



Come to the Ü for an Extreme Tech Experience

Über Conf will take place July 12 - 15, 2011 in Denver, CO. This event will focus on the best practices, new languages, and latest advancements on the Java Platform.

This is an exciting time of innovation and change. Java is not just a language. Java is a technology platform and ecosystem. **Über Conf will educate developers and explore the powerful languages and tools which are changing the way we create software using the Java Platform.**

» Technology Deep Dive

There are no beginner sessions here. This is your opportunity to go beyond the basics and master critical skills. We expect that you are a **competent developer who is ready to solve problems** using today's best tools and practices.

» Agile Practices that Work

Software is a difficult industry with high rates of failure. To create winning teams, we embrace principles laid out by the *Agile Manifesto*. Speakers at *Über Conf* emphasize and present on topics such as: **Test Driven Development, Continuous Integration, Code Quality Measurements, Code Smells, Team Building, and Customer Collaboration.**

» Learn from the Best

Über Conf will bring together many of the industry's best project leaders, developers, authors, and trainers.

» Full Day iOS and Android Workshops

The market for mobile development exploded last year! This is the time to jump in to mobile development. *Über Conf* will offer full day iOS and Android workshops. These workshops will bring you up to speed and get you coding right away. In addition, there will be a full track of mobile sessions throughout the event.

» Hands on Half Day Workshops

In addition to the mobile development workshops, numerous other hands-on workshops will be offered. Bring your laptop and be ready to code. These sessions, led by project experts, will get you started coding right away. Details will be announced in the coming months.

» All Inclusive Travel Packages

Let us handle your travel arrangements. We have great package deals available including: conference registration, 4 nights lodging, and airfare within the Continental US.

For more information please visit <http://uberconf.com/> or email jzimmerman@nofluffjuststuff.com

Define Über

- 1 : a superlative example of its kind
- 2 : to an extreme degree



Topics at Über Conf

- Android and iPhone
- HTML5
- NoSQL
- Cloud Computing
- Hadoop
- Automation Tools
- Gradle
- JVM Internals, Java 7
- Languages on the JVM: Groovy, Scala, JRuby, Clojure
- ROA
- Spock, GEB
- Agility

Featured Speakers



Jez Humble
Author of 'Continuous Delivery'



Terry Ryan
Author of 'Driving Technical Change'



Hamlet D`Arcy
Sr. Java/Groovy Developer, Groovy Committer



Hans Dockter
Founder of Gradle and CEO of Gradle Inc.

ÜberConf
Westin Westminster
July 12 - 15, 2011

Tue, Jul. 12, 2011										
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
8:00 - 9:00 AM	EARLY REGISTRATION: iOS & ANDROID WORKSHOPS ATTENDEES ONLY - WESTMINSTER BALLROOM FOYER									
9:00 - 6:00 PM	Fundamentals of iOS Apps Development (day long) Venkat Subramaniam		Android Workshop - Full Day Ted Neward							
5:00 - 6:30 PM	MAIN UBERCONF REGISTRATION - WESTMINSTER BALLROOM FOYER									
6:30 - 8:30 PM	DINNER/KEYNOTE - WESTMINSTER BALLROOM 3/4									
7:30 - 8:30 PM	Keynote: by Terry Ryan									
8:30 - 10:30 PM	OPENING NIGHT OUTDOOR RECEPTION - SOUTH COURTYARD									

Wed, Jul. 13, 2011										
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
7:00 - 8:00 AM	5K FUN RUN & POWER WALK - MEET IN LOBBY									
7:30 - 8:30 AM	BREAKFAST & LATE REGISTRATION - WESTMINSTER BALLROOM 3/4									
8:30 - 10:00 AM	Programming Concurrency (Full Day) Venkat Subramaniam	Game Theory and Software Development Matthew McCullough	HTML5: The JavaScript Parts Tim Berglund	Continuous Delivery Part I: Value proposition, the deployment pipeline, automated tests, CI Jez Humble	Agile.next Neal Ford	Architectural Choices around Open Source Solutions Alex Antonov	Effective Groovy Hamlet D'Arcy	What's new in Spring Craig Walls	Strategic Design Using DDD Paul Rayner	Building Workflow Applications with StonePath David Bock
10:00 - 10:30 AM	MORNING BREAK									
10:30 - 12:00 PM	Programming Concurrency (Full Day) (continued) Venkat Subramaniam	Monitoring 10 Critical Code Quality Metrics with Sonar Matthew McCullough	NoSQL Smackdown! Tim Berglund	Continuous Delivery Part I: Value proposition, the deployment pipeline, automated tests, CI (continued) Jez Humble	Emergent Design Neal Ford	Code Generation on the JVM: Writing Code that Writes Code Hamlet D'Arcy	Über Groovy Dave Klein	NoXML: Spring for XML-Haters Craig Walls	Using DDD Patterns for Supple Design Paul Rayner	Building Maintainable Javascript with Coffeescript David Bock
12:00 - 1:00 PM	LUNCH & OUTDOOR BREAK - WESTMINSTER BALLROOM 3/4 & NORTH COURTYARD									
1:00 - 1:30 PM	Keynote: Billy Williams									
1:30 - 3:00 PM	Programming Concurrency (Full Day) (continued) Venkat Subramaniam	Cassandra: Radical NoSQL Scalability Tim Berglund	Designing RESTful Services Ian Robinson	Continuous Delivery Part II: components, going live, agile infrastructure, databases, organisational transformation Jez Humble	JBoss Drools: Rule Engine Development in Java Brian Sam-Bodden	Cryptography on the JVM: Boot Camp Matthew McCullough	Über Groovy (continued) Dave Klein	Behavior Driven Development with Client Side Javascript James Carr	New Ideas for Old Code Hamlet D'Arcy	Introduction to Lean-Agile Software Development Paul Rayner
3:00 - 3:15 PM	BREAK									
3:15 - 4:45 PM	Programming Concurrency (Full Day) (continued) Venkat Subramaniam	Intro to Hadoop MapReduce - Indepth Chris Wensel	Functional Thinking Neal Ford	Continuous Delivery Part II: components, going live, agile infrastructure, databases, organisational transformation (continued) Jez Humble	JBoss Drools: Rule Engine Development in Java (continued) Brian Sam-Bodden	Simpler Cryptography with 3 JVM Libraries Matthew McCullough	Effective Java Reloaded Matt Stine	NodeJS Bootcamp James Carr	Gaelyk: Cloud-Based Apps With Groovy Tim Berglund	Measure for Measure - Lean Principles for Effective Metrics and Motivation Paul Rayner
4:45 - 5:00 PM	BREAK									
5:00 - 6:30 PM	Java Web Application Security: Develop. Penetrate. Protect. Relax. Matt Raible	Hadoop Architecture - In Depth Chris Wensel	Effective Java Reloaded (continued) Matt Stine	Hands-on Scala 1/2 Day Workshop Venkat Subramaniam	The Future of Java Enterprise Testing Dan Allen	Agile Engineering Practices Neal Ford	Pragmatic Architecture Ted Neward	NodeJS Bootcamp (continued) James Carr	Gaelyk Workshop Part I Tim Berglund	Introducing Spring Roo: From Zero to Working Spring Application in Record Time Craig Walls
6:30 - 8:30 PM	DINNER/PANEL - WESTMINSTER BALLROOM 3/4									
8:30 - 10:00 PM	Developer Productivity Power Ups on Mac OSX Matthew McCullough	Cascading and Common Big Data Problems Chris Wensel	Software Craftsmanship: Positioning, Patterns and Practices Peter Bell	Hands-on Scala 1/2 Day Workshop (continued) Venkat Subramaniam	Completing the circle - Automated web tests as a team communication tool John Smart	Build Your Own Technology Radar Neal Ford	Architectural Kata Workshop Ted Neward	Joda Time and a Brief History of the World Daniel Hinojosa	The Groovy Ecosystem Andres Almiray	Spring Roo Workshop Craig Walls

Thu, Jul. 14, 2011										
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
8:00 - 9:00 AM	BREAKFAST - WESTMINSTER BALLROOM 3/4									
								Thinking In Git	JRuby in Depth	

