

New England Software Symposium

Sheraton Ferncroft

April 04 - 06, 2008

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

(event schedule as of April 3, 2008)

| Fri, Apr. 04, 2008 | | | | | | |
|--------------------|---|---|--|---|---|--|
| | North Shore B&C | North Shore A | Newburyport | Gloucester | Ipswich | Georgetown |
| 12:00 - 1:00 PM | REGISTRATION | | | | | |
| 1:00 - 1:15 PM | WELCOME | | | | | |
| 1:15 - 2:45 PM | A Thorough Introduction To Groovy Jeff Brown | Test Driven Design Neal Ford | Know your Java? Venkat Subramaniam | 10 Things Every Software Architect Should Know Richard Monson-Haefel | Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson | Give it a REST Brian Sletten |
| 2:45 - 3:15 PM | BREAK | | | | | |
| 3:15 - 4:45 PM | Powerful Metaprogramming Techniques With Groovy Jeff Brown | Evolutionary SOA Neal Ford | Caring about your Code Quality Venkat Subramaniam | Understanding Open Source Licensing Richard Monson-Haefel | 10 Tips for Getting Your Project Back on Track Jared Richardson | REST - Live! Brian Sletten |
| 4:45 - 5:00 PM | BREAK | | | | | |
| 5:00 - 6:30 PM | Agile Test Driven Development With Groovy Jeff Brown | 10 Ways to Improve Your Code Neal Ford | Design Patterns in Java and Groovy Venkat Subramaniam | Developing Rich Internet Applications Richard Monson-Haefel | Build Teams, Not Products Jared Richardson | RESTlet for the Weary Brian Sletten |
| 6:30 - 7:15 PM | DINNER | | | | | |
| 7:15 - 8:00 PM | Keynote: by Neal Ford | | | | | |

| Sat, Apr. 05, 2008 | | | | | | |
|--------------------|--|--|---|---|--|--|
| | North Shore B&C | North Shore A | Newburyport | Gloucester | Ipswich | Georgetown |
| 8:00 - 9:00 AM | BREAKFAST | | | | | |
| 9:00 - 10:30 AM | Simplifying Enterprise Applications with Spring, Part 1 Mark Fisher | Grails - Agile Web 2.0 The Easy Way Jeff Brown | FP on JVM Venkat Subramaniam | Productive Programmer: Acceleration & Automation Neal Ford | Developing Web Services Quickly using GroovyWS Mark Johnson | Restoring Agility: Getting Your Team Back on Track Jared Richardson |
| 10:30 - 11:00 AM | MORNING BREAK | | | | | |
| 11:00 - 12:30 PM | Simplifying Enterprise Applications with Spring, Part 2 Mark Fisher | Advanced Web Development With Grails Jeff Brown | DSL in Groovy Venkat Subramaniam | Productive Programmer: Canonality & Focus Neal Ford | Getting Started with BPEL Mark Johnson | Techniques 2008 Jared Richardson |
| 12:30 - 1:30 PM | LUNCH | | | | | |
| 1:30 - 3:00 PM | "Design Patterns" in Dynamic Languages Neal Ford | SOA Unplugged Mark Richards | Spring Dynamic Modules for OSGi Service Platforms Venkat Subramaniam | Configuring Spring with Annotations Mark Fisher | Promoted to Technical Lead - Now what do I do? Mark Johnson | Shippers Unite! Jared Richardson |
| 3:00 - 3:15 PM | BREAK | | | | | |
| 3:15 - 4:45 PM | Enterprise Security with Spring Mark Fisher | Java Persistence: Approaching the Silver Bullet Mark Richards | Groovy And Your Build Jeff Brown | Introduction to JRuby Neal Ford | Software Development Risk Analysis techniques Mark Johnson | Acceptance Testing Application Behavior Venkat Subramaniam |
| 4:45 - 5:30 PM | BIRDS OF A FEATHER SESSIONS | | | | | |

