

Great Lakes Software Symposium

Westin Chicago Northwest

April 18 - 20, 2008

<http://www.nofluffjuststuff.com/conference/chicago/2008/04/index.html>

(event schedule as of April 19, 2008)

Fri, Apr. 18, 2008					
	Grand Ballroom 1&2	Ballroom 4	Chambers	Marlborough	Leighton
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Structuring concurrent applications in JDK 5.0 Brian Goetz	10 Things Every Software Architect Should Know Richard Monson-Haefel	JavaServer Faces: A Whirlwind Tour David Geary	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Spring 2.5 - Spring without XML Ken Sipe	Effective Concurrent Java Brian Goetz	Developing Rich Internet Applications Richard Monson-Haefel	Facelets David Geary	Build Teams, Not Products Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	JMX and Spring: Manageability for Spring-based Applications Ken Sipe	Java Performance Myths Brian Goetz	Understanding Open Source Licensing Richard Monson-Haefel	Seam David Geary	10 Tips for Getting Your Project Back on Track Jared Richardson
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Jared Richardson				

Sat, Apr. 19, 2008					
	Grand Ballroom 1&2	Ballroom 4	Chambers	Marlborough	Leighton
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Architecture and Scaling Ken Sipe	A Thorough Introduction To Groovy Jeff Brown	The Java Memory Model Brian Goetz	Rich Faces David Geary	Agile Software Testing Strategies Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	7 Habits of Highly Effective Developers Ken Sipe	Agile Test Driven Development With Groovy Jeff Brown	Beyond ACID: transactions management, in theory and practice Brian Goetz	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Techniques 2008 Jared Richardson
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Squashing bugs with FindBugs Brian Goetz	Powerful Metaprogramming Techniques With Groovy Jeff Brown	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Give it a REST Brian Sletten	Restoring Agility: Getting Your Team Back on Track Jared Richardson
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	SOA Unplugged Mark Richards	Guice Dependency Injection Brian Pontarelli	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	REST - Live! Brian Sletten	Iteration 0 Ken Sipe

Sun, Apr. 20, 2008					
	Grand Ballroom 1&2	Ballroom 4	Chambers	Marlborough	Leighton
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Agile, Smagile: What's Working? - What's Not? David Hussman	Enterprise Messaging Using JMS (Part 1) Mark Richards	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Monitoring Software Quality with Continuous Integration Andrew Glover	RESTlet for the Weary Brian Sletten
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Architecture and Agility Are Not Mutually Exclusive David Hussman	Enterprise Messaging With JMS (Part 2) Mark Richards	Advanced Web Development With Grails Jeff Brown	Tactical Continuous Integration with Hudson Andrew Glover	Dependency management Brian Pontarelli
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman	EJB3 Core Specification (JSR-220) Mark Richards	Struts 2 basics Brian Pontarelli	Rich Clients, Rich Data Part I : Linking Brian Sletten	Groovin' builds Gant get any easier Andrew Glover
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Leading Agile Projects: Maintaining Sustainable Agility David Hussman	Java Persistence: Approaching the Silver Bullet Mark Richards	Struts 2 convention over configuration Brian Pontarelli	Rich Clients, Rich Data Part II : Consuming Brian Sletten	Easy BDD with Groovy Andrew Glover

Great Lakes Software Symposium

Westin Chicago Northwest

April 18 - 20, 2008

<http://www.nofluffjuststuff.com/conference/chicago/2008/04/index.html>
(event schedule as of April 19, 2008)

Monitoring Software Quality with Continuous Integration by Andrew Glover

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

Tactical Continuous Integration with Hudson by Andrew Glover

This session will walk attendees through a series of iterations on a fictional Java project where an automated build system is created that facilitates compilation, testing, inspection, and deployment. This build system is then plugged into the Hudson CI server and as features are coded using Agile techniques like developer testing, attendees will ultimately see firsthand how a Continuous Integration process reduces risk and improves software quality.