Thu, Jul. 14, 2011

	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
9:00 - 10:30 AM	REST in Practice - Full Day Workshop on Web-based Distributed Systems Ian Robinson and Jim Webber	Design for the Developer Terry Ryan	Applying Patterns: How to Spot Problem Code and What To Do About It Howard Lewis Ship	OSGi Demystified Kirk Knoernschild		Rapid Prototyping w/Solr Erik Hatcher	Spring MVC Workshop Craig Walls	Matthew McCullough	Neal Ford	Metrics for steering your projects to success David Bock
10:30 - 11:00 AM	MORNING BREAK									
11:00 - 12:30 PM	REST in Practice - Full Day Workshop on Web-based Distributed Systems (continued) Ian Robinson and Jim Webber	Visualizing Data on the Web Brian Sletten	The Busy Java Developer's Guide to Akka Ted Neward	Git Going with Distributed Version Control Matthew McCullough		Solr Recipes - 1/2 Day Workshop Erik Hatcher	Spring MVC Workshop (continued) Craig Walls	Modular Architecture - TODAY! Kirk Knoernschild	JRuby in Depth (continued) Neal Ford	Requirements and Estimating - state of the art Peter Bell
12:30 - 1:30 PM	LUNCH - WESTMINSTER BALLROOM 3/4									
1:30 - 3:00 PM	REST in Practice - Full Day Workshop on Web-based Distributed Systems (continued) Jim Webber	Grails: Bringing Radical Productivity to the JVM Part I Dave Klein	Busy Java Developer's Guide to Guava Ted Neward	Git Workshop (Bring A Laptop) Matthew McCullough	WebGL Brian Sletten	Solr Recipes - 1/2 Day Workshop (continued) Erik Hatcher	Personal Agility with the Pomodoro Technique Daniel Hinojosa	Modular Architecture - TODAY! (continued) Kirk Knoernschild	Enterprise Gradle (continued) Hans Dockter and Peter Niederwieser	ATDD/BDD with Cucumber Workshop (Bring A Laptop) Paul Rayner
3:00 - 3:15 PM	BREAK									
3:15 - 4:45 PM	REST in Practice - Full Day Workshop on Web-based Distributed Systems (continued) Ian Robinson and Jim Webber	Grails: Bringing Radical Productivity to the JVM Part II Dave Klein	Tricks of the Trade - What Every Developer Should Know About Application Security Frank Kim	HTML 5 Overview Brian Sletten	4 Practical Uses for Domain Specific Languages Neal Ford	Lucene for Solr Developers Erik Hatcher	Database Refactoring with Liquibase Tim Berglund	Scaling Agility Kirk Knoernschild	7 Reasons to #JBoss AS 7 Dan Allen	ATDD/BDD with Cucumber Workshop (Bring A Laptop) (continued) Paul Rayner
4:45 - 5:00 PM	BREAK									
						Securing Spring				

Thu, Jul. 14, 2011

	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
5:00 - 6:30 PM	Design Patterns in modern JVM Languages Venkat Subramaniam	Code Archaeology Matt Stine	JUnit Kung Fu: Getting More Out of Your Unit Tests John Smart	HTML 5 Overview (continued) Brian Sletten	How to Select and Adopt a Technology Peter Bell	Craig Walls	Database Refactoring Workshop Tim Berglund	Busy Developer's Guide to CouchDB Ted Neward	Painless Desktop Application Development: The Griffon Experience Andres Almiray	Working with Complex Adaptive (Human) Systems Esther Derby
6:30 - 7:30 PM	DINNER - WESTMINSTER BALLROOM 3/4									
7:30 - 8:30 PM	QUIK-FIRE SESSIONS: WESTMINSTER BALLROOM 3/4									
9:00 - 11:59 PM	UBERCONF 2011 OFF-SITE PARTY: HOSTED BY GRADLEWARE									

Fri, Jul. 15, 2011

	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Meadowbrook I	Meadowbrook II	Windsor	Waverly
8:00 - 9:00 AM	BREAKFAST - WESTMINSTER BALLROOM 3/4									
9:00 - 10:30 AM	Java/Groovy Cloud Computing Brian Sam-Bodden	Have Your Cake and Eat It Too: Meta-Programming Techniques for Java Howard Lewis Ship	Enterprise Messaging With Spring and ActiveMQ: Bruce Snyder	Smarter Testing with Spock Peter Niederwieser	Jenkins Continuous Integration in Action Matthew McCullough and John Smart	A Programmatic Introduction to Neo4j Workshop Ian Robinson and Jim Webber	All Together Now: How Teams Decide Esther Derby	Building Social-Ready Web Applications Craig Walls	Clojure: Lisp for the Real World Stuart Sierra	Building Semantic CSS with Compass and SASS David Bock
10:30 - 10:45 AM	MORNING BREAK									
10:45 - 12:15 PM	Java/Groovy Cloud Computing (continued) Brian Sam-Bodden	Rock SOLID Software Matt Stine	ActiveMQ In Action: Common Problems and Solutions Bruce Snyder	Functional Web Testing with Geb and Spock Peter Niederwieser	Jenkins Continuous Integration in Action (continued) Matthew McCullough and John Smart	A Programmatic Introduction to Neo4j Workshop (continued) Ian Robinson and Jim Webber	Improving Customer Conversations Esther Derby	Rethinking Object-Oriented: Clojure and the Expression Problem Stuart Sierra	JRuby in the Enterprise Jerry Gulla	Programming with Monads Venkat Subramaniam
12:15 - 1:30 PM	OUTDOOR BREAK & LUNCH - WESTMINSTER BALLROOM 3/4 & NORTH COURTYARD									
1:30 - 3:00 PM	Busy Java Developer's Guide to Physics Engines Ted Neward	Semantic Web Workshop Brian Sletten	Messaging and Concurrency in Your Applications Using Spring Bruce Snyder	Complexity Theory and Software Development Tim Berglund	Application Configuration using REST & Protocol Buffers Alex Antonov	A Programmatic Introduction to Neo4j Workshop (continued) Ian Robinson and Jim Webber	Motivation that Doesn't Misfire Esther Derby	Search Engine on a Shoestring: Clojure, Hadoop, Solr, and EC2 Stuart Sierra	Executable Specifications: Automating Your Requirements Document with Geb and Spock Matt Stine	Testing In Scala Daniel Hinojosa
3:00 - 3:15 PM	AFTERNOON BREAK									
3:15 - 4:45 PM	Busy Java Developer's Guide to Games Ted Neward	Semantic Web Workshop (continued) Brian Sletten	Styles of Application Integration Using Spring Bruce Snyder	Decision Making in Humane Software Teams Tim Berglund	Towards a More Usable Interface - Aesthetics Venkat Subramaniam	A Programmatic Introduction to Neo4j Workshop (continued) Ian Robinson and Jim Webber	Questionable Questions Esther Derby	Tools and Techniques to build Smart Java Applications Brian Sam-Bodden	Continuous QA Matt Stine	Getting Started with Tapestry Howard Lewis Ship
4:45 - 5:00 PM	CONCLUSION OF UBERCONF 2011 - THANK YOU FOR ATTENDING!									

ÜberConf

Westin Westminster
July 12 - 15, 2011

Software Craftsmanship: Positioning, Patterns and Practices by Peter Bell

None of us want to think of ourselves as "cowboy coders", but what does it mean to be a software craftsman, and is it a useful distinction? If so, what are some of the best patterns for honing our craft?

Requirements and Estimating - state of the art by Peter Bell

A chance for experience agile developers to learn and share state of the art tips for improving requirements gathering and project estimation. **Prerequisite:** *Experience working on agile projects*

How to Select and Adopt a Technology by Peter Bell

What's the point attending a conference unless you do something with the knowledge you gain? In this session we look at practical strategies for selecting new technologies and proven approaches for driving adoption back at the office. **Prerequisite:** *Frustration that you don't get to use all the cool technologies you learn about at No Fluff.*

HTML5: The JavaScript Parts by Tim Berglund

HTML5 wants to make some major changes to the way we deliver media over the web and the way we mark up our pages, but it also gives us a bunch of new stuff in the browser's programming model. To ignore these new JavaScript APIs is to give up on a richer browser UI and a lot of fun.

NoSQL Smackdown! by Tim Berglund

You've read that the relational model is old and busted, and there are newer, faster, web-scale ways to store your application's data. You've heard that NoSQL databases are the future! Well, what is all this NoSQL stuff about? Is it time to ditch Oracle, MySQL, and SQL Server in favor of the new guard? To be able to make that call, there's a lot you'll have to learn.

Cassandra: Radical NoSQL Scalability by Tim Berglund

Want to go deep on a popular NoSQL database? Cassandra is a scalable, highly available, column-oriented data store in use at Facebook, Twitter, Reddit, Rackspace, and other web-scale operations. It offers a compelling combination of a rich data model, a robust deployment track record, and a sound architecture, making it a good choice of NoSQL databases to study first. **Prerequisite:** *None, but NoSQL Smackdown! would be helpful preparation.*

Gaelyk: Cloud-Based Apps With Groovy by Tim Berglund

You love Groovy and you're a believer in cloud computing. For a larger project you might choose Grails and hosting on Amazon EC2, but what if you want to take advantage of the nearly massless deployments of a cloud provider like the Google App Engine? You could make Grails work, but it's not always the best fit. Enter Gaelyk.