| Sun, Apr. 06, 2008 | | | | | | |
|--------------------|--|---|---|--|---|---|
| | North Shore B&C | North Shore A | Newburyport | Gloucester | Ipswich | Georgetown |
| 8:00 - 9:00 AM | BREAKFAST | | | | | |
| 9:00 - 10:30 AM | Regular Expressions in Java Neal Ford | Guerilla Unit Testing Part 1: TestNG with Code Coverage Howard Lewis Ship | Enterprise Messaging Using JMS Mark Richards | Agile Estimation Mark Johnson | Exploring the JavaServer Faces Ecosystem Kito Mann | Internationalization and Localization in Java David Bock |
| 10:30 - 11:00 AM | BREAK | | | | | |
| 11:00 - 12:30 PM | Building DSLs in Static and Dynamic Languages Neal Ford | Guerilla Unit Testing Part 2: The Weird and Wonderful EasyMock Howard Lewis Ship | EJB3 Core Specification (JSR-220) Mark Richards | Requirements Drive Design and Development (RDDD) Mark Johnson | Architecting JavaServer Faces Applications Kito Mann | Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock |
| 12:30 - 1:30 PM | LUNCH | | | | | |
| 1:30 - 2:15 PM | EXPERT PANEL DISCUSSION | | | | | |
| 2:15 - 3:45 PM | Enterprise Integration with Spring, Part 1 Mark Fisher | Code Metrics & Analysis for Agile Projects Neal Ford | Java EE Command Pattern Architecture Mark Richards | Introduction to Tapestry 5 Howard Lewis Ship | Simplifying JavaServer Faces Component Development Kito Mann | Intermediate Maven David Bock |
| 3:45 - 4:00 PM | BREAK | | | | | |
| 4:00 - 5:30 PM | Enterprise Integration with Spring, Part 2 Mark Fisher | Agile Project Management (With Just a Bit About Mingle) Neal Ford | Transaction Design Patterns Mark Richards | Pragmatic Patterns with Tapestry 5 IoC Howard Lewis Ship | Introduction to JBoss Seam Kito Mann | Surviving Middle Management David Bock |

New England Software Symposium

Sheraton Ferncroft

April 04 - 06, 2008

<http://www.nofluffjuststuff.com/conference/boston/2008/04/index.html>
(event schedule as of April 3, 2008)

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 with almost no slides. You should come prepared to pair program with me and everyone else in the room. We will tackle any reasonable suggestions of what might benefit from a RESTful interface. We will generate working code and explore the iterative process by which good REST-oriented APIs are created. Bring your ideas for open source projects that we might want to expose through a resource-oriented 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.

Internationalization and Localization in Java by David Bock

Internationalization and Localization in Java is easy, right? Everyone knows you just store your strings in some resource bundles, set the locale, wave your hands a little bit, and your application is good-to-go. Right? Maybe not... Java provides some great utilities to get started, but leaves you needing more when it comes to things like screen layout, cultural sensitivities, semantic differences in translation, use of color and iconography, and other issues.

Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools by David Bock

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? How many times have you been the 'new guy' on an established project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good structural guidelines and several tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

Intermediate Maven by David Bock

Maven is a build tool that does a lot, demos well, and leaves the build maintainers managing what seems like unbridled complexity. It doesn't have to be that way - Maven is driven by some strong 'build process methodology', and that complexity can become manageable by wrapping your head around it. Furthermore, you can migrate to Maven 'piecemeal', by mapping your existing ant build to the Maven Lifecycle and calling your existing Ant tasks - you can decide to sip the Maven kool-aid. Ideally, a build tool should be so simple and approachable that it fades into the project background and allows anyone to maintain it. Unfortunately, Maven's power comes at the expense of this ideal - Maven's philosophy is more like "the build process is so important that the people maintaining it should be steeped in the ways of Maven". This talk will give you the exposure you need without elevating The Maven Way to a religion.

Surviving Middle Management by David Bock

Most good developers eventually have the opportunity to be managers. Whether they call you the "project manager", "Technical Lead", "Lead Developer", or some other classic middle-management title, you become the 'goto' guy between management and developers. You're the guy who is expected to keep the project

in-line, track a schedule, and occasionally answer the question "How's it going?", and perhaps still contribute at a technical level. So how do you do that?

Guerilla Unit Testing Part 1: TestNG with Code Coverage by Howard Lewis Ship

Part one (of two) covers the TestNG unit testing framework, and shows how it integrates with tools such as Emma or Cobertura (for code coverage) and Selenium (for integration testing).

