

Central Ohio Software Symposium

Crowne Plaza Dublin

July 25 - 27, 2008

<http://www.nofluffjuststuff.com/conference/columbus/2008/07/index.html>
(event schedule as of July 26, 2008)

Fri, Jul. 25, 2008					
	Ballroom C-E	Ballroom A	Ballroom B	Waterford	Limerick
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	JSF 2.0 Preview Kito Mann	10 Things Every Software Architect Should Know Richard Monson-Haefel	10 Tips for Getting Your Project Back on Track Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Advanced Rules Programming with Drools Brian Sam-Bodden	Building Enterprise Applications with JavaServer Faces and Spring Kito Mann	Developing Rich Internet Applications Richard Monson-Haefel	Techniques 2008 Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Grails for (Recovering) Struts Developers: A Groovy Alternative Scott Davis	Understanding Open Source Licensing Richard Monson-Haefel	Simplifying JavaServer Faces Component Development Kito Mann	10 ways to use Hibernate effectively Brian Sam-Bodden	Build Teams, Not Products Jared Richardson
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Jared Richardson				

Sat, Jul. 26, 2008					
	Ballroom C-E	Ballroom A	Ballroom B	Waterford	Limerick
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Introduction to JBoss Seam Kito Mann	Refactoring JavaScript Stuart Halloway	Guerilla Unit Testing Part 1: TestNG with Code Coverage Howard Lewis Ship
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	Shippers Unite! Jared Richardson	Git control of your source Stuart Halloway	Making EJB Meaningful with JBoss Seam Joseph Nusairat	Guerilla Unit Testing Part 2: The Weird and Wonderful EasyMock Howard Lewis Ship
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	YSlow: Building Your Website for Speed Scott Davis	Software Development Risk Analysis techniques Mark Johnson	How to Fail with 100% Code Coverage Stuart Halloway	Meta-programming in Groovy Brian Sam-Bodden	Introduction to Tapestry 5 Howard Lewis Ship
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Real World JSON Scott Davis	The Software Development Manager's Dashboard Mark Johnson	Testing EJB3 Joseph Nusairat	JVM Language Shootout: Clojure, JRuby, and Scala Stuart Halloway	Pragmatic Patterns with Tapestry 5 IoC Howard Lewis Ship
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION				

Sun, Jul. 27, 2008					
	Ballroom C-E	Ballroom A	Ballroom B	Waterford	Limerick
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Introduction to REST: What can we learn from it? John Heintz	Getting to Acceptance: Validating your requirements with FitNesse Mark Johnson	New Features in Spring Web Keith Donald	Basic JPA & JPAQL Pratik Patel
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Developing Web Services Quickly using GroovyWS Mark Johnson	Glassbox: Open Source Java Monitoring and Troubleshooting John Heintz	The Agile Product Owner David Bock	Spring Web Flow 2 Deep Dive Keith Donald	Enterprise JPA - Tips and Tricks for JEE5 Persistence Pratik Patel
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Estimating vs. Guessing - How Agile Teams Estimate Their Work David Bock	The Quest for the Holy Grails Joseph Nusairat	Adding Behavior to Java Annotations John Heintz	Decorating Web Pages with Ajax using Spring JavaScript Keith Donald	Getting Started with BPEL Mark Johnson
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Intermediate Maven David Bock	GIS for Web Developers: Adding Where to Your Application Scott Davis	Tool support for Agile Databases: Introducing Liquibase John Heintz		Promoted to Technical Lead. Now what do I do? Mark Johnson

Central Ohio Software Symposium

Crowne Plaza Dublin

July 25 - 27, 2008

<http://www.nofluffjuststuff.com/conference/columbus/2008/07/index.html>
(event schedule as of July 26, 2008)

Beginning Drools - Rule Engines in Java by Brian Sam-Bodden

Drools is an open source pure-Java implementation of a forward chaining rules engine. Drools can be used in a J2SE or J2EE application and allows you to express rules programatically or by building domain specific rule languages. Learn how Business Rules with Drools can make your Java applications more flexible and robust.