Gaelyk Workshop Part I by Tim Berglund

Once you've been introduced to Gaelyk and the Groovy way it wraps the services the Google App Engine, it's time to write some code. Bring your laptop for a hands-on Gaelyk hack session in which we build a working Gaelyk app utilizing as many of the GAE services as we can pack into a 180 minutes of coding! **Prerequisite:** *Gaelyk: Lightweight Groovy on the Google App Engine (or a working knowledge of Gaelyk)*

Database Refactoring with Liquibase by Tim Berglund

Most teams manage database change using an ad-hoc system of SQL migration scripts manually applied to various development, staging, and production servers. Some even contrive automated processes, but rarely does this surplus build engineering deliver value directly to the customer. We should be writing applications, not build tools.

Database Refactoring Workshop by Tim Berglund

Take one ugly legacy schema, a toolbox full of simple database refactorings, and a world-class schema refactoring tool, and you've got 90 minutes of workshop that will equip you to bring a culture of database responsibility to your team. **Prerequisite:** *Database Refactoring with Liquibase*

Complexity Theory and Software Development by Tim Berglund

Some systems are too large to be understood entirely by any one human mind. They are composed of a diverse array of individual components capable of interacting with each other and adapting to a changing environment. As systems, they produce behavior that differs in kind from the behavior of their components. Complexity Theory is an emerging discipline that seeks to describe such phenomena previously encountered in biology, sociology, economics, and other disciplines.

Decision Making in Software Teams by Tim Berglund

Alistair Cockburn has described software development as a game in which we choose among three moves: invent, decide, and communicate. Most of our time at No Fluff is spent learning how to be better at inventing. Beyond that, we understand the importance of good communication, and take steps to improve in that capacity. Rarely, however, do we acknowledge the role of decision making in the life of software teams, what can cause it to go wrong, and how to improve it.

Agile.next by Neal Ford

Agile has matured to the point of mainstream success. Even large companies have discovered that it helps them build better quality software faster. But the agile practices that are mainstream today have been around for a long time. What is the next wave of innovation in the Agile world going to bring?

Emergent Design by Neal Ford

Emergent design is a big topic in the agile architecture and design community. This session covers the theory behind emergent design and shows examples of how you can implement this important concept.

Prerequisite: *understanding of architectural and design concepts*

Functional Thinking by Neal Ford

Learning the `_syntax_` of a new language is easy, but learning to `_think_` under a different paradigm is hard.

Agile Engineering Practices by Neal Ford

Most of the time when people talk about agile software development, they talk about project and planning practices and never mention actual development practices. This talk delves into best development practices for agile projects, covering all of its aspects. **Prerequisite:** *Having worked in an organization that values bureaucracy more than individuals*

Build Your Own Technology Radar by Neal Ford

A Technology Radar is a tool that forces you to organize and think about near term future technology decisions, both for you and your company.

JRuby in Depth by Neal Ford

Like hamburger & fries and turkey & dressing, JRuby allows you to harness the awesome power of Ruby in your Java projects. This workshop describes the origins, capabilities, and limitations of JRuby, the 100% pure-Java implementation of the Ruby programming language. This workshop also demonstrates some areas where it makes sense to mixin Ruby and Java code: building swing applications, testing, and dynamic programming.

JRuby in Depth by Neal Ford

Like hamburger & fries and turkey & dressing, JRuby allows you to harness the awesome power of Ruby in your Java projects. This workshop describes the origins, capabilities, and limitations of JRuby, the 100% pure-Java implementation of the Ruby programming language. This workshop also demonstrates some areas where it makes sense to mixin Ruby and Java code: building swing applications, testing, and dynamic programming.

4 Practical Uses for Domain Specific Languages by Neal Ford

Domain Specific Languages seems like a cool idea, but where's the payoff? This talk provides an overview of how to build both internal and external DSLs (including the state of the art tools), stopping along the way to show how this is practical to your day job.

Game Theory and Software Development by Matthew McCullough

The full title of this talk reveals its grand aims: Game Theory and Software Development: Explaining Brinkmanship, Irrationality, and Other Selfish Sins Once in a while, a topic, seemingly orthogonal to software development, presents a great opportunity to showcase how engineering can benefit from knowledge of seemingly more social disciplines. In this talk, the fundamental principles of economics' Game Theory are compared to often inexplicable behaviors and decisions we frequently observe in programming projects.

Monitoring 10 Critical Code Quality Metrics with Sonar by Matthew McCullough

You're serious about improving the quality of your code base, but with 10,000 lines of code, where do you start and how do you ensure the greatest ROI for the re-work your team members will perform? Sonar is an open source tool that brings together the best of breed static and dynamic analysis of Java projects. The result is a unified view of problematic areas of your code on a time-line basis, allowing the team to attack the problems with the best ROI, and maintain a more watchful eye for positive and risky trends in the codebase in the future.

Cryptography on the JVM: Boot Camp by Matthew McCullough

Does your application transmit customer information? Are there fields of sensitive customer data stored in your DB? Can your application be used on insecure networks? If so, you need a working knowledge of encryption and how to leverage Open Source APIs and libraries to make securing your data as easy as possible. Encryption is quickly becoming a developer's new frontier of responsibility in many data-centric applications.

Simpler Cryptography with 3 JVM Libraries by Matthew McCullough

Cryptography at first seems like a daunting topic. But after a basic intro and the leverage of the Java Cryptography Extension (JCE), it seems downright feasible to add encryption and decryption capabilities to your application. Developers weren't satisfied with just the JCE and its plug-in concepts though. Over the last few years, framework architects have made strides in either wrapping or re-writing the approachable JCE in more convenient APIs and fluent interfaces that make effective and accurate crypto down right simple. Explore three of these libraries -- Jasypt, BouncyCastle and KeyCzar -- and how they can be leveraged to make your next Java cryptography and data security effort a simple exercise and not a tribulation.

Prerequisite: *Basic understanding of cryptography (hashing, symmetric, asymmetric)*

Developer Productivity Power Ups on Mac OSX by Matthew McCullough

You're a talented coder and you apply many agile practices to your daily workflow. Still, you are looking for that next boost to better keep track of information, manage your open applications, make working with the terminal more productive, recall information quickly, manage files rapidly, and produce documentation in a portable and effective manner. This presentation will show you how to apply DevonThink, Delicious bookmarks, RSS feeds, Pinboard.in, Pomodoro, Things, LaunchBar, Bash profiles, mind maps, markdown files and spotlight filters to become a more productive developer that has a world of information sorted and accessible at a moment's notice. **Prerequisite:** *Very basic developer proficiency on the Mac.*

Thinking In Git by Matthew McCullough

Git is an innovative version control system that is taking the development world by storm. With that innovation comes new opportunities to leverage Git for more agile and productive workflows. This presentation steps up a level of abstraction from Git syntax and instead showcases the incredible team, branch and workflow dynamics that are easily accomplished with Git. **Prerequisite:** *Basic knowledge of any version control system*

Git Going with Distributed Version Control by Matthew McCullough

Many development shops have made the leap from RCS, Perforce, ClearCase, PVCS, CVS, BitKeeper or SourceSafe to the modern Subversion (SVN) version control system. But why not take the next massive stride in productivity and get on board with Git, a distributed version control system (DVCS). Jump ahead of the masses staying on Subversion, and increase your team's productivity, debugging effectiveness, flexibility in cutting releases, and repository redundancy at \$0 cost. Understand how distributed version

control systems are game-changers and pick up the lingo that will become standard in the next few years.

Prerequisite: *Basic understanding of Subversion or similar version control system*

Git Workshop (Bring A Laptop) by Matthew McCullough

Git is a version control system you may have been hearing a bit about lately. But simply hearing more about it may not be enough to convince you of its value. Getting hands on experience is what really counts. In this workshop, you'll bring your Windows, Mac or Linux laptop and walk through downloading, installing, and using Git in a collaborative fashion. **Prerequisite:** *Basic knowledge of a version control system. Subversion knowledge is a plus, but not imperative.*

Jenkins Continuous Integration in Action by Matthew McCullough and John Smart

The team dynamics and agile process revolution of the last several years has taught us that continuous integration (CI) is a necessary part of a healthy agile team. Jenkins (formerly Hudson) is the idea and footprint leader in the CI space. A recent survey stated that over 70% of all CI installations have Jenkins in their DNA. What's so awesome about this particular CI tool? Get on board with a ground-up survey of how to install, apply, upgrade, and leverage the free an open source Jenkins Continuous Integration server for your build, whether it be Ant, Maven, Gradle, JavaScript, Rake, or just shell scripts.

Jenkins Continuous Integration in Action by Matthew McCullough and John Smart

The team dynamics and agile process revolution of the last several years has taught us that continuous integration (CI) is a necessary part of a healthy agile team. Jenkins (formerly Hudson) is the idea and footprint leader in the CI space. A recent survey stated that over 70% of all CI installations have Jenkins in their DNA. What's so awesome about this particular CI tool? Get on board with a ground-up survey of how to install, apply, upgrade, and leverage the free an open source Jenkins Continuous Integration server for your build, whether it be Ant, Maven, Gradle, JavaScript, Rake, or just shell scripts.

