

# Desert Southwest Software Symposium

Crowne Plaza Phoenix Airport

July 27 - 29, 2007

<http://www.nofluffjuststuff.com/sh/2007-07-phoenix>

(event schedule as of July 27, 2007)

Fri, Jul. 27, 2007					
	Terminal 1	Terminal 2/3	Terminal 4	Terminal 5 East	Terminal 5 West
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	The Busy Java Developer's Guide to Java Platform Security Ted Neward	JavaServer Faces: A Whirlwind Tour David Geary	Annotation Hammer Venkat Subramaniam	Groovy: Greasing the Wheels of Java Scott Davis	Gradual Agile: The Secret to Introducing Agile Practices Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary	Domain Driven Design Venkat Subramaniam	Groovy and Java: The Integration Story Scott Davis	Shippers Unite! Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	The Busy Java Developer's Guide to Reflection Ted Neward	Agile Software Testing Strategies Jared Richardson	OSGi: A Well Kept Secret Venkat Subramaniam	Real World Grails Scott Davis	Ajaxian Faces David Geary
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Scott Davis				

Sat, Jul. 28, 2007					
	Terminal 1	Terminal 2/3	Terminal 4	Terminal 5 East	Terminal 5 West
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Metrics-driven Agile Development Neal Ford	The Zen of REST Scott Davis	Continuous Integration with Cruise Control Jared Richardson	Drooling with Groovy and Rules Venkat Subramaniam
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Software Development Techniques Jared Richardson	Mocking Web Services Scott Davis	Building DSLs in Static and Dynamic Languages Neal Ford	Designing for Ajax Nathaniel Schutta
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	The Google Web Toolkit, Part One David Geary	10 Ways to Improve Your Code Neal Ford	Spring 2.0: New and Noteworthy Ben Hale	get Fit Venkat Subramaniam	Ruby for Java programmers Bruce Tate
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	The Google Web Toolkit, Part Two David Geary	Implementing SOA Neal Ford	Spring and Hibernate in the Middle Tier Ben Hale	Build Teams, Not Products Jared Richardson	Rails for Java Programmers Bruce Tate
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Jul. 29, 2007					
	Terminal 1	Terminal 2/3	Terminal 4	Terminal 5 East	Terminal 5 West
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Introduction to Object-Relational Mapping with Hibernate Brian Sam-Bodden	What's New in Java 6 Jason Hunter	Building ChangingThePresent: Agility in Action Bruce Tate	Debugging and Testing the Web Tier Neal Ford	Dynamic Languages and the JVM Nathaniel Schutta
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	10 ways to use Hibernate effectively Brian Sam-Bodden	Test Infecting the Legacy Organization Nathaniel Schutta	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Advanced Selenium Neal Ford	XQuery By Example: Building an Email Archive System Jason Hunter
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Business Rules Engines in Java and J2EE- An Introduction to the Drools Rules Engine Brian Sam-Bodden	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford	AOP and JMX: A match made in heaven Ben Hale	Introduction To Agile Web Development With Grails Jeff Brown	Web Publishing 2.0 Jason Hunter
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Complex Builds with Ant Brian Sam-Bodden	Productive Programmer: Automation and Canonicity Neal Ford	Acegi Security: The security framework with the funny name Ben Hale	Advanced View Techniques With Grails Jeff Brown	Forgotten Web Algorithms Jason Hunter

# Desert Southwest Software Symposium

Crowne Plaza Phoenix Airport

July 27 - 29, 2007

<http://www.nofluffjuststuff.com/sh/2007-07-phoenix>  
(event schedule as of July 27, 2007)

## **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 a 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 of JMX, you can introduce amazing management and monitoring capabilities without changing your 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.

## **Introduction to Object-Relational Mapping with Hibernate by Brian Sam-Bodden**

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

## **10 ways to use Hibernate effectively by Brian Sam-Bodden**

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

## **Business Rules Engines in Java and J2EE- An Introduction to the Drools Rules Engine by Brian Sam-Bodden**

Software development is expensive, when business rules are hard-coded in your application's source code, changes and additions to those rules translate to wasted time and money. Good object-oriented, component-based approaches can alleviate the burden of keeping up with changes in the business world but they still require that expert knowledge of the changes be passed from the decision makers to the business analysts and finally to programmers that need to implement these changes. Business Rule Engines and Business Rule Languages are based on the basic premise of separation of concerns by empowering business domain experts to express the rules of business in a way that it is directly usable by applications.

