmer: Automation and Canonicity Neal Ford | Groovy For Java Programmers Jeff Brown | OSGi: A Well Kept Secret Venkat Subramaniam | The Busy Developer's Guide to Rules and Rules Engines Using JESS Ted Neward | Subversion: A Quick Start Guide Jared Richardson |
| 12:30 - 1:15 PM | LUNCH | | | | |
| 1:15 - 2:15 PM | EXPERT PANEL DISCUSSION | | | | |
| 2:15 - 3:45 PM | Debugging and Testing the Web Tier Neal Ford | Introduction To Agile Web Development With Grails Jeff Brown | Annotation Hammer Venkat Subramaniam | Acegi Security: The security framework with the funny name Ben Hale | Pragmatic Tracer Bullets Jared Richardson |
| 3:45 - 4:00 PM | BREAK | | | | |
| 4:00 - 5:30 PM | Advanced Selenium Neal Ford | Advanced View Techniques With Grails Jeff Brown | Spring into Groovy Venkat Subramaniam | Software Development Techniques Jared Richardson | Harnessing the Power of Maven Ryan Breidenbach |

Central Ohio Software Symposium

Crowne Plaza Dublin

June 08 - 10, 2007

<http://www.nofluffjuststuff.com/sh/2007-05-columbus>

Spring 2.0: New and Noteworthy by Ben Hale

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.

Spring and Hibernate in the Middle Tier by Ben Hale

To today's JEE developer, there are two indispensable tools for creating applications; Spring and Hibernate. Together these two frameworks comprise one of the most powerful and often used stacks in the industry. While it is possible to do amazing things it's not always obvious how best to use them to maximize value. This session aims to correct that.

AOP and JMX: A match made in heaven by Ben Hale

You're winding down project and you get that dreaded email from your project manager, "How hard would it be to add some performance monitoring to the system?" Well, after this session, you'll be able to respond, "No problem at all!" It turns out that with a pinch of AOP and a dash JMX, you can introduce amazing management and monitoring capabilities without changing you're mainline code one bit.

Acegi Security: The security framework with the funny name by Ben Hale

Security is one of the major requirements in modern day enterprise applications and yet it is also one of the weakest parts of most developers toolboxes. The problem is of course that security is HARD! It turns out that rather than reinventing the wheel for each application, developers can turn to a great security framework out there already; Acegi.

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.

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.

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.

Shippers Unite! by Jared Richardson

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

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.

Subversion: A Quick Start Guide by Jared Richardson

Subversion is a free source code management system that's very powerful.

Pragmatic Tracer Bullets by Jared Richardson

Are your product designs hit or miss? Do you have trouble building a loosely coupled system? Is your code incestuous? Refactoring not an option with your code base? Tracer Bullets help keep your project out of the fire. Tracer Bullet Development: * helps you create great software * lends itself to an iterative cycle * can be used for demos early and often * is easily refactored * allows your teams to work in parallel * makes a very testable system

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.

Groovy For Java Programmers by Jeff Brown

Groovy is an agile dynamic language for the Java platform. Groovy has a Java like syntax along with many features inspired by languages like Python, Ruby and Smalltalk. This session covers a lot of ground including many interactive examples to highlight the powerful language features that make Groovy compelling. A lot of momentum is building in the Groovy and Grails communities right now and this session is aimed at Java developers who want to leverage the power of Groovy.

Introduction To Agile Web Development With Grails by Jeff Brown

Grails brings the powerful "coding by convention" paradigm to Groovy and Java. Grails is not just another flavor in the pool of web development frameworks for Java. Grails leverages the powerful dynamic features of Groovy while taking advantage of best of breed technologies like Hibernate, Spring, Sitemesh and Quartz to make web application development both fun and easy.

Advanced View Techniques With Grails by Jeff Brown

Grails provides view technologies that offer great flexibility and power without the complexity introduced by other Java web application frameworks. Custom tag libraries are a snap. GSP Templates provide a simple mechanism for reusing UI elements. Sitemesh is integrated to help provide a consistent presentation across the entire application. Grails provides simple mechanisms for leveraging the power of Ajax.

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.

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.

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.

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.

Building DSLs in Static and Dynamic Languages by Neal Ford

This session discusses building Domain Specific Languages and DSL-style code in Java, Groovy, and Ruby. It discusses the different types of DSLs, details on how to implement them in Java, Groovy, and Ruby, and example problem domains where DSLs make sense.

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.

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.

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.

Developing Web Applications with Spring MVC by Ryan Breidenbach

Although Spring's rise to ubiquity was driven by its IoC and AOP services, it offers so much more. One of the most powerful of these services is Spring's MVC framework. If you are familiar with other Java web frameworks such as Struts or WebWork, you will recognize that Spring MVC is a similar request-response web framework driven by a Front Controller, in this case the DispatcherServlet. But Spring also provides

considerable functionality out of the box. One of the beauties of the Spring MVC is that, like the rest of Spring, all of its components are quite modular. This allows you to mix and match different implementations of the various MVC components as you see fit for your application. In addition, Spring MVC provides rich capabilities for data binding. This allows you to transparently bind form parameters to your application objects without having to create any intermediate objects. Spring's validation features allow you to capture both binding and application errors. And with Spring 2.0's much improved JSP tag library, both creating forms and displaying validation errors is quite simple.

Harnessing the Power of Maven by Ryan Breidenbach

2006 appeared to be the year that Maven achieved the momentum it needed to overtake Ant as the build tool of choice for Java developers. A lot of that has to do with the vastly improved Maven 2. But a lot of it has to do with the simplicity, organization and power that Maven brings to projects. The session will bring developers new to Maven with everything it has to offer. This includes creating your very first Maven project, learning the significance of the POM file, how to let Maven and its repositories manage your dependencies, and how to let Maven report on the health of your own projects. And for the Ant users in the audience, you will get to see a side-by-side comparison of the two build tools' build philosophies.

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.

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.

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.

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 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.

Java Annotations: From Definition to Consumption by Ted Neward

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll

look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

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.

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 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.

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.

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.

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.

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.

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.

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.

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.

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.