Android Workshop - Full Day by Ted Neward

First there was iPod. Then iPhone. Then iPad. And with each new release, the mobile device market grew hotter and hotter. Now, as Google's entry into this race, the Android system, begins to hit its stride as a competitor platform to the iOS, as a Java developer you're intrigued—it's Java (well, assuming you ask anybody except Oracle), and it's a mobile device, and it's open source, and.... What's not to love?

Pragmatic Architecture by Ted Neward

Building an application is not the straightforward exercise it used to be. Decisions regarding which architectural approaches to take (n-tier, client/server), which user interface approaches to take (Smart/rich client, thin client, Ajax), even how to communicate between processes (Web services, distributed objects, REST)... it's enough to drive the most dedicated designer nuts. This talk discusses the goals of an application architecture and why developers should concern themselves with architecture in the first place. Then, it dives into the meat of the various architectural considerations available; the pros and cons of JavaWebStart, ClickOnce, SWT, Swing, JavaFX, GWT, Ajax, RMI, JAX-WS, , JMS, MSMQ, transactional processing, and more.

Architectural Kata Workshop by Ted Neward

Fred Brooks said, "How do we get great designers? Great designers design, of course." So how do we get great architects? Great architects architect. But architecting a software system is a rare opportunity for the non-architect. The kata is an ancient tradition, born of the martial arts, designed to give the student the opportunity to practice more than basics in a semi-realistic way. The coding kata, created by Dave Thomas, is an opportunity for the developer to try a language or tool to solve a problem slightly more complex than "Hello world". The architectural kata, like the coding kata, is an opportunity for the student-architect to practice architecting a software system.

The Busy Java Developer's Guide to Akka by Ted Neward

With the rise of multi-core processors, and their growing ubiquity (on client machines, to say nothing of the server machines on which Java applications most frequently execute), the need to "program concurrently" has risen from "nice-to-have" to "mandatory" requirement, and unfortunately the traditional threading-and-locking model is just too complicated for most Java developers--even the brightest of the lot--to keep track of with any degree of reliability. As a result, numerous new solutions are emerging, each of

them with their own strengths and weaknesses, leaving the Java developer in a bit of a quandary as to which to examine.

Busy Java Developer's Guide to Guava by Ted Neward

"The Google Guava project contains a host of new features/classes for use by the Java programmer. Intended as a drop-in supplement for the standard JDK APIs, Guava provides features like immutable and forwarding collections, some concurrency utilities, more support for primitives, and so on.

Busy Developer's Guide to CouchDB by Ted Neward

With the rise of the NoSQL movement, a whole new crop of different ways to store data suddenly became available to the Java developer. Unfortunately, what didn't come with them was an owner's manual. CouchDB, for example, was the first of the NoSQL databases to be named as such, and offers features not found in the traditional RDBMS: A distributed, robust, incremental replication document-oriented database server with bi-directional conflict detection and management, accessible via a RESTful JSON API, stored ad-hoc and schema-free with a flat address space, that is both query-able and index-able, featuring a table oriented reporting engine that uses JavaScript as a query language. (With a list of buzzwords like that, what's not to love?)

Busy Java Developer's Guide to Physics Engines by Ted Neward

"From Wikipedia: "A physics engine is computer software that provides an approximate simulation of certain simple physical systems, such as rigid body dynamics (including collision detection), soft body dynamics, and fluid dynamics, of use in the domains of computer graphics, video games and film. Their main uses are in video games (typically as middleware), in which case the simulations are in real-time. The term is sometimes used more generally to describe any software system for simulating physical phenomena, such as high-performance scientific simulation."

Busy Java Developer's Guide to Games by Ted Neward

Games? What do games have to do with good business-oriented applications? Turns out, a lot of interesting little tidbits of user-interface, distribution, and emergence, found normally in the games we play, have direct implications on the way enterprise applications can (or should) be built.

Visualizing Data on the Web by Brian Sletten

We are far from the early days of ugly HTML. We have sophisticated visualization tools available to us now to help our users consume complex data in attractive and informative ways. Come hear how you can adopt these visualization systems (calling them libraries is inappropriate) today.

WebGL by Brian Sletten

HTML 5 has introduced us to the Canvas API, 2D graphics and the pleasures of plugin-free video and audio playback. One of the next hurdles we will face is native support for 3D graphics for simulations, visualizations and games.

HTML 5 Overview by Brian Sletten

People are confused about the status of HTML 5. Is it ready? Is it not? What is part of the spec and what isn't? We'll talk about the situation in the "HTML 5 and the Kitchen Sink" discussion, but as always, the proof is in the pudding. We will introduce the most exciting new features of HTML 5 and its related technologies and build examples that use them.

HTML 5 Overview by Brian Sletten

People are confused about the status of HTML 5. Is it ready? Is it not? What is part of the spec and what isn't? We'll talk about the situation in the "HTML 5 and the Kitchen Sink" discussion, but as always, the proof is in the pudding. We will introduce the most exciting new features of HTML 5 and its related technologies and build examples that use them.

Semantic Web Workshop by Brian Sletten

The Web is changing faster than you can imagine and it is going to continue to do so. Webs of Documents are giving way to machine-processable Webs of Information. We no longer care about data containers, we only care about data and how it connects to what we already know. Perhaps the concepts of the Semantic Web initiative are new to you. Or perhaps you have been hearing for years how great technologies like RDF, SPARQL, SKOS and OWL are and have yet to see anything real come out of it. Whether you are jazzed or

jaded, this workshop will provide you with the understanding of a technological tidal wave that is heading in your direction.

Semantic Web Workshop by Brian Sletten

The Web is changing faster than you can imagine and it is going to continue to do so. Webs of Documents are giving way to machine-processable Webs of Information. We no longer care about data containers, we only care about data and how it connects to what we already know. Perhaps the concepts of the Semantic Web initiative are new to you. Or perhaps you have been hearing for years how great technologies like RDF, SPARQL, SKOS and OWL are and have yet to see anything real come out of it. Whether you are jazzed or jaded, this workshop will provide you with the understanding of a technological tidal wave that is heading in your direction.

Effective Java Reloaded by Matt Stine

Even with the recent explosion in alternative languages for the JVM, the vast majority of us are still writing code in "Java the language" in order to put bread on the table. Proper craftsmanship demands that we write the best Java code that we can possibly write. Fortunately we have a guide in Joshua Bloch's *Effective Java.*

Effective Java Reloaded by Matt Stine

Even with the recent explosion in alternative languages for the JVM, the vast majority of us are still writing code in "Java the language" in order to put bread on the table. Proper craftsmanship demands that we write the best Java code that we can possibly write. Fortunately we have a guide in Joshua Bloch's *Effective Java.*

Code Archaeology by Matt Stine

Feature requests are steadily pouring in, but the team cannot respond to them. They are paralyzed. The codebase on which the company has "bet the business" is simply too hard to change. It's your job to clean up the mess and get things rolling again. Where do you begin? Your first task is to get the lay of the land by applying a family of techniques we'll call "Code Archaeology."

Rock SOLID Software by Matt Stine

Object-oriented programming was formally introduced in the 1970's with the advent of Smalltalk. C++ took it mainstream in the 1980's, and Java carried it to the next level in the 1990's. Unfortunately, if you examine the vast majority of Java codebases, what you'll find is a bunch of C-style structs (a.k.a. JavaBeans) and functions. As these codebases grow, a number of design smells can potentially crop up, which in turn cripple our ability to respond to change. We need SOLID principles that we can apply to keep our software clean and malleable.

Executable Specifications: Automating Your Requirements Document with Geb and Spock by Matt Stine

One of the hallmarks of lean software development is the elimination of waste. Several of the key wastes in software development revolve around incomplete, incorrect, or obsolete documentation, especially documentation of requirements. One effective means of ensuring that your requirements documentation is complete, correct, and up-to-date is to make it executable. That sounds nice, but how do we get it done, especially in the world of modern, cross-browser web applications?

Continuous QA by Matt Stine

You've got your build automated using Ant/Maven/Gradle and you're building and running your unit test suite every time you check-in. That's easy. In fact, with Jenkins you can do this in under 5 minutes. However, running your user acceptance tests every check-in is a bit more tricky. Not only do you have to build and package the project, you need to deploy it somewhere and then fire off your acceptance test suite. Fortunately there are a few tools out there that properly glued together provide a very effective solution to this problem.

Fundamentals of iOS Apps Development (day long) by Venkat Subramaniam

Come to this workshop for an in depth understanding of the fundamentals of developing applications on the iOS platform for iPhone and iPad devices.