Groovin' builds Gant get any easier by Andrew Glover

There's no question that Ant is the de facto standard for building Java applications; however, even its creator has acknowledged an inherent limitation with Ant's expressiveness due to its reliance on XML. Recently, the popularity of Ruby and the Rails framework has brought to focus Ruby's de facto build platform: Rake. Rake's expressiveness comes from its reliance on Ruby itself to define a DSL for software assembly. While Rake's ultimate focus is Ruby, there are a number of interesting projects that utilize expressive DSLs for building Java including Gant, which uses Groovy as a DSL format and builds upon Ant's existing cornucopia of tasks.

Easy BDD with Groovy by Andrew Glover

Behavior-driven development, or BDD, has attracted a lot of attention via RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework modeled off of the xUnit paradigm. But JBehave isn't the only framework available for Java developers-- with the advent of Groovy, new options are available for embracing BDD in the spirit of RSpec's innovative behavior based DSL.

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.

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.

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.

Beyond ACID: transactions management, in theory and practice by Brian Goetz

Transactions are the software building blocks of enterprise applications, but not all transactional systems are created equally. This talk covers the basics of what transactions are, why they are essential to building reliable enterprise software, the fundamental properties of transactions, and how transactions are supported and implemented in popular frameworks such as Java EE and Spring.

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.

Guice Dependency Injection by Brian Pontarelli

This presentation covers the latest dependency injection framework named Guice. Guice was written by the developers at Google and makes dependency injection lighter, faster and easier to write. Attendees will learn how to dependency inject their classes using Guice annotations and modules.

Dependency management by Brian Pontarelli

This talk covers the difficult subject of dependency management and uses the Savant open source framework to illustrate how to tackle some of the more difficult problems of dependency management. During this talk we'll cover the basics of dependency management, software versioning, compatibility, upgrading, and much more.

Struts 2 basics by Brian Pontarelli

This talk will cover the basics of Struts 2, the latest version of Struts and the marriage of WebWork and Struts 1. We'll be discussing the features of Struts 2 and how developers can get up and running with Struts 2.

Struts 2 convention over configuration by Brian Pontarelli

This talk focuses on how developers can create Struts 2 applications with little or no configuration using the Struts 2 Convention Plugin. This plugin leverages Struts 2 plugin system and can be dropped into any Struts 2 application. We'll cover how to add the plugin to an application and start coding Struts 2 applications without configuration.

What's Going On? : Complex Event Processing w/ Esper by Brian Sletten

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

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.

REST - Live! by Brian Sletten

You've read the articles, the books, the Ph.D. thesis and all of the meta-commentary about building RESTful APIs, but you're still not sure where to begin. This is an interactive session and has almost no slides. You should come prepared to discuss ideas and maybe pair program with me and everyone else in the room. Bring your ideas for open source projects that we might want to expose through a resource-oriented model. Bring your concerns about your domains that you are convinced don't fit this model.

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.

Rich Clients, Rich Data Part I : Linking by Brian Sletten

You hear a lot of talk about rich clients, but the richness they purport to provide is predicated on having access to rich data as well as a rich user interaction style. Without the right levels of abstraction, it is hard to address and link all of the data we have to care about these days. Additionally, the web sites that do support the notion of linking require you to publish your data into TheirSpace. Forget that. You want to be able to link publicly available information to sensitive information in YourSpace. Ever since we started doing relational joins, we've looked for ways to tie data together. The problem is, the relational model is a bit tired and doesn't move at the speed of the Net. We need schemes that integrate relational data, web pages, XML files, RSS feeds and various other sources of information.

Rich Clients, Rich Data Part II : Consuming by Brian Sletten

You hear a lot of talk about rich clients, but the richness they purport to provide is predicated on having access to rich data as well as a rich user interaction style. Without the right levels of abstraction, it is hard to address and link all of the data we have to care about these days. Additionally, the web sites that do support the notion of linking require you to publish your data into TheirSpace. Forget that. You want to be able to link publicly available information to sensitive information in YourSpace. Ever since we started doing relational joins, we've looked for ways to tie data together. The problem is, the relational model is a bit tired and doesn't move at the speed of the Net. We need schemes that integrate relational data, web pages, XML files, RSS feeds and various other sources of information.

