



Come to the Ü for an Extreme Tech Experience

Über Conf will take place June 14 - 17, 2010 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.**

» Hands On Workshops

At *Über Conf* you will not just listen to lectures. You will have the opportunity to participate in workshops, get your hands dirty, and write code.

» Learn from the Best

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

» Rates

Take advantage of our all inclusive travel package. This package includes: conference registration, 3 nights lodging, and airfare in the continental US.

	Register By	Price	All-Inclusive Price
Super Earlybird Rate	Monday, April 12	\$1,250	\$2,100
Earlybird Rate	Friday, May 14	\$1,400	\$2,250
Regular Rate		\$1,550	\$2,400

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

- Languages on the JVM
- Security
- SOA and ROA
- Enterprise Java
- Cloud Computing
- Java Internals
- Agility
- Mobile Dev (iPhone, Android)

Featured Speakers



Alex Antonov
Technical Lead on the Core Frameworks team at Orbitz Worldwide



Tim Berglund
Developer, Consultant, Author



Hans Dockter
Founder and Project Lead of Gradle



Neal Ford
Application Architect at ThoughtWorks, Inc.



Brian Goetz
Author of Java Concurrency in Practice



David Hussman
Agility Coach/Instructor /Practitioner

Über Conf

Westin Westminster

June 14 - 17, 2010

Mon, Jun. 14, 2010								
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Windsor	Waverly
12:00 - 1:00 PM	EARLY REGISTRATION - UBERCONF							
1:00 - 5:00 PM		iPad/iPhone Seminar - Part I Matthew McCullough and Ben Ellingson						
5:00 - 6:30 PM	MAIN UBERCONF REGISTRATION							
6:30 - 7:30 PM	DINNER							
7:30 - 8:30 PM	Keynote: by Cliff Click							
8:30 - 10:00 PM	OPENING NIGHT RECEPTION							

Tue, Jun. 15, 2010								
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Windsor	Waverly
7:00 - 8:00 AM	5K FUN RUN & POWER WALK							
7:30 - 8:30 AM	BREAKFAST & LATE REGISTRATION							
8:30 - 10:00 AM	Encryption on the JVM: Boot Camp Matthew McCullough	Mastering Spring MVC 3 Keith Donald	Patterns of Modular Architecture Kirk Knoernschild	Cloud computing deep dive for Google App Engine and Amazon EC2 Rohit Bhardwaj	Transforming to Groovy Venkat Subramaniam	Hydras and Hypermedia Ian Robinson	Agility as a Tool: Getting Ready to Iterate David Hussman	Test Driven Development in Java: Live and Uncensored Ben Rady
10:00 - 10:30 AM	MORNING BREAK							
10:30 - 12:00 PM	Java EE 6 & GlassFish v3: Paving the path for the future Arun Gupta	Inside Spring Web Flow 3 Development Keith Donald	Common AntiPatterns and How To Avoid Them Mark Richards	Practical Agile Database Development Tim Berglund	Groovy Power Features Paul King	The Counterintuitive Web Ian Robinson	Products and People over Process and Dogma David Hussman	Enterprise testing to make your application foolproof Rohit Bhardwaj
12:00 - 12:30 PM	OUTDOOR BREAK							
12:30 - 1:30 PM	LUNCH							
1:30 - 3:00 PM	Architect for Scale Michael Nygard	Extreme Productivity with Roo Keith Donald	Complexity Theory and Software Development Tim Berglund	The Busy Java Developer's Guide to Collections Ted Neward	Groovy Power Features (continued) Paul King	Redesigning Agility: Incorporating Design Thinking David Hussman	The Art of (Java) Benchmarking Cliff Click	Enterprise Security, Privacy and Data compliance Rohit Bhardwaj
3:00 - 3:15 PM	BREAK							
3:15 - 4:45 PM	Resource Oriented Architecture using REST Alex Antonov	Hadoop Workshop Matthew McCullough	Developing Web Applications with Spring Keith Donald	The Busy Java Developer's Guide to Functional Java Ted Neward	Automated deployment with Maven and friends - going the whole nine yards John Smart	Agile Velocity Ken Sipe	Fast Bytecodes for Funny Languages Cliff Click	High Performance Persistence with Redis Michael Nygard
4:45 - 5:00 PM	BREAK							
5:00 - 6:30 PM	Building RESTful ROA Architecture at Orbitz Alex Antonov	Hadoop Workshop (continued) Matthew McCullough	Developing Web Applications with Spring (continued) Keith Donald	The Busy Java Developer's Guide to Advanced Collections Ted Neward	Zen and the Art of Build Script Maintenance John Smart	Dynamic Languages Practices Paul King	Architecture: Non-Functional Requirements Ken Sipe	Continuous Testing on the JVM Ben Rady
6:30 - 8:30 PM	DINNER/OUTDOOR BREAK/PANEL DISCUSSION							
8:30 - 10:00 PM	So you want to be an Architect Ken Sipe	iPad/iPhone Seminar - Part II Matthew McCullough and Ben Ellingson	BJDG-Games Ted Neward	The Art of Messaging Mark Richards	Coding Dojo with John Smart John Smart	When the Fur Flies: Dev and Ops Cooperation when the Worst Happens Michael Nygard	Visualizations for Code Metrics Neal Ford	Building a language on the JVM Ola Bini