Programming Concurrency (Full Day) by Venkat Subramaniam

Starting threads is easy, benefiting from them without being burnt is hard. The multithreading API in Java has evolved quite a bit since the early stages. There are three options for concurrency in pure Java today: the modern JDK API, the Software Transaction Memory, and the actor based concurrency. In this workshop, using practical examples—all in pure Java, you will dive deep into all three of these options, learn how to use these, learn their pros and cons, and clearly understand which option to pick when.

Programming Concurrency (Full Day) by Venkat Subramaniam

Starting threads is easy, benefiting from them without being burnt is hard. The multithreading API in Java has evolved quite a bit since the early stages. There are three options for concurrency in pure Java today: the modern JDK API, the Software Transaction Memory, and the actor based concurrency. In this workshop, using practical examples—all in pure Java, you will dive deep into all three of these options, learn how to use these, learn their pros and cons, and clearly understand which option to pick when.

Programming Concurrency (Full Day) by Venkat Subramaniam

Starting threads is easy, benefiting from them without being burnt is hard. The multithreading API in Java has evolved quite a bit since the early stages. There are three options for concurrency in pure Java today: the modern JDK API, the Software Transaction Memory, and the actor based concurrency. In this workshop, using practical examples—all in pure Java, you will dive deep into all three of these options, learn how to use these, learn their pros and cons, and clearly understand which option to pick when.

Programming Concurrency (Full Day) by Venkat Subramaniam

Starting threads is easy, benefiting from them without being burnt is hard. The multithreading API in Java has evolved quite a bit since the early stages. There are three options for concurrency in pure Java today: the modern JDK API, the Software Transaction Memory, and the actor based concurrency. In this workshop, using practical examples—all in pure Java, you will dive deep into all three of these options, learn how to use these, learn their pros and cons, and clearly understand which option to pick when.

Hands-on Scala 1/2 Day Workshop by Venkat Subramaniam

In this workshop we will take some common everyday tasks and learn how to program it using Scala.

Hands-on Scala 1/2 Day Workshop by Venkat Subramaniam

In this workshop we will take some common everyday tasks and learn how to program it using Scala.

Design Patterns in modern JVM Languages by Venkat Subramaniam

The GOF design patterns were quite centered around OOP languages. Now that we have dynamic and functional languages on the JVM, there are quite a few other patterns that come in handy with these capabilities.

Programming with Monads by Venkat Subramaniam

Monads are quite unheard of in imperative style of programming. Monads, however, play an interesting roll in a purely functional programming language. Come to this session if you're heard of monads and wondered what in the world they were.

Towards a Humane Interface—Aesthetics and Usability by Venkat Subramaniam

A successful application has to focus on three dimensions—value (business), design (engineering) and usability. Usability is not only about the wow factor. It is about making the application easier and intuitive to use. In this presentation we will learn the fundamentals of creating a usable application. We will look at some basic dos and don't. These will help you move forward from being a programmer to a good application developer.

What's new in Spring by Craig Walls

In this session, I'll lead a guided tour through the latest that Spring has to offer. Whether you're a Spring veteran or a Spring newbie, there will be something new for nearly everyone.

NoXML: Spring for XML-Haters by Craig Walls

In this presentation, we'll explore all of the ways to do bean wiring in Spring We'll take a pragmatic view of each style, evaluating their strengths, weaknesses, and applicability to varying circumstances.

Introducing Spring Roo: From Zero to Working Spring Application in Record Time by Craig Walls

In this example-driven session we'll see how to swiftly develop Spring applications using Spring Roo. We'll start with an empty directory and quickly work our way up to a fully functioning web application. You'll see how Roo handles a lot of heavy-lifting that you'd normally have to do yourself when working with Spring. And we'll stop at a few scenic points along the way to see how Roo accomplishes some of its magic.

Spring Roo Workshop by Craig Walls

In this hands-on workshop, we'll work together developing a Spring application using Spring Roo.

Spring MVC Workshop by Craig Walls

For as long as there has been a Spring Framework, there has been Spring MVC, a web framework built around the principals of Spring. Although it was originally designed around a deep hierarchy of controller classes and focused on HTML-oriented views, Spring MVC has evolved in the past few years to embrace an annotation-oriented model and RESTful web development.

Spring MVC Workshop by Craig Walls

For as long as there has been a Spring Framework, there has been Spring MVC, a web framework built around the principals of Spring. Although it was originally designed around a deep hierarchy of controller classes and focused on HTML-oriented views, Spring MVC has evolved in the past few years to embrace an annotation-oriented model and RESTful web development.

Securing Spring by Craig Walls

In this session, I'll show you how to secure your Spring application with Spring Security 3.0. You'll see how to declare both request-oriented and method-oriented security constraints. And you'll see how SpEL can make simple work of expressing complex security rules.

Building Social-Ready Web Applications by Craig Walls

Businesses are increasingly recognizing the value of connecting with their customers on a more personal level. Companies can utilize social networking to transition from "Big Faceless Corporation" to "Friend" by taking their wares to the online communities where their customers are. In this age of social media, those communities are found at social network sites such as Facebook, Twitter, and LinkedIn.

The Future of Java Enterprise Testing by Dan Allen

This talk unveils the missing link in enterprise Java development: simple, portable integration tests. While development life is simple with unit tests and mocks, they only take you so far. Arquillian, a container-oriented testing framework layered atop TestNG and JUnit, tears down this barrier. It brings your test to the runtime rather than requiring you to manage the runtime from your test. That means you can test real components that rely on real enterprise services in a real runtime.

7 Reasons to # JBoss AS 7 by Dan Allen

Fast, fast, fast. Blazing fast! No doubt, that's the main reason to love JBoss AS 7. This talk dispells a long-standing misconception that Java EE application servers are inherently slow. With JBoss AS 7, you get to keep more memory for your applications AND you experience a 10-fold reduction in startup time over previous revisions.

The Groovy Ecosystem by Andres Almiray

Groovy is the fastest growing JVM language out there. It might be because it is so easy for Java developers to pick it up, but also because there's an increasing number of projects and libraries that make use of Groovy as the starting point.

Painless Desktop Application Development: The Griffon Experience by Andres Almiray

Despite of all the buzz and hype around webapps over the last 8 years fact is that desktop applications are still found in many places, specially in the enterprise. However the legends are true: building desktop applications is a hard job. But it does not have to be. Enter Griffon. Griffon aims to bring back the fun and productivity to desktop application development in the same way Grails did it (and continues to do so) on the

web. Griffon is rooted in the JVM but has Grails in its DNA. This means you'll find yourself right at home if you're a Java veteran, same goes for all of you that made the jump to Grails.

Architectural Choices around Open Source Solutions by Alex Antonov

The audience will be presented with a number of open source technologies that would enable building different layers of a multi-tier system.

Application Configuration using REST & Protocol Buffers by Alex Antonov

The presentation will cover different types of application configuration, their lifecycle and management. The audience will learn about approaches on how to separate the configuration API from consumption and provisioning. The benefits that can be achieved from a well defined API providing ease of development, nice IDE support, type handling and true data objects, while maintaining the flexibility of being able to retrieve configuration from different sources and in different formats. They will learn about supporting different data stores, such as CouchDB, flat files, remote services and file repositories, as well as supporting different file formats like XML, JSON, Protobuf binary, etc.

Building Workflow Applications with StonePath by David Bock

Stonepath is a workflow modeling methodology with its roots in a long-running Java project at the U.S. State Department. Starting with techniques for deriving requirements/user stories from your users, user interface patterns, state-and-task based workflow modeling, and some domain modeling ideas, you can build comprehensive 'Enterprise' applications for managing aspects of workflow and group coordination.

Building Maintainable Javascript with Coffeescript by David Bock

CoffeeScript is a little language that compiles into JavaScript. Underneath all of those embarrassing braces and semicolons, JavaScript has always had a gorgeous object model at its heart. CoffeeScript is an attempt to expose the good parts of JavaScript in a simple way. The golden rule of CoffeeScript is: "It's just JavaScript". The code compiles one-to-one into the equivalent JS, and there is no interpretation at runtime. You can use any existing JavaScript library seamlessly (and vice-versa). The compiled output is readable and pretty-printed, passes through JavaScript Lint without warnings, and runs in every JavaScript implementation.

Metrics for steering your projects to success by David Bock

There are a lot of things we can measure about our source code, but what about the "project as a whole" and its overall health? Are there ways of measuring the effectiveness of our processes? Are there things we can measure that would point to project automation wins? Is there a way to measure team 'morale'?

Building Semantic CSS with Compass and SASS by David Bock

Compass is a tool that can help you build cleaner, better structured, and less error-prone CSS. Semantic CSS is a technique where your CSS vocabulary describes *WHAT* things are on your page, rather than *WHERE* they are. Together, this tool and this concept can radically improve the structure of your html.

