

Great Lakes Software Symposium

Westin Chicago Northwest

November 16 - 18, 2007

<http://www.nofluffjuststuff.com/sh/2007-11-chicago>
(event schedule as of November 14, 2007)

Fri, Nov. 16, 2007						
	Ballroom 3/4	Chambers	Radcliffe	Prescott	Marlborough	Leighton
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Give it a REST Brian Sletten	Enterprise Performance and Scalability Ted Neward	Creating Agile Requirements David Hussman	Groovy: Greasing the Wheels of Java Scott Davis	JavaScript for Java Programmers Stuart Halloway	Java 6 Features, what's in it for you? Venkat Subramaniam
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Domain Driven Design Venkat Subramaniam	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Getting Agile Planning and Tracking Up and Running David Hussman	Groovy and Java: The Integration Story Scott Davis	Prototype: Ajax and JavaScript++ Stuart Halloway	RESTlet for the Weary Brian Sletten
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	10 Ways to Improve Your Code Neal Ford	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Executable Documentation David Hussman	This Week In Refactoring Stuart Halloway	Real World Grails Scott Davis	NetKernel : XML Processing for the 21st Century Brian Sletten
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Scott Davis					

Sat, Nov. 17, 2007						
	Ballroom 3/4	Chambers	Radcliffe	Prescott	Marlborough	Leighton
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	OSGi: A Well Kept Secret Venkat Subramaniam	Implementing SOA Neal Ford	The Secrets of GORM Scott Davis	Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman	Developing Applications Using in Spring 2.0 Ryan Breidenbach	Designing for Ajax Nathaniel Schutta
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Spring into Groovy Venkat Subramaniam	Building DSLs in Static and Dynamic Languages Neal Ford	Continuous Integration with Cruise Control Jared Richardson	Leading Agile Projects: Maintaining Sustainable Agility David Hussman	Java Annotations: From Definition to Consumption Ted Neward	Test Infecting the Legacy Organization Nathaniel Schutta
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Introduction to JRuby Neal Ford	Applied AOP Brian Sletten	Introduction to Hibernate Scott Leberknight	Shippers Unite! Jared Richardson	The Busy Java Developer's Guide to ClassLoaders Ted Neward
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Rails for JRuby Neal Ford	Harnessing the Power of Maven Ryan Breidenbach	Spring/Hibernate Integration Basics Scott Leberknight	Software Development Techniques Jared Richardson	Database Refactoring Pramod Sadalage
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS					

Sun, Nov. 18, 2007						
	Ballroom 3/4	Chambers	Radcliffe	Prescott	Marlborough	Leighton
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Making The Right Persistence Framework Choice Mark Richards	The Google Web Toolkit, Part One David Geary	Groovy For Java Programmers Jeff Brown	JMX and Spring: Manageability for Spring-based Applications Ken Sipe	Code Metrics & Analysis for Agile Projects Neal Ford	Advanced Hibernate Scott Leberknight
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	EJB3 Core Specification (JSR-220) Mark Richards	The Google Web Toolkit, Part Two David Geary	Test Driven Development With Groovy And Grails Jeff Brown	The Busy Java Developer's Guide to Reflection Ted Neward	FP for Java Programmers Venkat Subramaniam	The Art of Producing Software: Applying Lean Concepts to Transform Your Software Development Organization John Carnell
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	Introduction To Agile Web Development With Grails Jeff Brown	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford	Introducing the Eclipse Rich Client Platform Scott Delap	Java Memory, Performance and the Garbage Collector Ken Sipe	Intro to Java Persistence API (JPA) Mark Richards	Build Teams, Not Products Jared Richardson
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Advanced Techniques With Grails Jeff Brown	Productive Programmer: Automation and Canonicity Neal Ford	Developing Enterprise Business Applications in Eclipse RCP Scott Delap		Advanced Java Persistence API (JPA) Mark Richards	Agile Software Testing Strategies Jared Richardson

Great Lakes Software Symposium

Westin Chicago Northwest

November 16 - 18, 2007

<http://www.nofluffjuststuff.com/sh/2007-11-chicago>
(event schedule as of November 14, 2007)

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.

RESTlet for the Weary by Brian Sletten

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community.

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.

Applied AOP by Brian Sletten

Most people new to Aspect-Oriented Programming (AOP) are fed up with separation of concerns zealots explaining how great their techniques are at dealing with... logging. Ok, you get it. Logging is a cross-cutting concern that can be appropriately modularized. What else does AOP have to offer? A lot, it turns out. This talk will give an introduction to the motivations of AOP as well as a series of concrete examples drawn from enterprise and client side Java. Come learn how AspectJ-flavored AOP can begin to benefit you immediately either in development or production environments. Learn how to enforce architectural policies, find Swing threading issues, reduce the invasiveness of the Observer design pattern or even improve the reusability of your domain models. Now that Spring 2.0 provides support for AspectJ, the time has never been better to learn about these new (but backwards compatible) ways of thinking about building software.

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.

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.

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".

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 to create what is being called executable documentation or executable specifications. 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.

Leading Agile Projects: Finding Your Groove in the First 4 Iterations by David Hussman

Summary: There are many books about agile but few provide a path for guiding you through the beginning of an agile project. Whether you are preparing for your first agile project, or taking the lead for the first time, this session will provide a guided tour filled with practical advice and a pile of anecdotes.

Leading Agile Projects: Maintaining Sustainable Agility by David Hussman

Once your agile project is rolling along, there are many bumps and roadblocks which can derail the train. Whether you are leading the project formally or informally, there are techniques you can use to keep the project alive and innovative. This session will cover skills and techniques for leading sustainable project communities.