Guerilla Unit Testing Part 2: The Weird and Wonderful EasyMock by Howard Lewis Ship

In part two (of two) we go in depth on EasyMock, the weird and wonderful tool for creating mock objects on the fly. We'll do a good bit of live coding as we examine how to use, tame and extend this powerful tool.

Introduction to Tapestry 5 by Howard Lewis Ship

Tapestry 5 is a complete rewrite of Tapestry from the ground up. It takes everything good about Tapestry and cranks the volume up to eleven, while removing the frustrating parts of using Tapestry. This session takes the wraps off this new and innovative technology, showing off important new features such as live class reloading (the ability to change your Java classes and continue using the application without interruption or redeployment), the simplified coding model, and the total lack of XML. This session is of interest to those already using Tapestry 4, and those new to Tapestry and ready to jump on the bandwagon.

Pragmatic Patterns with Tapestry 5 IoC by Howard Lewis Ship

Everyone likes the Gang of Four design patterns, but it's not always clear just how to make use of them in your day to day coding efforts. Hidden inside Tapestry 5 is an Inversion of Control (IoC) container that is structured around several common patterns (Chain of Command, Strategy, Facade and Filter Chain will be covered). This isn't academic navel-gazing ... this is about leveraging the common patterns so that you can write code you can easily test, and about creating frameworks and toolkits that can be easily extended. We'll see how Tapestry uses these patterns, and go from there into how you can apply the same techniques to your own projects, resulting in better, cleaner, more testable code.

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.

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.

Build Teams, Not Products by Jared Richardson

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

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.

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.

Shippers Unite! by Jared Richardson

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

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.

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*

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.

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*

Groovy And Your Build by Jeff Brown

There are numerous roles that Groovy may play in your build process to greatly simplify the management of the build while bringing more capabilities. This session will detail a lot of the things that Groovy can do to improve your build and lessen the amount of effort you spend on your build. **Prerequisite:** *A Thorough Introduction To Groovy*

Exploring the JavaServer Faces Ecosystem by Kito Mann

This session examines the ecosystem that is growing around JavaServer Faces. **Prerequisite:** *Basic familiarity with web application development in Java.*

Architecting JavaServer Faces Applications by Kito Mann

Over the past few years, a lot of time has been spent explaining what JSF is, and how different pieces of it work. However, little attention has been given to the process of architecting applications. This makes JSF architecture seem like a black art, since there are so many possible approaches to the application's architecture. **Prerequisite:** *Experience with JavaServer Faces.*

Simplifying JavaServer Faces Component Development by Kito Mann

The benefits of using JavaServer Faces UI components to rapidly construct complex, interactive user interfaces have become quite clear over the past couple of years. However, the standard process for developing these UI components is currently quite tedious. Fortunately, there are better solutions available.

Introduction to JBoss Seam by Kito Mann

JBoss Seam is a popular open-source application framework for Java Platform, Enterprise Edition (Java EE) 5.0. For web application developers, a significant benefit of Seam is that it greatly enhances JavaServer Faces technology. This session explains key Seam features such as tight integration with EJB3, Hibernate and JPA integration, conversations, RESTful web pages, and so on. **Prerequisite:** *Basic understanding of Java web application development. Familiarity with Dependency Injection/Inversion of Control, JSF, and JPA/Hibernate persistence is a plus.*

Simplifying Enterprise Applications with Spring, Part 1 by Mark Fisher

Developing enterprise applications isn't easy. You not only have to worry about constantly evolving business logic, but also need to address infrastructure concerns ranging from transaction management and security to manageability and integration with diverse external applications. Spring, the most popular lightweight

enterprise application framework, comes to the rescue by simplifying the common needs of enterprise applications. This session (part 1 of 2) presents the core concepts of the Spring Framework.

Simplifying Enterprise Applications with Spring, Part 2 by Mark Fisher

This session (part 2 of 2) will cover advanced concepts in the Spring framework. While the core concepts in the first session will get you started with Spring, the advanced concepts in this session will help you be more effective at developing Spring-based applications.