Wed, Jun. 16, 2010								
	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Windsor	Waverly
8:00 - 9:00 AM	BREAKFAST							
9:00 - 10:30 AM	Implementing Evolutionary Architecture Neal Ford	Using Apache Camel Mark Richards	Git Source Code Control Workshop Matthew McCullough	Scala for Java Programmers Venkat Subramaniam	Grails: Bringing Radical Productivity to the JVM Dave Klein	A Crash Course in Modern Hardware Cliff Click	Iteration-less, Value-Based Planning Ben Rady	RDFA : Weaving Richness and Meaning in the Web Brian Sletten
10:30 - 11:00 AM	MORNING BREAK							
11:00 - 12:30 PM	Agile Engineering Practices Neal Ford	Turtles and Architecture Kirk Knoernschild	Git Source Code Control Workshop (continued) Matthew McCullough	Scala for Java Programmers (continued) Venkat Subramaniam	Grails: Bringing Radical Productivity to the JVM (continued) Dave Klein	BJDG-Parallelization Patterns Ted Neward	Design for Operations Michael Nygard	NetKernel: Making IT Matter Again Brian Sletten
12:30 - 1:30 PM	LUNCH							
1:30 - 3:00 PM	End-to-End Performance Optimization Aleksandar Seovic	Agile Tools - Taking Your Agile Practices To The Next Level Paul King	Emergent Design Neal Ford	An in depth look at Apache Wicket Andrew Lombardi	JRuby for the win Ola Bini	DSLs in Scala: Internal and External Michael Nygard	Introduction to Gradle Hans Dockter	NetKernel: Making IT Matter Again (continued) Brian Sletten
3:00 - 3:15 PM	BREAK							
3:15 - 4:45 PM	Stability Antipatterns Michael Nygard	Continuous Integration - Hudson John Smart	Pragmatic Architecture Ted Neward	Hands on with Apache Wicket Andrew Lombardi	JRuby in Depth Ola Bini and Neal Ford	Resource-Oriented Concurrent Processing Jeremy Deane	Gradle Hans Dockter	Working with Complex Adaptive (Human) Systems Esther Derby
4:45 - 5:00 PM	BREAK							
							Gradle (continued)	