Behavior Driven Development with Client Side Javascript by James Carr

Back in 2004 there wasn't much in the world of test driven development for javascript... you just had jsunit. Since then there has been an explosion of frameworks for javascript and it is almost to the point where it is difficult to choose the right one.

NodeJS Bootcamp by James Carr

You might have heard a bit about nodejs, now it's time to get fully immersed in it and not just learn it in detail but gear up to start becoming an active member of the nodejs development community.

NodeJS Bootcamp by James Carr

You might have heard a bit about nodejs, now it's time to get fully immersed in it and not just learn it in detail but gear up to start becoming an active member of the nodejs development community.

Effective Groovy by Hamlet D`Arcy

"Effective Java" by Joshua Bloch is the gold standard for how to write correct and idiomatic Java code. Wouldn't it be great if the same thing existed for Groovy? Well here it is. This interactive, live coding session discusses what separates good Groovy code from the bad, what makes some code great, and how best to

use the many available libraries. We'll also look at several static analysis tools for Groovy that aid in these pursuits. This session appeals to both those with a Java knowledge looking to learn Groovy and advanced Groovy programmers looking to learn more.

Code Generation on the JVM: Writing Code that Writes Code by Hamlet D`Arcy

"The Pragmatic Programmer" admonished us all to "write code that writes code": use code generators to increase productivity and avoid duplication. Today's language communities have clearly caught on, as more and more frameworks generate code at compile or runtime. This session covers Project Lombok, Cofaja, Spring Roo, GContracts, Groovy++, and more. We'll reviews the different approaches, including examples of how and why we'd want to do this. Come see how these frameworks are using things like Java and Groovy AST Transformations, AspectJ intertype definitions, and ASM bytecode generators. You'll get an in-depth look at language tools and production deployed AST Transforms and code generators. Audience: developers searching for cutting edge solutions to increasing team velocity.

New Ideas for Old Code by Hamlet D`Arcy

Left unattended software can expand into a complex, brittle maintenance nightmare. But don't despair! This session teaches strategies for modernizing even the most horrid code swamps, examining incremental refactorings and the dos and don'ts of testing legacy code. We'll also tackle the harder, cultural issues: how to inspire your co-workers and keep your moral high even on the dirtiest jobs.

Working with Complex Adaptive (Human) Systems by Esther Derby

The world abounds with complex theories and complex advice about complex adaptive systems. But most of them aren't very helpful when it comes to knowing what to do to make a system work better. In this interactive session, we'll explore three levers that you can use to influence patterns of behavior in complex adaptive systems...such as software development teams.

All Together Now: How Teams Decide by Esther Derby

Have you had the experience of watching smart people argue endlessly over which technology to use? Have you walked out of a meeting believing the group had agreed, only to find out that five different people thought they'd agreed to five different things? People in our industry pride themselves on their brain power and ability to make good decisions. And most of us are good at thinking, learning, and deciding—on our own. When we work collaboratively on interdependent work, though, we need to think and decide as a group if we want to realize the benefits of the team effect. In this session, we'll experience a group decision. Then, we'll look at the pieces and parts of the process to see what we can learn about how groups think and decide together. We'll tease out the techniques that will help you help groups take advantage of all their expertise, see other points of view, and arrive at high-quality decisions.

Improving Customer Conversations by Esther Derby

It's not easy to build the right product. People sometimes don't know exactly what they need, want things that won't help, and don't imagine what's possible. Agile project capture requirements on cards that contain a statement of want and benefit and notes on how to confirm the need is met. The intention isn't to fully document the requirement on the card, but to make a note and create a reminder for a conversation with the customer. Whether you are using agile methods or traditional requirements, valuable products start with understanding the customers context, their problems, what they want, and how they use a product. However, most people aren't born with the ability to speak naturally in user stories or fully formed requirements statements. So we must learn how to ask the right questions, draw out pertinent information and understand the customer's world in those conversations.

Motivation that Doesn't Misfire by Esther Derby

Many managers ask me, "How can I motivate my team?" The zeroth step in boosting motivation is to stop doing things that demotivate people. But what is a manager to do after that? Prizes, treats, rewards, pep talks, and recognition events don't cut it. Why? Many of the common attempts to improve motivation rely on an external source of motivation. That assumes that people need a carrot (or a stick) to keep them going. Research shows a contrary conclusion—that intrinsic motivation has more sustaining power.

Questionable Questions by Esther Derby

Questions are powerful. Presidential historians believe that the questions presidents ask and the way they ask those questions have huge ramifications. Questions asked and not asked influence policy initiatives, invasions, and trips to the moon. Likewise, success or failure hangs on the questions managers and

technical people ask when planning releases, making decisions, considering strategy alternatives or looking for improvements. Yet we don't often stop to consider the questions we ask. Every question contains assumptions and while the question opens one avenue of inquiry, it closes others. In this session, we'll consider the questions we do ask, the questions we don't ask, the questions we could ask. We'll look at what the questions people ask us reveal assumptions, who gets to ask questions, questions that mislead, and when its best not to ask questions, but rather to go and see.

Enterprise Gradle by Hans Dockter and Peter Niederwieser

This presentation introduces the audience to the powerful concepts of Gradle through many real-world examples that are demonstrated live. By the end of the presentation, the participants understand how Gradle helps to elegantly solve the challenges that we face in our daily enterprise builds.

JRuby in the Enterprise by Jerry Gulla

Bring the power of Ruby and Rails to your enterprise! Interested in harnessing the power of Ruby on Rails but not sure how to incorporate it into your existing Java/J2EE/Spring world? In this session, we'll talk about real-world experiences, lessons learned and best practices for developing a new Ruby on Rails using JRuby.

Rapid Prototyping w/Solr by Erik Hatcher

Got data? Let's make it searchable! Rapid Prototyping with Solr will demonstrate getting documents into Solr quickly, provide some tips in adjusting Solr's schema to match your needs better, and finally will discuss how to showcase your data in a flexible search user interface. We'll see how to rapidly leverage faceting, highlighting, spell checking, and debugging. Even after all that, there will be enough time left to outline the next steps in developing your search application and taking it to production.

Solr Recipes - 1/2 Day Workshop by Erik Hatcher

Solr Recipes provides quick and easy steps for common use cases with Apache Solr. Bite-sized recipes will be presented for data ingestion, textual analysis, client integration, and each of Solr's features including faceting, more-like-this, spell checking/suggest, and others.

Solr Recipes - 1/2 Day Workshop by Erik Hatcher

Solr Recipes provides quick and easy steps for common use cases with Apache Solr. Bite-sized recipes will be presented for data ingestion, textual analysis, client integration, and each of Solr's features including faceting, more-like-this, spell checking/suggest, and others.

Lucene for Solr Developers by Erik Hatcher

You're Solr powered, and needing to customize its capabilities. Apache Solr is flexibly architected, with practically everything pluggable. Under the hood, Solr is driven by the well-known Apache Lucene. Lucene for Solr Developers will guide you through the various ways in which Solr can be extended, customized, and enhanced with a bit of Lucene API know-how. We'll delve into improving analysis with custom character mapping, tokenizing, and token filtering extensions; show why and how to implement specialized query parsing, and how to add your own search and update request handling.

Joda Time and a Brief History of the World by Daniel Hinojosa

JodaTime is Java Date/Time and Calendaring done right. There are many problems with the original Date/Time API that came prepackaged in the early Java days. There are even more problems with the subsequent Calendar API. One of the obvious issues is that Calendar is mutable and can unintentionally be changed. Another issue is that constructing Calendars in Java involves setting certain fields at certain times during coding, but not always getting the expected result. Joda Time repairs those issues and offers a robust and immutable date, time, and duration API.

Personal Agility with the Pomodoro Technique by Daniel Hinojosa

Time is very precious and is often threatened by phone calls, emails, co-workers, bosses, and most of all, yourself. The Pomodoro Technique reigns in unfocused time and gives your work the urgency and the attention it needs, and it's done with a kitchen timer.

Testing In Scala by Daniel Hinojosa

Most introductory programming books include a chapter on testing, seemingly as an afterthought. For the test-driven developer, that's a little too late. Some programmers approach a new programming language with a few test-cases to understand a concept. Others thrive under fire and want to hit the ground running in a

new programming language by creating an application. Regardless of your profile, this presentation will help you get started with a Scala testing environment so you can concentrate on the finer points of the language.

Continuous Delivery Part I: Value proposition, the deployment pipeline, automated tests, CI by Jez Humble

The first 1h30 will introduce the value proposition of continuous delivery, and present the deployment pipeline, the key pattern at the heart of continuous delivery. The second 1h30 will discuss creating and maintaining automated tests, and continuous integration and patterns for developing on mainline such as feature toggles and dark launching.