Configuring Spring with Annotations by Mark Fisher

In this session, we will take a deep-dive into annotation-based dependency injection with Spring 2.5. You will learn how to combine annotation and XML formats, how to customize component scanning, and how to leverage Java 6 annotations within a Spring application. Since there is no "one size fits all" solution to application configuration, we will wrap up the discussion with general guidelines to consider when employing this approach.

Enterprise Security with Spring by Mark Fisher

Spring Security (formerly known as 'Acegi') enables self-contained, consistent, and extensible solutions for securing your applications. Version 2.0 provides major enhancements including a domain-specific XML namespace, convention-based defaulting, and annotation support. This provides a significantly simpler experience for developers while still supporting the same degree of flexibility.

Enterprise Integration with Spring, Part 1 by Mark Fisher

The first part of this two-part session will focus on the essentials of Enterprise Integration with Spring. The discussion will begin with a whirlwind tour of the enterprise integration support libraries in the Spring Framework core. We will then explore message-driven architecture concepts in light of well-established design principles such as loose coupling and separation of concerns.

Enterprise Integration with Spring, Part 2 by Mark Fisher

The second part of this two-part session will introduce Spring Integration, a new addition to the Spring portfolio. We will begin with a high-level overview of Enterprise Integration Patterns as catalogued in the highly influential book of the same name. We will then embark on a demo-driven exploration of Spring Integration to see how it enables the development of applications based on those patterns. **Prerequisite:** *Enterprise Integration with Spring, Part 1*

Developing Web Services Quickly using GroovyWS by Mark Johnson

This session will explore GroovyWS as a tool to quickly produce and or consume a web service. Web Service testing becomes much easier without the need to purchase expensive testing tools using the GroovyWS framework.

Getting Started with BPEL by Mark Johnson

With all of these web services becoming available there is an increasing need for tools to pull together multiple web services into one composite service. BPEL is an up and coming approach to orchestrating a workflow consisting of Web Service calls.

Promoted to Technical Lead - Now what do I do? by Mark Johnson

The job of tech lead/development manager seemed so easy before. Now you are in the hot seat. You are responsible for getting the new product/project out the door on time and under budget. Your team, management, and finance departments are counting on you to make the right decisions. So what do you do?

Software Development Risk Analysis techniques by Mark Johnson

Once you leave academic "hello world" projects, software development is full of unknowns which result in the high rate of project failure we see too often in industry. Reasons for a project failure will vary based on the stakeholder interviewed. This session will provide a software development risk framework and examples you can apply in your projects to reduce or at least soften the impact of failure.

Agile Estimation by Mark Johnson

As developers we dread when management requests a project estimate. Typically, you do not have the opportunity to understand all the requirements, the team composition is unknown, and you have been given until tomorrow end of day to produce an estimate.

Requirements Drive Design and Development (RDDD) by Mark Johnson

Validate that requirements are not missed during the design and development process by creating Requirements document test fixtures to clarify and validate the requirements between the end users, business analysts, architects, and developers early in the project.

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?

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.

Enterprise Messaging Using JMS 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 this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, message headers, request/reply, and pub/sub messaging. In the first part of this session I will go over the basics of messaging and the JMS API. Then, in the second part of the session, I will abandon the presentation slides and through interactive coding using OpenJMS I will demonstrate how to send messages, receive messages, use message properties, and how to use the different message types (Text, Object, Map, Bytes, and Stream). At the close of the session I demonstrate how to use the pub/sub model and show how to code durable and non-durable subscribers.

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 EE Command Pattern Architecture by Mark Richards

Tired of dealing with EJBs but cannot use other frameworks like Spring? How would you like to replace all of your remote Stateless Session Beans with POJOs and still access them remotely within Java EE? By using the Java EE Command Pattern we can write EJBs as POJOs and solve many of the issues facing

EJB, including testability, configuration complexity, and performance, and still remain within the boundaries of the Java EE container. The Java EE Command Pattern is a simple pattern that can significantly reduce the complexity of large-scale Java EE enterprise applications. In this session we will explore the numerous issues facing a typical EJB architecture and learn how the use of the Java EE Command Pattern can solve these issues. We will walk through the different design alternatives and see how the command pattern is implemented in both EJB3 and in Spring. Through interactive coding examples you will learn what components make up the Command Pattern framework and what simple coding changes are required to convert a complex remote EJB-based application to a much simpler remote POJO-based application.