Wed, Jun. 16, 2010

	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Windsor	Waverly
5:00 - 6:30 PM	Stability Antipatterns (continued) Michael Nygard	Continuous Integration - Hudson (continued) John Smart	Pragmatic Architecture (continued) Ted Neward	Hands on with Apache Wicket (continued) Andrew Lombardi	JRuby in Depth (continued) Ola Bini and Neal Ford	Resource-Oriented Enterprise Service Bus Jeremy Deane	Hans Dockter	The New Work of management in Agile Organizations Esther Derby
6:30 - 7:30 PM	DINNER							
7:30 - 9:00 PM	Performance and Scalability Revisited: In-Memory Data Grids Aleksandar Seovic	iPad/iPhone Seminar - Part III Matthew McCullough	How to Approach Refactoring Venkat Subramaniam	Successful Software Management: 17 Lessons Learned Johanna Rothman	From Java to Ruby and Back Alex Antonov	Business Architecture Foundations of IT Ian Robinson	Grails Integration Strategies Dave Klein	Developing a Complex External DSL Vaughn Vernon
9:00 - 10:30 PM	UBERCONF PARTY							

Thu, Jun. 17, 2010

	Westminster I	Westminster II	Standley I	Standley II	Cotton Creek I	Cotton Creek II	Windsor	Waverly
8:00 - 9:00 PM	BREAKFAST							
9:00 - 10:30 AM	Hacking - The Dark Arts Ken Sipe	iBeans: The Simplest Service Integrations You've Ever Implemented Matthew McCullough	Introduction to Functional Programming with Clojure Stuart Halloway	TDD Regular Code/ Multithreaded Code! Venkat Subramaniam	XML and Web Services with Groovy Paul King	Getting the best of both worlds: OSGi & Java EE together Arun Gupta	Manage Your Project Portfolio: A Lean and Agile Approach Johanna Rothman	Implementing Domain-Driven Designs Vaughn Vernon
10:30 - 10:45 AM	MORNING BREAK							
10:45 - 12:15 PM	Security Boundaries Ken Sipe	Migrating to Maven 3.0 Matthew McCullough	The Best of OO: Clojure Types and Protocols Stuart Halloway	TDD Regular Code/ Multithreaded Code! (continued) Venkat Subramaniam	Gaelyk: Cloud-Based Apps With Groovy Tim Berglund	Solr - Case Study Eric Pugh	Agile Program Management: Another Approach to Large Projects Johanna Rothman	The Domain-Driven Design Metamodel Vaughn Vernon
12:15 - 1:30 PM	OUTDOOR BREAK & LUNCH							
1:30 - 3:00 PM	Security Code Review Ken Sipe	Android mobile application development Rohit Bhardwaj	Clojure Stuart Halloway	Semantic Web Workshop Brian Sletten	ATDD/BDD with Cucumber Paul Rayner	Solr Eric Pugh	Coaching as a Two-Way Relationship Johanna Rothman	Open Source Business Intelligence - Part I Tim Berglund
3:00 - 3:15 PM	AFTERNOON BREAK							
3:15 - 4:45 PM	Enterprise Security API library from OWASP Ken Sipe	Android mobile application development (continued) Rohit Bhardwaj	Clojure (continued) Stuart Halloway	Semantic Web Workshop (continued) Brian Sletten	ATDD/BDD with Cucumber (continued) Paul Rayner	Solr (continued) Eric Pugh	Coaching as a Two-Way Relationship (continued) Johanna Rothman	Open Source Business Intelligence - Part II (continued) Tim Berglund
4:45 - 5:00 PM	UBERCONF - CONCLUSION							

Über Conf

Westin Westminster

June 14 - 17, 2010

iPad/iPhone Seminar - Part I by Matthew McCullough and Ben Ellingson

You're adept at Java. You've got a solid grasp of that ecosystem. But you keep hearing about iPhone this, iPad that. It worries you just a bit that you haven't yet spent the time to explore this new frontier. Cure that with a nearly Java-free intense four hour boot camp on the iPad. During this session, you'll use your UberConf-supplied device to get started coding on the XCode platform, learning the Objective-C language, testing, and deploying your apps.

Encryption 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.