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.

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?

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.

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.

Test Driven Development With Groovy And Grails by Jeff Brown

The value of Test Driven Development (TDD) has become widely accepted. The practice has extended beyond just XP teams. Good TDD practices yield high quality software and help teams maintain confidence in their software as complexity grows. The dynamic nature of Groovy makes TDD easy and fun. Groovy may

be used to unit test not only Groovy code but other code as well. Testing Java code with Groovy is a snap. Learn to use the power of Groovy to test your systems.

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 Techniques With Grails by Jeff Brown

Grails represents technology that offers 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.

The Art of Producing Software: Applying Lean Concepts to Transform Your Software Development Organization by John Carnell

Waste is an insidious beast that drains the productivity of development teams and the organizations they work in. Many organizations are now realizing that by turning their gaze inward they can streamline their overall development processes, deliver higher quality products faster and save significant amounts of money. In this talk we will look at the ?Lean? techniques first developed by companies like Toyota and how they can be applied to common software development practices. We will walk through such concepts as identifying the different types of waste you might encounter in a software development effort, using Value Stream Mapping (VSM) to help measure the impact of that waste and different techniques you can use to eliminate that waste.

JMX and Spring: Manageability for Spring-based Applications by Ken Sipe

This session describes management of Java resources using the Java Management Extensions JMX API. JMX provides a unified framework to instrument Java systems with monitoring and management capabilities.

Java Memory, Performance and the Garbage Collector by Ken Sipe

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

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.

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.

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.

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.

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.

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.

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.

Code Metrics & Analysis for Agile Projects by Neal Ford

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

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 Canonality by Neal Ford

This session discusses how to use the Productive Programmer principles of automation and canonality 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.

Database Refactoring by Pramod Sadalage

Evolve an existing database schema a small bit at a time to improve the quality of its design without changing its semantics.

Developing Applications Using in Spring 2.0 by Ryan Breidenbach

In 2006, Spring 2.0 was released and Spring became ubiquitous in the enterprise Java space. So if you haven't taken a peek at Spring yet, what are you waiting for?!? If the answer is "This session," you have come to the right place. This session will familiarize you with the benefits Spring provides and why they have made the framework so successful. In addition, we will focus on the efforts that went into Spring 2.0 to make these benefits much more developer friendly. By the end of this session, you should have a grasp of how Spring 2.0 simplifies the utilization of Spring's core features such as bean wiring, AOP, and the simplification of using enterprise Java. If your head is not spinning too much.

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 of 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.

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 Secrets of GORM by Scott Davis

GORM (the Grails Object/Relational Mapper) is one of the many high points of the Grails web framework. GORM is a thin Groovy wrapper over Hibernate, but that doesn't begin to capture excitement of what GORM brings to the party. Imagine being able to call `book.save()` and `book.delete()` on your `Book` class; calling `Book.get(1)` to retrieve your book from the database by primary key; using `Book.list()` to pull an `ArrayList` of `Book` objects into your application. Now imagine getting all of that functionality (and more) for free with each new class you define. No interfaces to implement. No abstract classes to extend. Persistence that is transparent, automatic, and simple to use: GORM.

Introducing the Eclipse Rich Client Platform by Scott Delap

Rich client application development using Java can be intimidating giving the vast flexibility in application design and structure. It also can be frustrating to create the large number of support services (persistence, menus, event and job frameworks) that a large scale rich client applications needs. The Eclipse Rich Client Platform is one project attempting to solve these issues by providing a core infrastructure that not only provides the day to day services a rich client application developer needs, but also providing a suggested path to guide you down the road of designing your application. This presentation introduces both the Eclipse RCP and the tools provided by the Eclipse IDE that assist developers in writing RCP apps.

Developing Enterprise Business Applications in Eclipse RCP by Scott Delap

This tutorial emerges out of the combined experience the presenter gained while working on a large Eclipse RCP/J2EE enterprise application (2.3 million lines of code in the RCP application). It will address the gap between the standard functionality of Eclipse RCP and what is needed for the creation of polished highly usable business applications.

Introduction to Hibernate by Scott Leberknight

This session introduces the Hibernate Object/Relational Mapping (ORM) framework, showing the basics of persisting Java objects to relational databases. No prior knowledge of Hibernate or ORM is assumed.

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.

Advanced Hibernate by Scott Leberknight

This session covers advanced Hibernate topics beyond simple object persistence including session management, object locking, detachment and versioning, lazy loading performance issues and query tuning, advanced O/R mapping support, legacy database considerations, and the Hibernate cache architecture.

JavaScript for Java Programmers by Stuart Holloway

This presentation covers JavaScript from the perspective of a Java programmer. We assume that you may be using an Ajax toolkit, but still need to be able to read, modify, and test the JavaScript code in your application. You will learn the common idioms of JavaScript by looking at working code from Prototype and Scriptaculous.

Prototype: Ajax and JavaScript++ by Stuart Holloway

Learn to simplify Ajax development with Prototype through a series of real-world examples. Along the way, learn to code in Prototype's modern JavaScript style, taking advantage of Prototype's extensions to JavaScript's object model

This Week In Refactoring by Stuart Halloway

Contributing to open source is great for your career. In a few short hours, you can learn, teach, promote your skills, and improve the quality of the community. In this talk, we will show you how, by doing it.

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?

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

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

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

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

Java 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 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 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.

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.

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.

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.

FP for Java Programmers by Venkat Subramaniam

Most interest around Functional Programming (FP) has been academic until recently. Recent commercial languages are beginning to exploit FP features. Knowing more about FP will not only help us make better use of these features, but to exploit those. In this session we will take a close look at FP.