

Greater Wisconsin Software Symposium

Sheraton Milwaukee Brookfield

March 02 - 04, 2007

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

Fri, Mar. 02, 2007						
	Brookfield 1	Brookfield 2	Brookfield 3	Brookfield 4	Brookfield 5	Moorland 7
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Data Integration : Beyond Cutesy Mashups Brian Sletten	Mocking Web Services Scott Davis	JavaServer Faces: A Whirlwind Tour David Geary	Groovy for Java Programmers Venkat Subramaniam	Structuring concurrent applications in JDK 5.0 Brian Goetz	Agile Immersion David Hussman
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	NetKernel : XML Processing for the 21st Century Brian Sletten	The Zen of REST Scott Davis	Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary	Agile Web Development with Grails Venkat Subramaniam	Effective Concurrent Java Brian Goetz	Creating Agile Requirements David Hussman
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Abusing Maven For Fun and Profit : (Near) Zero-Admin Deployments Brian Sletten	Atom: From Blogging to Data Syndication Scott Davis	Ajaxian Faces David Geary	Spring into Groovy Venkat Subramaniam	The Java Memory Model Brian Goetz	Executable Documentation David Hussman
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Scott Davis					

Sat, Mar. 03, 2007						
	Brookfield 1	Brookfield 2	Brookfield 3	Brookfield 4	Brookfield 5	Moorland 7
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Squashing bugs with FindBugs Brian Goetz	Annotation Hammer Venkat Subramaniam	Give it a REST Brian Sletten	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Continuous Integration with Cruise Control Jared Richardson	Cutting an Agile Groove: Agile Transition Tips David Hussman
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Subversion: A Quick Start Guide Jared Richardson	Domain Driven Design Venkat Subramaniam	Spring 2.0: New and Noteworthy Ben Hale	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Ajax development with the Yahoo! UI Library and Grails Scott Davis	Getting Agile Planning and Tracking Up and Running David Hussman
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	Groovy and Java: The Integration Story Scott Davis	Effective Hibernate Scott Leberknight	Acegi Security: The security framework with the funny name Ben Hale	The Google Web Toolkit, Part One David Geary	Java Performance Myths Brian Goetz	Agile Software Testing Strategies Jared Richardson
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Real World Grails Scott Davis	Spring/Hibernate Integration Basics Scott Leberknight	AOP and JMX: A match made in heaven Ben Hale	The Google Web Toolkit, Part Two David Geary	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Making Architecture Work Through Agility Mark Richards
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS					

Sun, Mar. 04, 2007						
	Brookfield 1	Brookfield 2	Brookfield 3	Brookfield 4	Brookfield 5	Moorland 7
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Bringing Ruby & Rails to the JVM Charles Nutter	Java 6 Features, what's in it for you? Venkat Subramaniam	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Intro to Java Persistence API (JPA) Mark Richards	Debugging and Testing the Web Tier Neal Ford	Build Teams, Not Products Jared Richardson
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Become Super Powerful with JRuby Charles Nutter	Java NIO Brian Pontarelli	The Busy Developer's Guide to Rules and Rules Engines Using JESS Ted Neward	Advanced Java Persistence API (JPA) Mark Richards	Advanced Selenium Neal Ford	Shippers Unite! Jared Richardson
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	OSGi: A Well Kept Secret Venkat Subramaniam	ACEGI Authentication - The AJAX way Brian Pontarelli	The Busy Java Developer's Guide to Reflection Ted Neward	Introducing the iBATIS Persistence Framework Mark Richards	Implementing SOA Neal Ford	The Cornerstone of a Great Shop Jared Richardson
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Drizzling with Groovy and Rules Venkat Subramaniam	Versioning your SOA Brian Pontarelli	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Making The Right Persistence Framework Choice Mark Richards	10 Ways to Improve Your Code Neal Ford	Software Development Techniques Jared Richardson

Greater Wisconsin Software Symposium

Sheraton Milwaukee Brookfield

March 02 - 04, 2007

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

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.

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.

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.

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.

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.

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.

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.

Java NIO by Brian Pontarelli

The Java NIO packages that were added in JDK 1.4 and these packages allow Java applications to perform true non-blocking IO operations. This presentation will cover the basics of the standard IO packages, which date back to the beginning of Java, and some of the shortcomings they have. This will be followed by coverage of the newer NIO packages and how they address these issues.

ACEGI Authentication - The AJAX way by Brian Pontarelli

Learn how to use ACEGI in an AJAX friendly way so that when you user's sessions expire, they can quickly log back into the application and continue where they left off. This will cover in detail the changes that were made to the ACEGI framework to support AJAX authentication, why ACEGI needed to be changed and how to implement this solution in a new or existing application that uses the ACEGI framework for authentication.

Versioning your SOA by Brian Pontarelli

Learn how to manage service oriented architecture applications over time. This talk will focus on how to deploy a SOA application and version components individually. It discusses the finer points of upgrades and how to architect your system so that each deployment doesn't mean stopping and starting "the whole world" and how to attempt to achieve the "four nines" (99.99%) uptime ideal.

Data Integration : Beyond Cutesy Mashups by Brian Sletten

It seems like you can't sit through any kind of presentation these days without hearing about how cool web mashups are. Google Maps is the Bomb and being able to map Starbucks stores is something to build a company out of. As if.