Hadoop Workshop by Matthew McCullough

Moore's law has finally hit the wall and CPU speeds have actually decreased in the last few years. The industry is reacting with hardware with an ever-growing number of cores and software that can leverage "grids" of distributed, often commodity, computing resources. But how is a traditional Java developer supposed to easily take advantage of this revolution? The answer is the Apache Hadoop family of projects. Hadoop is a suite of Open Source APIs at the forefront of this grid computing revolution and is considered the absolute gold standard for the divide-and-conquer model of distributed problem crunching. The well-travelled Apache Hadoop framework is currently being leveraged in production by prominent names such as Yahoo, IBM, Amazon, Adobe, AOL, Facebook and Hulu just to name a few.

Hadoop Workshop by Matthew McCullough

Moore's law has finally hit the wall and CPU speeds have actually decreased in the last few years. The industry is reacting with hardware with an ever-growing number of cores and software that can leverage "grids" of distributed, often commodity, computing resources. But how is a traditional Java developer supposed to easily take advantage of this revolution? The answer is the Apache Hadoop family of projects. Hadoop is a suite of Open Source APIs at the forefront of this grid computing revolution and is considered the absolute gold standard for the divide-and-conquer model of distributed problem crunching. The well-travelled Apache Hadoop framework is currently being leveraged in production by prominent names such as Yahoo, IBM, Amazon, Adobe, AOL, Facebook and Hulu just to name a few.

iPad/iPhone Seminar - Part II by Matthew McCullough and Ben Ellingson

Workshop part II will focus on debugging. First, we will revisit deployment to iPad / iPhone devices, cover common application debugging techniques, writing unit tests, and debugging tools. We will also include time for you to ask questions and work through any issues you are having.

Git Source Code Control Workshop by Matthew McCullough

You've heard about Git, Mercurial, Bazaar and the Distributed Version Control System revolution. In this deeply hands on session, we'll load Git on participants laptops, build repositories and share out pieces of work. We'll explore the optimized agile workflows that Git facilitates, building branches for each story card and merging with our team mates, even when a network isn't present. We'll clone an existing Subversion repository, work on it in a Git fashion, and push just the "good changes" back to Subversion, showcasing the incredibly polished interoperability of this radical source code control tool.

Git Source Code Control Workshop by Matthew McCullough

You've heard about Git, Mercurial, Bazaar and the Distributed Version Control System revolution. In this deeply hands on session, we'll load Git on participants laptops, build repositories and share out pieces of work. We'll explore the optimized agile workflows that Git facilitates, building branches for each story card and merging with our team mates, even when a network isn't present. We'll clone an existing Subversion

repository, work on it in a Git fashion, and push just the "good changes" back to Subversion, showcasing the incredibly polished interoperability of this radical source code control tool.

iPad/iPhone Seminar - Part III by Matthew McCullough

Workshop part III will focus on design. The iPad isn't simply a large iPhone. The technology is pretty much the same, but a quick look at some of the new apps and you will see how cool this new device and software platform is. In this session, we will talk about Apple's Human Interface Guidelines and what's new on the iPad platform. We will also provide more time for you to ask questions and work through any issues you may be having.

iBeans: The Simplest Service Integrations You've Ever Implemented by Matthew McCullough

No app is an island nowadays and your bleeding edge Java & JavaScript apps demand that you integrate with Facebook, Amazon, Gmail, Google Search, Twitter or S3 just to name a few. Make your next integration project a breeze by leveraging the successful work of others from the iBeans Central repository, or if necessary, simply author a new iBean and contribute it back for the benefit of all. iBeans a new ultra-light service integration framework written in Java, but targeting both Java and JavaScript. It provides a centralized mechanism for community contributions of beans to the most commonly used services such as Twitter, Flickr, Gmail and more.

Migrating to Maven 3.0 by Matthew McCullough

Explore what's new on the cutting edge release of Maven, version 3.0. We'll explore the performance improvements, features that make debugging Maven issues easier, and changes to POMs that may require modifications to your build, but will result in more determinate build outputs.