Advanced Rules Programming with Drools by Brian Sam-Bodden

In this session you'll learn some of the more advanced features of Drools; a pure-Java Rule Engine. This session will walk through the construction of an advanced Rules application covering such topics as: - Fine control and monitoring of a Working Memory session - Using Decision Tables - Advanced Rule Language Features - Building Domain Specific Languages - Managing your Rules **Prerequisite:** *Beginning Drools*

10 ways to use Hibernate effectively by Brian Sam-Bodden

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

Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden

Learn how to build featured rich applications using the Eclipse Rich Client Platform. The Eclipse platform is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

Boosting Programmer productivity with Mylyn by Brian Sam-Bodden

Mylyn is a task-focused toolkit for the Eclipse IDE that allows developers to focus on tasks in a way that they never been able to do before. Mylyn eliminates the constant context switching produced by typical ways IDEs are used. No more scrolling/browsing/searching/tagging/sending emails with progress updates... Mylyn provides a new way of working that allows you to focus on specific tasks by reducing information overload. Mylyn also provides a framework for integrating with the most commonly usage task tracking systems and version control systems. In this talk you'll learn how Mylyn can boost your productivity as a Java developer by letting you get the most out of your IDE.

Meta-programming in Groovy by Brian Sam-Bodden

This session explores some of the programming techniques that a powerful dynamic language enables, in particular meta-programming or the art of writing code that writes code. Meta-programming techniques are being used extensively in many successful frameworks based on dynamic languages such as Rails, Grails and countless others. Learn how you can use meta-programming in Groovy to improve and streamline your Java applications.

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.

The Agile Product Owner by David Bock

Agile software development isn't just about the development team or managers... the customer has an active role too. The customer should be prioritizing the stories in each release, potentially working onsite in constant contact with the development team, and even participating in daily status meetings. Done well,

the customer's presence has a positive influence on the development iteration. Done poorly, the customer detracts from the team's focus. So how do you be the customer of an agile team? How do you teach someone to be that customer?

Estimating vs. Guessing - How Agile Teams Estimate Their Work by David Bock

Estimating is regarded as little more than 'educating guessing', but so much can hang on the quality of those estimates. With good estimates we can set clear expectations for project delivery, but with bad estimates we can run over schedule and over budget, or worse. We often estimate when we know the least about the work that needs to get done - so how can we make the best of what is potentially a bad situation?

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.

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.

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.

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.

Build Teams, Not Products by Jared Richardson

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

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.

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.

Shippers Unite! by Jared Richardson

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

Introduction to REST: What can we learn from it? by John Heintz

REST is a description of how the Web works, what use is that to developers just trying to build or integrate applications? This presentation introduces REST, explains the key differences/constraints, and then highlights how these concepts can improve key parts of application and service development: * scalability, integration, evolvability

Glassbox: Open Source Java Monitoring and Troubleshooting by John Heintz

In this session you will learn about the Glassbox open source troubleshooting and monitoring tool. Glassbox enable detection of common application problems such as database failures, slow operations, thread contention, and excessive distributed calls. Glassbox enables low overhead monitoring and troubleshooting without needing to "bake in" instrumentation up front.

Adding Behavior to Java Annotations by John Heintz

Java's Annotations provide a way to add data to program elements. Annotations are used to configure containers, describe persistence configuration, set security roles, and are defined by nearly every recent JSR standard. This presentation explains the processing options available for consuming Annotations and demonstrates the techniques with live code demonstrations.

Tool support for Agile Databases: Introducing Liquibase by John Heintz

This presentation introduces and demonstrates Liquibase: a new Java tool to support automating database refactoring and deployment.

Making EJB Meaningful with JBoss Seam by Joseph Nusairat

JBoss Seam is one of Java's ways of providing agile development to enterprise Java applications. Seam is designed to take away the headache of linking JSF, EJB3, AJAX, and jBPM by using interception directed by user defined annotations. The end result is smooth and easily manageable code without the plumbing usually required for functionality. This session will take you through the process of how Seam works, creating a sample application, and what Seam can buy you for enterprise setup including the pros and cons.