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

Facelets by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

Seam by David Geary

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0. Two frameworks. And then you have to learn to use them together. Why do we have to learn two frameworks just to retrieve "Hello World" from a database and show it in a view. Isn't that crazy? Now you can use one framework, and use one component model. One. Isn't that nice? Seam, a framework built on JSF and EJB3.0, unifies the JSF and EJB component models. Seam is a steam roller, quickly gathering market share among JSF newbies and longtime believers alike. Come see what it's all about. **Prerequisite:** *Some knowledge of JSF is required. If you don't know what a managed bean is, for instance, then attend JSF Whirlwind before this session.*

Rich Faces by David Geary

This talk explores the RichFaces Ajax framework, which is really two frameworks: Ajax4jsf and RichFaces components. In this session you will see how to implement low-level Ajax functionality using Ajax4JSF,

and how to use high-level Ajax components from RichFaces. **Prerequisite:** *Some knowledge of JSF is required, in addition to familiarity with Ajax.*

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

Agile, Smagile: What's Working? - What's Not? by David Hussman

With the growth of agile comes the need to add a new line to the Agile Manifesto: Success over Dogma. The number of people who can say agile is growing faster than the number of people benefiting from agile practices. There are now many successful agile projects, yet there are also a growing number of projects claiming to be agile but not seeing any of the benefits agile methods provide. This session will discuss successful adoptions of agile, dumb things you can do to muck it up, and more.

Architecture and Agility Are Not Mutually Exclusive by David Hussman

Being agile does not mean living life one iteration at a time. Agile projects without a long view can run into the common design problems of the past. Planning iteration by iteration is often foolish and feeds the myth that agile projects do not think beyond a few weeks. Successful agile projects plan within iterations and across iterations. The later planning is called release planning and it is the forum where agility first engages architecture and other cross cutting concerns.

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

Although 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. **Prerequisite:** *Leading Agile Projects: Finding Your Groove in the First 4 Iterations*

Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

Build Teams, Not Products by Jared Richardson

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

10 Tips for Getting Your Project Back on Track by Jared Richardson

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

Career 2.0: Take Control of Your Life by Jared Richardson

Has your career been a random product of your manager's whims or company's needs? Never rely on your company to keep your skills current and marketable. Take control of your own career with a proven strategy.

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.

Techniques 2008 by Jared Richardson

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

Restoring Agility: Getting Your Team Back on Track by Jared Richardson

An agile team is first and foremost "a team". When that gets lost in the rush to get a product out the door, the people suffer as well as the products. It's bad for the company, but even worse for the team members. We'll learn how to defuse some of the more common problems you'll run into on dysfunctional teams.

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Powerful Metaprogramming Techniques With Groovy by Jeff Brown

Metaprogramming is a key component in building truly dynamic and flexible applications with Groovy. Groovy's metaprogramming capabilities bring great new possibilities to the table that would be very difficult or just plain impossible to write with Java alone. This session will demystify a lot of the magic that seems to be going on inside of a Groovy application. **Prerequisite:** *A Thorough Introduction To Groovy*

Grails - Agile Web 2.0 The Easy Way by Jeff Brown

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

Advanced Web Development With Grails by Jeff Brown

Grails makes web application development both fun and easy. This session dives beyond the basics to cover advanced details of Grails that bring the really exciting features to your applications. **Prerequisite:** *Grails - Agile Web 2.0 The Easy Way*

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

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.

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

7 Habits of Highly Effective Developers by Ken Sipe

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

Iteration 0 by Ken Sipe

The success of an Agile / SCRUM project is a successful start. The first interaction is often referred to as iteration 0. Other iterations have a set of stories with clear acceptance criteria which establishes the velocity of the team and its effort. What then is accomplished in iteration 0? How do we get an Agile process started.