Resource Oriented Architecture using REST by Alex Antonov

A presentation explaining the building blocks necessary to build a distributed system using RESTful-style resource definition as well as benefits of using REST as they apply to testing, data caching, predictable resource discovery and transparent future system evolution

Building RESTful ROA Architecture at Orbitz by Alex Antonov

In the beginning Orbitz had a Jini based distributed system. The system design provided easy scalability and stability, but at the cost of tight coupling because of many shared modules and components, as well as Java serialization rules. In order to improve cohesion between individual services the decision has been made to migrate to a RESTful web services architecture. The new design is based on Google Protocol Buffers to define message formats and Spring/Spring MVC to handle client-server interaction. This resulted in a loosely coupled federation of services, each with its individual release and deployment schedule, which enabled more developer innovation and easier access to more data in a uniform fashion.

From Java to Ruby and Back by Alex Antonov

A presentation demoing clients and services written in different languages (Java & Ruby), demonstrating technologies that enable distributed systems to span languages and provide an automated way of returning alternative data representations (like XML, JSON, Protobuf, etc.) for different clients, while using the same backing data.

Practical Agile Database Development by Tim Berglund

Do your team's agile practices extend to the database? Agile methods are fairly well-understood as they apply to code, but these principles are not commonly understood or practiced on the databases that typically accompany enterprise software projects. Learn the tools, techniques, and mindset your team needs to make incremental improvements to the database's design over time with confidence.

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.

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.

Open Source Business Intelligence - Part I by Tim Berglund

Traditionally, business intelligence tools have been a high-cost part of any enterprise's software inventory. Recently, options have emerged that allow architects to build a credible business intelligence stack out of entirely open-source components. In this brief overview, we will demonstrate ETL, reporting, and analytics tool that can be deployed free or at low cost. Learn how to turn your company's transactional database into a rich data asset with a business-friendly user interface that integrates into your existing software infrastructure.

Open Source Business Intelligence - Part II by Tim Berglund

Once you're familiar with the concepts of data warehousing, star schemas, cubes, and pivot tables, then it's time to dive in and look at how the tools really work. Continuing from the quick demos in Part I, in this talk we'll walk through the process of transforming a transactional database into a star schema, then we'll use an open-source analytics tool to build a "cube" with that schema. Concepts and procedures gently introduced in Part I will be explored more thoroughly, and new tooling will be introduced. **Prerequisite:** *Open Source Business Intelligence Part I (or a solid grasp of BI concepts)*

Cloud computing deep dive for Google App Engine and Amazon EC2 by Rohit Bhardwaj

In this session we will take a deep dive at few cloud computing examples from real world and participants will be able to know how to use cloud computing for Google App Engine, Amazon EC2 and few others.

Enterprise testing to make your application foolproof by Rohit Bhardwaj

Enterprise software solutions are an essential part of many large enterprises. Given the critical role enterprise software solutions play, it is imperative that they are tested effectively and efficiently all the time. It is as important, if not the most important, as any other phase of the Software Development Lifecycle (SDLC). But testing an enterprise application is easier said than done. This presentation targets seasoned software developers, testers and project managers who are looking for guidance in implementing an effective application testing strategy. We will discuss the rationale behind application enterprise testing and explore building blocks of effective testing and explain their importance. Then we will explore how to do effective root-cause analysis. We will discuss the typical output of a performance test and how to perform effective analysis. We will learn the effects of particular software environments on testing. The approach is generic; so many details regarding your applications will depend on the characteristics of the technologies you use. Later on we will explore at two tools PushToTest and CloudTest to automatically test web applications. Attendees will learn different test strategies for testing.

Enterprise Security, Privacy and Data compliance by Rohit Bhardwaj

Data integrity, security, recovery, privacy and regulatory compliance are most important attributes for enterprise implementation. Enterprise customers ask for transparency in how the vendors will provide security programs. Many question need to be asked for any cloud implementation to policy makers, architects, coders and testers. In this presentation we will explore data security and storage, privacy and data compliance issues. We will explore the security management in cloud. Presentation is useful for anyone starting from Executives to developers who are going to implement the enterprise Applications in both private and public cloud.