Testing EJB3 by Joseph Nusairat

Vital to any development is the ability to test that application. Most frameworks have their very own testing suites, EJB3 is no exception and there are a variety of testing apparatuses. We will examine testing with JBoss Embedded, EJB3Unit and mocking with Easy Mock and JMocks. **Prerequisite:** Knowledge of EJB3.

The Quest for the Holy Grails by Joseph Nusairat

This presentation is designed to give users an introduction to what Grails has to offer. The presentation will show how fast one can get up and running with Grails.

New Features in Spring Web by Keith Donald

Spring has a number of interesting modules for web application development, including Spring Web MVC, Spring Web Flow, Spring JavaScript, and Spring Faces. This session will provide an overview of these modules and show how they relate to one another. By the end of this session, you'll understand how Spring

simplifies the development and deployment of rich web applications. You'll also gain a glimpse into the roadmap for Spring Web 3.0.

Spring Web Flow 2 Deep Dive by Keith Donald

Web Flow is a Spring Web MVC extension that allows you to define Controllers using a higher-order domain-specific-language. This language is designed to model user interactions that require several requests into the server to complete, or may be invoked from different contexts. This session dives deep into the features of the Web Flow definition language, and illustrates how to use it to create sophisticated controller modules.

Decorating Web Pages with Ajax using Spring JavaScript by Keith Donald

Spring JavaScript is a JavaScript abstraction framework that allows you to progressively enhance a web page with behavior. The framework consists of a public JavaScript API along with an implementation that builds on the Dojo Toolkit. Spring.js aims to simplify the use of Dojo for common enterprise scenarios while retaining its full-power for advanced use cases. Come to this session to learn to use Spring.js and Dojo to create compelling user interfaces.

JavaServerFaces: The Biggest Loser of Java Web Frameworks? by Keith Donald

The Biggest Loser is a popular fitness TV show where severely overweight participants undergo radical transformations to get their lives back on track. Some might say JavaServerFaces has the reputation of being the severely overweight web framework of Java. Is it possible for JSF to shed the pounds and transform itself in the eyes of Java developers? Come to this session to find out.

JSF 2.0 Preview by Kito Mann

JavaServer Faces, the standard Java web development framework, has gained quite a few fans and detractors over the past few years. Regardless of the camp, most agree that the framework can improve. JSF 2.0, currently under development through the Java Community Process, aims to be a dramatic leap forward for the framework. **Prerequisite:** *familiarity with JSF*

Building Enterprise Applications with JavaServer Faces and Spring by Kito Mann

For developers who are currently using Spring and JavaServer Faces together, this session explains how to handle common application development concerns such as conversational scope, transaction management, and application partitioning.

Simplyfing 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. **Prerequisite:** *Basic understanding of JSF.*

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.*

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.

The Software Development Manager's Dashboard by Mark Johnson

Development teams are increasingly scattered all over the world, often a development manager now days will never meet the entire team face to face.

Getting to Acceptance: Validating your requirements with FitNesse by Mark Johnson

How do you know when you are "DONE" and the assignment is complete? Well of course you are done when your requirements are complete. But it always happens that your interpretation differs from the customer/management's interpretation.

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

You have just received the much desired promotion to Technical Team Lead The team is waiting your direction. You What should you do now?

Basic JPA & JPAQL by Pratik Patel

Doing basic Object-to-Relational Mapping is fun and easy with JPA. Annotate your persistent classes, define a couple of configuration parameters, and you're off and running. This session starts with a basic object model and adds persistence using annotations. Learn how to do mappings for your object model for simple and complex relationships. Also learn how to map Java5 constructs like Enumerations. Unit testing with JPA can be tricky. Where do you use mock objects? How can I structure my unit tests to exercise my DAO's effectively? How do I unit test JPAQL? Do I need to enhance or can I use a LoadTimeWeaver in my unit tests? This presentation will show, using live code examples, how to effectively unit test JPA components so developers can have confidence in the code they build using JPA.