NetKernel : XML Processing for the 21st Century by Brian Sletten

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

Abusing Maven For Fun and Profit : (Near) Zero-Admin Deployments by Brian Sletten

Ok, I can't promise you profit, but hopefully you'll have fun. Maven 2 introduces a number of new features (including that performance feature) that make it a swell project management tool for development. Come hear about how we can abuse Maven to manage distributed deployment scenarios before the Modules JSR is done.

Give it a REST by Brian Sletten

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

Bringing Ruby & Rails to the JVM by Charles Nutter

The Ruby programming language has exploded in popularity, spurred in part by the agility of the Rails web framework. Rails has in turn changed the way we look at web development. The two together are forcing developers to rethink how applications should be written. The world is changing. With JRuby you're now able to run Rails apps alongside your existing Java applications, calling the same services and leveraging the same infrastructure. All the scalability, reliability, and performance of Enterprise Java is now available to Rails developers.

Become Super Powerful with JRuby by Charles Nutter

The explosion of popularity for dynamic languages on the JVM has changed the way we look at development. No longer is the Java platform tied to a single language, and no longer do you only have a single tool in your toolbox. Dynamic languages like Groovy, Python, Ruby, and others enable entirely new ways of solving our software problems. This session will explore one of those languages, Ruby, and show how it will make even complicated development tasks manageable and even fun.

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.

Agile Immersion by David Hussman

Have you heard about SCRUM or XP but never done it? If you want to give it a try, this session will allow you to participate in planning and executing several agile iterations. A working knowledge of either XP or SCRUM will be helpful but not mandatory.

Creating Agile Requirements by David Hussman

Successful project communities balance written requirements with a healthy amount of discussion. This is at the core of requirements that could be deemed #agile#. Many agile projects choose to use user stories, but others may be using use cases or other forms of written requirements. This session is for anyone wanting to improve their requirements, including the creation of good requirement and the presentation styles that help people focus on creating great software products, and stop focusing on documents.

Executable Documentation by David Hussman

Why is so much documentation worthless? Wouldn't it be nice if your documentation actually reflected what your system does? One way to do this is through executable documentation. If you are struggling with ambiguous requirements, lack of contact with the business, or a chasm between development and testing, this session is for you.

Cutting an Agile Groove: Agile Transition Tips by David Hussman

If you are thinking about, just starting, or in the midst of transitioning to agile, this session is overflowing with practices, tip, techniques, and experiences. Stop talking about whether or not you are doing agile and come learn how to setup and maintain agility that extends beyond a single project.

Getting Agile Planning and Tracking Up and Running by David Hussman

If your company is using agile or thinking about it, this session will show you how to plan and tracking an agile project. Examples projects will be discussed, including the glory and horrors. Various planning tools that help distributed teams will be presented as well as a collection of lo-fi tools which truly help find and address the issue that plagues so many projects: #when are we going to complete this project#.

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.

Subversion: A Quick Start Guide by Jared Richardson

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

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.

Build Teams, Not Products by Jared Richardson

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

Shippers Unite! by Jared Richardson

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

The Cornerstone of a Great Shop by Jared Richardson

Continuous Integration drives a number of key practices (build automation, test automation, small check ins, etc), so it's a great first practice to bring into your shop.

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.

Dependency Management Techniques by Kirk Knoernschild

Why is software so difficult to change? When you establish your initial vision for the software's design and architecture, you imagine a system that is easy to modify, extend, and maintain. Unfortunately, as time passes, changes trickle in that exercise your design in unexpected ways. Unlike what you had anticipated, each change begins to resemble nothing more than another hack, until finally the system becomes a tangled web of code that few developers care to venture through. Eventually, modifications to the software intended to improve the system have the opposite affect of breaking other parts of the system. The software is beginning to rot.

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.

Introducing the iBATIS Persistence Framework by Mark Richards

Hibernate has evolved as the de facto standard for persistence in most Java-based applications. However, many people are turning to iBATIS as an open source persistence alternative. iBATIS is a powerful open source persistence framework that is rapidly gaining in popularity, particularly within the Spring community. In this session you will learn why iBATIS is becoming so popular, how iBATIS differs from Hibernate and JPA (JSR-220), and how to use iBATIS. Through interactive coding examples I will demonstrate how to configure iBATIS within Spring and also as standalone, how to map SQL statements, and how to map Stored Procedures. I will also discuss and demonstrate techniques for dynamic SQL using iBATIS, using XML with iBATIS, and finally the various caching strategies available within iBATIS. We will end this session by discussing various techniques and best practices for using iBATIS in small and large-scale applications.

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.

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.

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.

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.

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.

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.

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..."

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.

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.

Effective Hibernate by Scott Leberknight

Hibernate seems simple on the surface yet when you go beyond very simple use cases it can become much more complex. Intended for beginner to intermediate-level Hibernate developers, come see how to put Hibernate to effective use on your projects.

Spring/Hibernate Integration Basics by Scott Leberknight

Hibernate is a very popular Java transparent persistence framework, but you often need to create additional infrastructure to manage sessions, transactions, and lazy-loading in a clean and elegant manner. See how Spring can help.

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 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 and an interface and a factory.

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.

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.

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.

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.

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.