Continuous Delivery Part I: Value proposition, the deployment pipeline, automated tests, CI by Jez Humble

The first 1h30 will introduce the value proposition of continuous delivery, and present the deployment pipeline, the key pattern at the heart of continuous delivery. The second 1h30 will discuss creating and maintaining automated tests, and continuous integration and patterns for developing on mainline such as feature toggles and dark launching.

Continuous Delivery Part II: components, going live, agile infrastructure, databases, organisational transformation by Jez Humble

In the 3rd 1h30, we will discuss componentised or service architectures, patterns for low-risk releases, and agile infrastructure management. In the 4th 1h30, I cover data management and organizational transformation. If there's time, there will be a bonus session on architectural patterns. **Prerequisite:** *Continuous Delivery Part I*

Continuous Delivery Part II: components, going live, agile infrastructure, databases, organisational transformation by Jez Humble

In the 3rd 1h30, we will discuss componentised or service architectures, patterns for low-risk releases, and agile infrastructure management. In the 4th 1h30, I cover data management and organizational transformation. If there's time, there will be a bonus session on architectural patterns. **Prerequisite:** *Continuous Delivery Part I*

Tricks of the Trade - What Every Developer Should Know About Application Security by Frank Kim

Learn how to exploit security vulnerabilities that are commonly found in the arsenal of malicious attackers. We won't simply talk about issues like Cross Site Scripting (XSS) and Cross Site Request Forgery (CSRF), but will show how hackers abuse these potentially devastating defects by finding and exploiting vulnerabilities in real world open source web applications built in Java. We will proceed to walk through the source code and actually fix these issues using secure coding techniques. We will also discuss best practices that can be used to build security into your SDLC.

Über Groovy by Dave Klein

You've probably heard about Groovy, the dynamic language for the JVM. You may have heard that Groovy is what Java would have been if it had been written in the 21st century. Maybe you've even seen some demos of the seemingly magical things that you can do with Groovy. Well, now it's time to download the bits and experience the fun and productivity for yourself.

Über Groovy by Dave Klein

You've probably heard about Groovy, the dynamic language for the JVM. You may have heard that Groovy is what Java would have been if it had been written in the 21st century. Maybe you've even seen some demos of the seemingly magical things that you can do with Groovy. Well, now it's time to download the bits and experience the fun and productivity for yourself.

Grails: Bringing Radical Productivity to the JVM Part I by Dave Klein

The goal of this hands-on tutorial is to get started and get productive with Grails. We'll do this by jumping right in and building an application, from design to deployment.

Grails: Bringing Radical Productivity to the JVM Part II by Dave Klein

In Part II of this session, we will continue the build out process with the Grails application.

OSGi Demystified by Kirk Knoernschild

In 2007, OSGi was heralded as a contender for most important technology of the decade. Today, most developers have heard of OSGi, but few are using it to develop their enterprise software applications. OSGi might be the most heralded technology that nobody is using. Is OSGi failing? Who is using it? And what exactly are its benefits? Is it really too complex for the average enterprise developer?

Modular Architecture - TODAY! by Kirk Knoernschild

Modularity is coming to the Java platform! But contrary to popular belief, you don't need a framework or a new runtime to start building modular software applications. You can start today. Learn how!

Modular Architecture - TODAY! by Kirk Knoernschild

Modularity is coming to the Java platform! But contrary to popular belief, you don't need a framework or a new runtime to start building modular software applications. You can start today. Learn how!

Scaling Agility by Kirk Knoernschild

Agile methods are proven on small teams. But really...almost any process works with a team of one. As team size increases, however, challenges mount. How do we organize the team? How do I manage Sprints across the teams? Is it possible to conduct effective code reviews? What are the essential practices that maintain team unity?

Applying Patterns: How to Spot Problem Code and What To Do About It by Howard Lewis Ship

How do you spot bad code? How do you turn it into good code? We'll be looking at code examples from real applications. We'll start by identifying the problem with the code: things like maintainability, clarity, and testability. Then we'll look for ways to improve that code: perhaps introducing base classes, perhaps other refactorings based on Gang of Four Patterns.

Have Your Cake and Eat It Too: Meta-Programming Techniques for Java by Howard Lewis Ship

Ever been envious of how easily Python, Ruby and even JavaScript can "meta-program"? It's like magic: seemingly simple or innocuous code takes over big responsibilities; new methods appear out of thin air. Your code, your primary code, stays simple and easy to follow. Now, we know you can do that for scripting languages, but what do we do about Java? With the proper context, it is possible to emulate many of those same capabilities, by applying a simple set of code transformations at runtime.

Getting Started with Tapestry by Howard Lewis Ship

Apache Tapestry is designed to be a fast, easy to use, high-performance, general purpose web framework. Tapestry is laser-focused on giving you maximum bang for your programming buck, and this shows up as a broad range of well-integrated features. Don't let unfamiliarity get in the way of learning this powerful, friendly tool.

Gradle Workshop by Peter Niederwieser

This two-part workshop provides a hands-on introduction to Gradle. You will learn the fundamentals of Gradle's build language, leverage some of its built-in tasks and plugins, use Gradle's Ant integration, learn how to manage dependencies with Gradle, and top it off with a multi-project build. Please bring your laptops! Familiarity with Java is assumed; familiarity with Groovy is a plus but not required.

Gradle Workshop by Peter Niederwieser

This two-part workshop provides a hands-on introduction to Gradle. You will learn the fundamentals of Gradle's build language, leverage some of its built-in tasks and plugins, use Gradle's Ant integration, learn how to manage dependencies with Gradle, and top it off with a multi-project build. Please bring your laptops! Familiarity with Java is assumed; familiarity with Groovy is a plus but not required.

Smarter Testing with Spock by Peter Niederwieser

Spock is a developer testing framework for Java and Groovy applications. Even though it is fully JUnit-compatible on the outside, Spock isn't just another JUnit clone - its goal is to take developer testing to

the next level! With its Groovy-powered and highly expressive testing language, Spock boosts productivity and brings back the fun to testing.

Functional Web Testing with Geb and Spock by Peter Niederwieser

Geb is a next generation Functional Web Testing tool that removes the ceremony and tedium of traditional web testing, leaving you with a concise, pragmatic and productive environment in which to work. It combines the power of Groovy with the WebDriver/Selenium 2.0 browser automation library to provide a programmer's DSL for modeling pages (known as the PageObject Pattern) and easily automating real browsers such as Internet Explorer, FireFox and Chrome as well as the HTMLUnit library. Geb can be used standalone, or with testing frameworks such as Spock, JUnit, EasyB or Cucumber.

Java Web Application Security: Develop. Penetrate. Protect. Relax. by Matt Raible

In this session, you'll learn how to implement authentication in your Java web applications using Spring Security, Apache Shiro and good ol' Java EE 6 Container Managed Authentication. You'll also learn how to secure your REST API with OAuth and lock it down with SSL.

Strategic Design Using DDD by Paul Rayner

Not every part of a software system will be well-designed. How do you know where to put the time and effort to refine the design, or refactor existing code? Learn how strategic Domain-Driven Design (DDD) patterns can show you how to know which parts of your system matter most to your business and how to focus your team's design efforts most effectively.

Using DDD Patterns for Supple Design by Paul Rayner

Come on a guided tour of how applying Domain-Driven Design (DDD) building block patterns can make your code cleaner, more expressive, and more amenable to change. We cover examples of DDD patterns such as entities, value objects, closure of operations and side-effect-free functions.

Introduction to Lean-Agile Software Development by Paul Rayner

Successful software development is about building the right product at the right time for your customers. This means focusing attention on the right places in the portfolio of projects and products that your company provides, and optimizing the entire value stream from "concept to cash" for your customers and the development teams.

Measure for Measure – Lean Principles for Effective Metrics and Motivation by Paul Rayner

This presentation explores the nature of motivation and the place of metrics and measurement in software development, and how lean software development principles and practices shed light on motivation and metrics and how they can be used to support deep organizational improvement.

ATDD/BDD with Cucumber Workshop (Bring A Laptop) by Paul Rayner

Acceptance Test-Driven Design (ATDD), or Behavior-Driven Development (BDD), employs the approach of specification by example. Cucumber is such an amazing ATDD tool because it's so good at mapping stories and acceptance criteria to automated functional tests. Product Owners, developers and testers collaborate together to write acceptance criteria in natural language and unobtrusively automate tests for them. This is a hands-on workshop that will have you writing and automated acceptance tests on your own laptop by the conclusion of this session.

ATDD/BDD with Cucumber Workshop (Bring A Laptop) by Paul Rayner

Acceptance Test-Driven Design (ATDD), or Behavior-Driven Development (BDD), employs the approach of specification by example. Cucumber is such an amazing ATDD tool because it's so good at mapping stories and acceptance criteria to automated functional tests. Product Owners, developers and testers collaborate together to write acceptance criteria in natural language and unobtrusively automate tests for them. This is a hands-on workshop that will have you writing and automated acceptance tests on your own laptop by the conclusion of this session.