Enterprise JPA - Tips and Tricks for JEE5 Persistence by Pratik Patel

As with many technologies, the basics are easy. The hard part comes when the developer needs to do sophisticated integration, development, and testing as part of an enterprise application. A large enterprise application requires the developer to think of issues that affect the development, scalability and robustness of the application. This presentation will cover the advanced topics described below. A large enterprise application often will have several sub-projects that each contain their own JPA persistence unit. This opens up a number of questions around how to organize the persistence units and how the code between sub-projects should interoperate. Developers will gain insight into these issues and will see a couple of solutions using live code examples.

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.

Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

Grails for (Recovering) Struts Developers: A Groovy Alternative by Scott Davis

Struts enjoys an unprecedented marketshare in the Java web development space -- 60%-70% according to most surveys. As newer, modern web frameworks come to the scene, very little attention is paid to the real costs of migrating an existing Struts application. This talk shows you ways to mix Groovy into a legacy Struts application, dramatically reducing both the lines of code and the complexity. We'll also introduce you to Grails (a Groovy-based web framework) whose URL-mapping capabilities allow it to replace your Struts application without breaking legacy URLs.

YSlow: Building Your Website for Speed by Scott Davis

How optimized is your website? YSlow, a FireFox/FireBug plugin, doesn't pull any punches. It gives any website an A, B, C, D, or F rating based on 14 individual analysis points. You'll be amazed (or depressed) at what YSlow thinks of your site. In this talk, we'll walk through these points step by step, learning what Yahoo! (the creator of this utility) does to keep its web properties running as quickly as possible.

Real World JSON by Scott Davis

JavaScript Object Notation is becoming a familiar delivery platform for Web 2.0 content. JSON gives you all of the flexibility of a RESTful web service without the hassle of trying to deal with deeply nested, complex XML in a language that is conspicuously lacking in native XML support. In this talk, we look at popular websites (like Yahoo!) that offer JSON output. We look at client-side JavaScript code that effortlessly consumes JSON in the browser. We even look at ways to easily generate JSON from Java Servlets (using JSON.org libraries) and the native support for JSON that Grails offers out of the box.

GIS for Web Developers: Adding Where to Your Application by Scott Davis

Based on the book GIS for Web Developers, this talk demonstrates how you can build your own Google Maps in-house using nothing but open source software. The Portland, Oregon Transit Authority recently migrated from a proprietary web mapping solution to the suite of 100% free and open source software discussed in this book. We look at Java-based clients, Java-based servers, and everything in between. We also discuss integrating free, public domain data from sources like the US Census Bureau and the USGS. If you're looking for real-world examples of AJAX in use, you'll find it here. If you're looking for real-world examples of web services in use, you'll find it here.

Refactoring JavaScript by Stuart Halloway

The rise of Ajax and Rich Web Applications, plus the success of dynamic languages, has caused people to revisit the JavaScript language. Now that we take JavaScript seriously as a language, it is time to get serious about the quality of JavaScript code, through refactoring. In this talk, we will approach refactoring JavaScript in three phases: Test first, then refactor. Bring JavaScript code under test, so that you can refactor with confidence. Refactoring 101. Explore some important refactorings: composed method, extract method,

introduce named parameter, and extract object Common problems. Work through three problems endemic to legacy JavaScript code: making JavaScript unobtrusive, refactoring to prototype-based inheritance, and refactoring to functional style.

Git control of your source by Stuart Halloway

Git is not the next step in evolution of centralized source control, following in the footsteps of cvs, svn, etc. These tools provide centralized history of deltas, where git provides distributed history of trees of content. In this talk, you will see the advantages of the git approach: Incredible speed. Local, disconnected operation. Source control workflow customized to your team. Centralized, distributed, or layered, you can build it with git. Cheap and easy branching, tagging, and merging. Editing and refactoring your commits.

How to Fail with 100% Code Coverage by Stuart Halloway

Over the last few years, we have taken dozens of projects to 100% coverage, and there are still plenty of things that can go wrong. We will look at examples the various problems, and show how to prevent them from infecting your project.

JVM Language Shootout: Clojure, JRuby, and Scala by Stuart Halloway

In this talk, we will explore and compare three of the most interesting new JVM languages: Clojure, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all three of them succeed in this mission. However, these languages have very different design goals. We will explore these differences, and help you decide when and where these languages might fit into your development toolkit.