## **Complex Builds with Ant by Brian Sam-Bodden**

Ant has revolutionized the way we build applications in Java and it has become a de facto standard in the Java world. As applications grow in complexity some developers are finding themselves dealing with ever growing and complex builds. Complex builds have to deal with Multiple Operating System, multiple Application Servers, multiple APIs and multiple stages of development.

## **Ruby for Java programmers by Bruce Tate**

With the explosion of Ruby on Rails and the Java community interest in features like closures and continuations, the Ruby programming language is an excellent one for all developers to know. As the JRuby virtual machine picks up steam, Ruby becomes a must language to understand. The best way to learn Ruby is to see it in action.

### **Rails for Java Programmers by Bruce Tate**

The productivity of Ruby on Rails cannot be denied, but the explosion of Ruby on Rails left many developers, with hard commitments to Java deployment platforms, out in the cold. The continued evolution of JRuby can change that. JRuby is a Ruby implementation on the Java virtual machine. And yes, it runs Rails. In this session, you will learn Rails as it was meant to be learned, within the context of building a live site, from scratch.

### **Building ChangingThePresent: Agility in Action by Bruce Tate**

ChangingThePresent is the increasingly popular charity donations portal that lets you give donation gifts instead of another pair of fuzzy slippers. The site is built and maintained under unusual circumstances. The team is distributed, with no more than two developers in any one place. The team uses agile techniques such as automated testing, heavy customer involvement, and a SCRUM-like release plan to deliver the core features.

### **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.

### **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.

### **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.

### **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.

### **Build Teams, Not Products by Jared Richardson**

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

### **What's New in Java 6 by Jason Hunter**

The Java 6 (Mustang) release should make your life easier, for a change. It doesn't alter the core language like Java 5 did. It doesn't pack in so many sub-JSRs that you'll be overwhelmed by the amount you have to learn. Instead Java 6 adds several handy things that honestly should have been added before. Among the improvements we'll cover in this fast-paced class: \* A new Console class \* A real Compiler API \* A GIF writer \* Pluggable Locale data \* Access to disk partition size data \* Array reallocation \* Low-level floating point functions \* Reflective access to parameter names \* Access to network interface details \* Pluggable annotation processing \* Improved class file format \* Streaming XML with StAX \* A new Scripting interface

### **XQuery By Example: Building an Email Archive System by Jason Hunter**

The classic searchable email archive system is cluged together -- a frankenstein monster combining a relational database with a search engine, with Java just barely able to keep the two together. In this talk we'll demonstrate how email is more content than data, how it's better encoded in XML rather than relational tables, and how Java can convert emails to XML and drive an XQuery backend to produce a simpler and more scalable email archive system.

### **Web Publishing 2.0 by Jason Hunter**

If we're moving toward Web 2.0, what does that mean for online publishing? In this talk I'll answer that question. Based on my experience as Principal Technologist at Mark Logic working with dozens of the largest online publishers, I'll present a vision for how the Web 2.0 concepts like personalization, collective intelligence, the long tail, and the importance of "owning the data" can and should reshape the face of online publishing -- and how XML, XQuery, and XML-aware text search act as the key enablers. I'll also introduce new Web Publishing 2.0 concepts like "Sweat the content" and "Give answers not links".

### **Forgotten Web Algorithms by Jason Hunter**

In this talk I'll explain -- without any needless math or boring proofs -- several fun algorithms of interest to back-end web programmers. Each algorithm was selected because it's really practical, really interesting, or both. The algorithms aren't always the same but can include: public key cryptography, credit card checksum validation, TCP Slow Start, two's complement, priority queues, the XOR swap, and the Google MapReduce function for massively distributed calculation.

### **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.

### **Designing for Ajax by Nathaniel Schutta**

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decide where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

### **Dynamic Languages and the JVM by Nathaniel Schutta**

With all the attention being paid to Ruby and its hip cousin Rails, many in the Java camp may be feeling like their party invitation is "lost in the mail". Fear not loyal Java lovers, the dynamic language meme is alive and well in your space! Between numerous JSRs and various languages, the JVM is becoming quite the dynamic disco. After an overview of what it means to be dynamic, this talk will look at JRuby, Groovy, and Rhino.

### **Test Infecting the Legacy Organization by Nathaniel Schutta**

When starting a new project, most developers make sure that testing is a priority. However, only the lucky few live in the idyllic world of greenfield development; the vast majority of us must contend with code written when "test" was a four letter word and testing was the sole responsibility of that "other" organization. We'll examine some techniques for introducing testing - not just to your code but to the rest of your development organization.

### **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.

### **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.

### **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.

### **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.

### **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.

### **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 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 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.

### **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.

### **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.

### **Drizzling 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.