Designing RESTful Services by Ian Robinson

REST is ready for the enterprise. Imagine an information platform that is open and available to systems throughout the enterprise estate. A platform that eschews integration in favour of composition, connected

data over siloed databases. A networked data structure with the power to implement valuable business behaviours: a distributed, hypermedia-driven application platform.

REST in Practice - Full Day Workshop on Web-based Distributed Systems by Ian Robinson and Jim Webber

The Web is fast becoming a serious competitor to traditional enterprise architecture approaches. This full day workshop will provide an introduction to RESTful Web Service techniques, both from a theoretical and practical perspectives.

REST in Practice - Full Day Workshop on Web-based Distributed Systems by Ian Robinson and Jim Webber

The Web is fast becoming a serious competitor to traditional enterprise architecture approaches. This full day workshop will provide an introduction to RESTful Web Service techniques, both from a theoretical and practical perspectives.

REST in Practice - Full Day Workshop on Web-based Distributed Systems by Ian Robinson and Jim Webber

The Web is fast becoming a serious competitor to traditional enterprise architecture approaches. This full day workshop will provide an introduction to RESTful Web Service techniques, both from a theoretical and practical perspectives.

A Programmatic Introduction to Neo4j Workshop by Ian Robinson and Jim Webber

Graph databases are an esoteric but powerful member of the NoSQL family. For highly connected data, graph databases can be thousands of times faster than relational databases, making Neo4j popular for managing complex data across many domains, from finance to social, telecoms to geospatial.

A Programmatic Introduction to Neo4j Workshop by Ian Robinson and Jim Webber

Graph databases are an esoteric but powerful member of the NoSQL family. For highly connected data, graph databases can be thousands of times faster than relational databases, making Neo4j popular for managing complex data across many domains, from finance to social, telecoms to geospatial.

A Programmatic Introduction to Neo4j Workshop by Ian Robinson and Jim Webber

Graph databases are an esoteric but powerful member of the NoSQL family. For highly connected data, graph databases can be thousands of times faster than relational databases, making Neo4j popular for managing complex data across many domains, from finance to social, telecoms to geospatial.

A Programmatic Introduction to Neo4j Workshop by Ian Robinson and Jim Webber

Graph databases are an esoteric but powerful member of the NoSQL family. For highly connected data, graph databases can be thousands of times faster than relational databases, making Neo4j popular for managing complex data across many domains, from finance to social, telecoms to geospatial.

Driving Technical Change by Terry Ryan

Ever been to a conference, get inspired, try to bring what you learned back to the office, only to be stymied by co-workers who aren't interested in rocking the status quo? It turns out that people tend to resist change in patterns, and like any pattern they can be overcome by using other people's experiences with those skeptics. This session will teach you how to identify the skeptics, how to counter them, and give you a strategic framework to convince your whole office.

Design for the Developer by Terry Ryan

"That's really useful, but it looks like it was designed by a developer." Ever heard that? Want to fix it? Think you don't have design ability? Here's a dirty little secret, design is a skill, it can be learned. This session will take you through the basics of design theory for applications. By the end you should be on your way to building not just useful apps that people have to use, but awesome apps that people love to use.

JBoss Drools: Rule Engine Development in Java by Brian Sam-Bodden

This workshop is aimed at Java and Java EE developers looking to understand and apply a Rule Engine to solve problems typically and painfully addressed with traditional programming techniques.

JBoss Drools: Rule Engine Development in Java by Brian Sam-Bodden

This workshop is aimed at Java and Java EE developers looking to understand and apply a Rule Engine to solve problems typically and painfully addressed with traditional programming techniques.

Java/Groovy Cloud Computing by Brian Sam-Bodden

In this workshop you will learn how to design, develop and deploy Java and Groovy applications on the Cloud. Learn about GAE (Google App Engine) and Amazon EC2 (Elastic Compute Cloud)

Java/Groovy Cloud Computing by Brian Sam-Bodden

In this workshop you will learn how to design, develop and deploy Java and Groovy applications on the Cloud. Learn about GAE (Google App Engine) and Amazon EC2 (Elastic Compute Cloud)

Tools and Techniques to build Smart Java Applications by Brian Sam-Bodden

In this session we will explore the Java tools, techniques and algorithms that enable us to filter, classify, relate and discover patterns in our data that might not immediately obvious. With the emergence of social networking applications a great deal of data and hidden connections that can be leveraged to build better and smarter applications.

Clojure: Lisp for the Real World by Stuart Sierra

Clojure is a new dynamic programming language for the Java Virtual Machine. Clojure introduces innovative ideas around state management and concurrency, while inheriting the best ideas from the long history of Lisp-like languages. It is a language designed to solve real problems, some of which are so pervasive in current programming practice that we don't even recognize them as problems.

Rethinking Object-Oriented: Clojure and the Expression Problem by Stuart Sierra

What is the Expression Problem? The question is far from academic: any programmer working in mainstream object-oriented languages is bound encounter it. As a young language on the JVM, Clojure has the opportunity to step back from mainstream approaches to object-oriented design, and get back to core concepts like *type* and *polymorphism*. Clojure, while not an object-oriented language itself, offers features that can achieve the same goals as OOP with greater flexibility.

Search Engine on a Shoestring: Clojure, Hadoop, Solr, and EC2 by Stuart Sierra

In 2006, Columbia law professor Tim Wu asked the question, "Why isn't legal research as easy as searching the web?" Out of that question came AltLaw, a free, open-source search engine for federal court decisions. With a nonexistent budget and only one full-time programmer, AltLaw built a search engine for over 700,000 documents by leveraging a powerful new programming language, Clojure, in conjunction with Hadoop for data processing, Solr/Lucene for search, and Amazon Web Services for infrastructure.

Completing the circle - Automated web tests as a team communication tool by John Smart

Acceptance Test Driven Development, or ATDD, has proven to be a very effective technique, both for driving and guiding development, and for enhancing communication between developers and other project stakeholders. But why stop there? Well designed Acceptance Tests can also act as a formidable documentation source and communication tool. Indeed, when written in a narrative, BDD-type style, Acceptance Tests have the potential to document in detail how the user interacts with the application.

JUnit Kung Fu: Getting More Out of Your Unit Tests by John Smart

JUnit is the de facto standard in Java testing. Yet many advanced JUnit features are little known and poorly understood. This session reviews some lesser-known features of JUnit, along with a few associated libraries, that can make your unit tests more powerful, expressive, and fun. The session is intended for Java developers, lead developers, and architects trying to introduce good testing practices into their teams.

Enterprise Messaging With Spring and ActiveMQ: by Bruce Snyder

Spring greatly simplifies JMS messaging by handling common scenarios for you by providing facilities for both synchronous and asynchronous messaging. This dramatically lowers the barrier to building message-driven applications. Apache ActiveMQ is an open source JMS message broker that provides client

access from many different languages and offers many advanced features necessary for enterprise level messaging.

ActiveMQ In Action: Common Problems and Solutions by Bruce Snyder

Why does ActiveMQ just stop sending messages and hang? What's the best way to build a JMS consumer and producer? What configuration should I use for connection pooling? How can I query ActiveMQ for the message I need? Should I cluster ActiveMQ or group clients across brokers? If you use ActiveMQ, chances are you have run into some questions that are easily answered with a little knowledge.

Messaging and Concurrency in Your Applications Using Spring by Bruce Snyder

Most Spring-based applications utilize a design based on layering. When using the standard layered application approach, the service layer is commonly used to encapsulate reusable, business-specific logic. Furthermore, communication between these services has taken place via synchronous method invocations. Spring also provides support for a message-driven communication and concurrent task messaging and concurrency support in Spring for your applications.

Styles of Application Integration Using Spring by Bruce Snyder

Different integration scenarios require different types of application integration styles. Knowledge of these integration types will help you understand how to add integration to your applications using Spring.

REST in Practice - Full Day Workshop on Web-based Distributed Systems by Jim Webber

The Web is fast becoming a serious competitor to traditional enterprise architecture approaches. This full day workshop will provide an introduction to RESTful Web Service techniques, both from a theoretical and practical perspectives.

Intro to Hadoop MapReduce - Indepth by Chris Wensel

This talk will introduce the Hadoop MapReduce model and common patterns and algorithms implemented to solve common problems.

Hadoop Architecture - In Depth by Chris Wensel

Hadoop architecture with discussion on how the MapReduce model influenced it.

Cascading and Common Big Data Problems by Chris Wensel

This session will quickly introduce the Cascading open-source project and how it was used in various projects to overcome problems and bottlenecks particular to large data analytics.

Cascading and Common Big Data Problems by Billy Williams

This session will quickly introduce the Cascading open-source project and how it was used in various projects to overcome problems and bottlenecks particular to large data analytics.