Android mobile application development by Rohit Bhardwaj

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Cool apps that surprise and delight mobile users—built by developers like you—are a huge part of the Android vision. In this presentation we will explore many examples of android. The Android SDK provides the tools and APIs necessary to begin developing applications that run on Android-powered devices. Cool apps that surprise and delight mobile users—built by developers like you—are a huge part of the Android vision. Google has also participated in the Android Market by offering several applications for its services. These applications include Google Voice for the Google Voice service, Scoreboard for following sports, Sky Map for watching stars, Finance for their finance service, Maps Editor for their MyMaps service, Places Directory for their Local Search, Google Goggles that searches by image, and My Tracks, a jogging application. Android phones that include the 'Google Experience' also have Google Search, Google

Calendar, Google Maps, Google Navigation and Gmail integrated. In this workshop we will explore many examples of android.

Android mobile application development by Rohit Bhardwaj

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Cool apps that surprise and delight mobile users—built by developers like you—are a huge part of the Android vision. In this presentation we will explore many examples of android. The Android SDK provides the tools and APIs necessary to begin developing applications that run on Android-powered devices. Cool apps that surprise and delight mobile users—built by developers like you—are a huge part of the Android vision. Google has also participated in the Android Market by offering several applications for its services. These applications include Google Voice for the Google Voice service, Scoreboard for following sports, Sky Map for watching stars, Finance for their finance service, Maps Editor for their MyMaps service, Places Directory for their Local Search, Google Goggles that searches by image, and My Tracks, a jogging application. Android phones that include the 'Google Experience' also have Google Search, Google Calendar, Google Maps, Google Navigation and Gmail integrated. In this workshop we will explore many examples of android.

Building a language on the JVM by Ola Bini

We are getting more and more languages - but sometimes the best solution is actually to build one yourself. Whether you choose to build a small DSL or a fullblown language, many of the techniques are the same.

JRuby for the win by Ola Bini

JRuby is an implementation of the Ruby language for the JVM. It gives full access to the Ruby language while running on top of Java. JRuby is the fastest and most fully-featured implementation of Ruby available.

JRuby in Depth by Ola Bini and 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 Ola Bini and 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.

Challenges and Directions in Java Virtual Machines by Cliff Click

Available core counts are going up, up, up! Intel is shipping quad-core chips; Sun's Rock has (effectively) 64 CPUs and Azul's hardware nearly a thousand cores. How do we use all those cores effectively? The JVM proper can directly make use of a small number of cores (JIT compilation, profiling), and garbage collection can use about 20 percent more cores than the application is using to make garbage--but this hardly gets us to four cores. Application servers and transactional—J2EE/bean--applications scale well with thread pools to about 40 or 60 CPUs, and then internal locking starts to limit scaling. Unless your application has embarrassingly parallel data (e.g. data mining; risk analysis; or, heaven forbid, Fortran-style weather-prediction), how can you use more CPUs to get more performance? How do you debug the million-line concurrent program?

The Art of (Java) Benchmarking by Cliff Click

People write toy Java benchmarks all the time. Nearly always they "get it wrong" -- wrong in the sense that the code they write doesn't measure what they think it does. Oh, it measures something all right -- just not what they want. This session presents some common benchmarking pitfalls, demonstrating pieces of real, bad (and usually really bad) benchmarks. The session is for any programmer who has tried to benchmark anything. It provides specific advice on how to benchmark, stumbling blocks to look out for, and real-world examples of how well-known benchmarks fail to actually measure what they intended to measure.

Fast Bytecodes for Funny Languages by Cliff Click

There are several languages that target bytecodes and the JVM machine as their new "assembler," including Scala, Clojure, Jython, JRuby, the JavaScript programming language/Rhino, and JPC. This session takes a quick look at how well these languages sit on a JVM machine, what their performance is, where it goes, and why.