SOA Unplugged by Mark Richards

Awareness about Service Oriented Architecture (SOA) has grown significantly in the past several years. Unfortunately, along with that growth has come a significant amount of confusion about what SOA really is. SOA has become such a ubiquitous buzzword that it now has many faces and means different things to different people. CIO's, managers, vendors, business users, architects, and developers all see SOA differently which creates a sea of confusion about what is and isn't SOA. In this highly interactive and thought provoking session we will look beyond the hype and marketure of SOA and explore SOA from an architecture and development point of view - in other words, SOA as an architecture pattern. During this session we will look at SOA use cases, services, integration, implementation, guiding architecture principles of SOA, and attempt to answer the following question: What is and isn't SOA?

Enterprise Messaging Using JMS (Part 1) by Mark Richards

The chances are good that at some point in your career you will need to use messaging to pass information between applications, subsystems, or external systems, particularly with service-oriented architecture on the rise. The Java Messaging Service (JMS) allows Java applications to implement messaging using a standard API, thereby removing the dependency on any particular messaging provider. In Part 1 of this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, and request/reply messaging. I will begin the session by going over the basics of messaging and the JMS API. Then, through interactive coding using OpenJMS I will demonstrate how to connect to the JMS provider, send messages, receive messages, and use message properties. Please note that this is a two part session.

Enterprise Messaging With JMS (Part 2) by Mark Richards

In Part 1 of the JMS session I covered messaging models, messaging basics, the JMS API, and point-to-point messaging. In this interactive code-intensive session I will cover some additional JMS topics such as browsing queues, publishing and subscribing to messages within the pub/sub model, durable and non-durable subscribers, message selectors, and message filtering. I will also discuss and demonstrate message prioritization, persistent and non-persistent messages, and finally message expiration (expiry). Note that this is Part 2 of a two-part JMS session. **Prerequisite:** *Enterprise Messaging With JMS (Part 1) or some knowledge of JMS*

EJB3 Core Specification (JSR-220) by Mark Richards

EJB3 (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will 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, dependency injection, declarative security, interceptors (aop), and Message-Driven Beans (MDB). For those of you who still like to write 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) - only the core specification.

Java Persistence: Approaching the Silver Bullet by Mark Richards

Java Persistence has come along way since the days of straight JDBC coding and custom framework development. We have at our disposal several outstanding open source frameworks such as Hibernate, Toplink, iBatis, and OpenJPA (just to name a few), and we now have a promising and emerging standards-based solution called Java Persistence API (JPA). However, all too often we find in the Java persistence space that it is a world of one-size-does-not-fit-all. We continually struggle with traditional ORM solutions like Hibernate when it comes to reporting queries, complex queries, complex relationships, and

stored procedures, and we also struggle with managing the enormous amount of SQL required for solutions such as iBATIS or JDBC-based frameworks. In this coding-intensive session we will take a detailed look at identifying and overcoming the challenges we face when using frameworks such as Hibernate, iBATIS, and JPA, and how to combine the various persistence frameworks to create an effective Java persistence solution that approaches (but of course does not reach) the silver bullet.

10 Things Every Software Architect Should Know by Richard Monson-Haefel

An effective software architect understands that every application is different and requires unique choices regarding programming language, middleware, integration, data access, user interface design, etc. Richard Monson-Haefel has distilled knowledge from his own experience and from personal interviews with the World's best software architects to define 10 principles every software architect should know in order to be effective.

Developing Rich Internet Applications by Richard Monson-Haefel

With literally hundreds of RIA products (e.g., Adobe Flash, Nexaweb, Backbase) and open source Ajax projects (e.g. Dojo, GWT, Prototype) to choose from. Picking the right RIA technology for the job requires months of research. Richard Monson-Haefel has been researching and writing about RIA alternatives for two years and has already done the research so you don't have to.

Understanding Open Source Licensing by Richard Monson-Haefel

What does GPL, LGPL, MIT, Apache licenses, copy left, and dual licensing mean? Richard Monson-Haefel explains both the legal and technical implications of the major open source licenses in plain English. He explains when and how you can use open source in the enterprise and in the development of software products and how to protect your organization from abusing open source licensing.