Transaction Design Patterns by Mark Richards

Most web-based applications rely solely on the database to manage transactions, thereby freeing the developer from having to worry about transaction management. While this works in some circumstances, there are times when the use of transactions is vital to the integrity and operations of an application and its corresponding data. In this session I will demonstrate through real-world coding examples why transactions are such a critical part of the application development process. I will review the basics of both programmatic and declarative transactions, then introduce three transaction design patterns and explain when they should be applied, how to use them, and what problems they solve. By the end of this session you will see that by using transaction design patterns you can build an effective transaction management strategy for your application with very little effort.

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

Evolutionary SOA by Neal Ford

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

10 Ways to Improve Your Code by Neal Ford

No one writes perfect code: even the best developers fall into bad habits and traps. This talk illustrates blind spots and helps you write better code.

Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

Productive Programmer: Acceleration & Automation by Neal Ford

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

Productive Programmer: Canonicity & Focus by Neal Ford

Getting work done in modern office environments is a daunting task. This session tackles 2 of the things that drag down developer productivity: lack of focus and creeping repetition.

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

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.

Regular Expressions in Java by Neal Ford

Regular expressions should be an integral part of every developer's toolbox, but most don't realize what an important topic it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

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.

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.

Agile Project Management (With Just a Bit About Mingle) by Neal Ford

You can read books about Agile projects, but you must consult real-world experience to really understand the dynamics of agile project management. This session discusses agile management topics including estimation, project tracking, and useful metrics (and how to obtain them). And just a little about Mingle, the agile project tracking tool from ThoughtWorks.

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.

Enterprise JavaBeans 3.0 by Richard Monson-Haefel

Richard Monson-Haefel, author "Enterprise JavaBeans 3.0" (O'Reilly), and a member of the EJB 2.1 and 3.0 expert groups that defined the specification, provides a no non-sense deep dive into how EJB 3.0 works, how best to develop EJB 3.0 applications, and when and where EJB 3.0 is appropriate.

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.

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.

Know your Java? by Venkat Subramaniam

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.

Caring about your Code Quality by Venkat Subramaniam

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.

Design Patterns in Java and Groovy by Venkat Subramaniam

You're most likely familiar with the Gang-of-four design patterns and how to implement them in Java. However, you wouldn't want to implement those patterns in a similar way in Groovy. Furthermore, there are a number of other useful patterns that you can apply in Java and Groovy. In this presentation we'll look at two things: How to use patterns in Groovy and beyond Gang-of-four patterns in Groovy and Java.

FP on JVM by Venkat Subramaniam

Functional Programming Languages (FPLs) have been around for a long time. A lot of features that we get excited about in dynamic languages are common place in FPLs. FPLs are gaining importance due to various changes in our industry. What's exciting is that you can use them on the JVM. In this presentation we will dig into the details of what makes FPLs so interesting and look at ways to use them on the JVM?in your Java projects.

DSL in Groovy by Venkat Subramaniam

DSL or Domain Specific Languages focus on a domain or problem at hand. They're expressive, but their restricted scope keeps them simple and small from the user point of view. However, designing them is not easy. In this presentation we will explore the features of Groovy and show how they can be used to create DSLs.

Spring Dynamic Modules for OSGi Service Platforms by Venkat Subramaniam

OSGi is a specification that helps with versioning Java modules at runtime. Spring helps with dependency injection of Java components and beans. Spring has embraced OSGi and allows you to integrate different OSGi implementations into your Spring applications. In this presentation we will look at the rational for mixing Spring and OSGi and look at code examples of the same.

Acceptance Testing Application Behavior by Venkat Subramaniam

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

Spring Dynamic Modules for OSGi Service Platforms by Venkat Subramaniam

OSGi is a specification that helps with versioning Java modules at runtime. Spring helps with dependency injection of Java components and beans. Spring has embraced OSGi and allows you to integrate different OSGi implementations into your Spring applications. In this presentation we will look at the rational for mixing Spring and OSGi and look at code examples of the same.