A Crash Course in Modern Hardware by Cliff Click

I walk through a tiny performance example on a modern out-of-order CPU, and basically show that (1) single-threaded performance is tapped out, (2) all the action is with multi-threaded programs and (3) the memory subsystem.

Resource-Oriented Concurrent Processing by Jeremy Deane

The majority of applications built on Java Platform in the past decade have been single-threaded. And the easiest way to address performance issues was to add a faster CPU. However, that option is no longer viable as the hardware industry has shifted from single-core processors to multi-core processors. Traditional concurrent development on the Java Platform requires in depth knowledge of threads, locks, and queues (oh, my!). Fortunately, new functional languages that run on the Java Platform, such as Scala, have made concurrent programming easier. An alternate approach is to implement concurrent processes using a resource oriented computing (ROC) platform. At the heart of this ROC platform is a microkernel that allows processing to scale linearly as more CPUs are added. Consequently, developers are freed from the complexity of Java concurrency and functional programming.

Resource-Oriented Enterprise Service Bus by Jeremy Deane

An Enterprise Service Bus (ESB) provides a platform for service provisioning. The core capabilities that enable provisioning across an enterprise include addressing, routing and transformations. Addressing is the ability to specify the location of a service regardless of transport. Service routing defines a message path across a number of servers or nodes and message content transformations are implemented using XML technologies such as XSLT and proprietary adapters. Resource Oriented Architecture (ROA) goes beyond RESTful web services and provides a more extensible transport-independent foundation. Furthermore, ROA pushes the integration functionality to the edge of the network (as a URI), translating into better service management and scalability.

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.

The New Work of management in Agile Organizations by Esther Derby

Sometimes I see teams that reject all direction and go their own way, declaring, 'We are self-organizing'. They are missing an important fact. When someone is paid by a company to be part of a team, that team exists within the organizational context. On the other hand, some managers hear the words "self-organizing" and believe the team is on its "own" that they can go into semi-retirement. But that's not the case, either. In fact, both are risky over-simplifications. When teams self-organize there's still plenty for managers to do, but management attention must shift from individual to team performance, and creating an environment where teams can excel.

Introduction to Gradle by Hans Dockter

Gradle allows you to describe your build using a rich, extensible build language based on Groovy. It provides exciting solutions for many of the big pain points that exist with current build systems. This session will be mostly driven by live demos. You will see how easy and elegant Gradle enables you to solve a broad range of requirements - over the full life cycle of typical and untypical Java builds.

Gradle by Hans Dockter

In this workshop, you will become familiar with all major concepts of Gradle and how to best use Gradle for simple as well as complex build scenarios. You will learn about the basic language elements of the Gradle Domain Specific Language (DSL), how to use Gradle's build-by-convention for plain Java and Java web projects, and how to use the Gradle plugin system. You will find out how easy it is to customize the very guts of Gradle for your build requirements. You will also learn about Gradle's smart performance optimization strategies like incremental builds and parallel testing. Some code is initially easier to understand with a basic

understanding of Groovy. But due to Groovy's similarity to Java, Groovy is not a prerequisite. There will be a very short Groovy intro at the beginning.

Gradle by Hans Dockter

In this workshop, you will become familiar with all major concepts of Gradle and how to best use Gradle for simple as well as complex build scenarios. You will learn about the basic language elements of the Gradle Domain Specific Language (DSL), how to use Gradle's build-by-convention for plain Java and Java web projects, and how to use the Gradle plugin system. You will find out how easy it is to customize the very guts of Gradle for your build requirements. You will also learn about Gradle's smart performance optimization strategies like incremental builds and parallel testing. Some code is initially easier to understand with a basic understanding of Groovy. But due to Groovy's similarity to Java, Groovy is not a prerequisite. There will be a very short Groovy intro at the beginning.

Mastering Spring MVC 3 by Keith Donald

A deep-dive into the latest capabilities of Spring MVC, Spring's foundational web application development platform.

Inside Spring Web Flow 3 Development by Keith Donald

A look inside the development of Spring Web Flow 3, the next-generation version of Spring's stateful controller framework for orchestrating multi-step user dialogs.

Extreme Productivity with Roo by Keith Donald

A deep-dive into Spring Roo, Spring's breakthrough code generation technology with support for round-tripping.

Developing Web Applications with Spring by Keith Donald

Summary: A three-hour, two-part workshop on building modern web applications with the Spring stack.

Detail: In part I, Keith will first provide a brief overview of what Spring offers web application developers. Then, from the ground up, he will take attendees through the process of designing and implementing a web application with the framework. Attendees will not only learn the Spring feature set, they will learn to apply the features effectively. Whether you are new to Spring, or an experienced Java developer who has used Spring before, this workshop will help you grow as an application developer.

Developing Web Applications with Spring by Keith Donald

Summary: A three-hour, two-part workshop on building modern web applications with the Spring stack.

Detail: In part I, Keith will first provide a brief overview of what Spring offers web application developers. Then, from the ground up, he will take attendees through the process of designing and implementing a web application with the framework. Attendees will not only learn the Spring feature set, they will learn to apply the features effectively. Whether you are new to Spring, or an experienced Java developer who has used Spring before, this workshop will help you grow as an application developer.

Visualizations for Code Metrics by Neal Ford

Judicious use of metrics improves the quality of your code. But interpreting metrics presents a challenge.

You have a list of numbers for a project - what does it mean? And what does it tell me about the health of the project overall? This sessions shows how to produce visualizations for software metrics, making them easier to understand and more valuable. It covers metrics at the individual method level all the way up to the overall architecture of the application. This isn't just a talk about how some tools produce visualizations: this session shows you how to generate your own visualizations, allowing you to customize it to the level in information density that shows real value on your project. I show how to produce projected graphs from dependencies, heat-maps for cyclomatic complexity and code coverage, using XSLT to extract visual information from XML configuration documents, and others. Metrics can't help you if you can't understand them. By creating visualizations, it helps leverage metrics to make your code better.

Implementing Evolutionary Architecture by Neal Ford

This talk describes an agile approach to architecture, and merges the current state-of-the-art thinking in both service oriented architectures(SOA) and web-based architectures like HTTP, REST, and hypermedia.

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*

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*

Java EE 6 & GlassFish v3: Paving the path for the future by Arun Gupta

The Java EE 6 platform is an extreme makeover from the previous versions. It is developed as JSR 316 under the Java Community Process. The Java EE 6 platform adds more power to the platform and yet make it more flexible so that it can be adopted to different flavors of an application. It breaks the "one size fits all" approach with Profiles and improves on the Java EE 5 developer productivity features. It enables extensibility by embracing open source libraries and frameworks such that they are treated as first class citizens of the platform. Several new specifications such as Java Server Faces 2.0, Servlet 3.0, Java Persistence API 2.0, and Context and Dependency Injection 1.0 are included in the platform. All these specifications are implemented in GlassFish v3 that provides a light-weight, modular, and extensible platform for your Web applications.

Getting the best of both worlds: OSGi & Java EE together by Arun Gupta

OSGi defines a module system and service platform for the Java language. GlassFish is the Java EE 6 Reference Implementation and uses an OSGi kernel to create a light-weight and modular Application Server. There is a lot of activity in the Enterprise Expert Group of OSGi about use of OSGi in enterprise Java environment. GlassFish is a container for standard Java EE applications and also supports what is called a "hybrid application". A hybrid application is a Java EE application as well as an OSGi bundle. It allows application components such as Servlets, EJBs to take full advantage of: - Features such as modularity/dependency management, service dynamism, etc. provided by OSGi service platform. - Services such as transaction management, security, persistence, etc. offered by the Java EE platform.

Introduction to Functional Programming with Clojure by Stuart Halloway

Functional programming has many advantages. For starters: * Pure functions have no dependence on context, so they are easy to write, easy to test, easy to read. Most importantly, they are easy to combine. * Referential transparency makes substitution, caching, and lazy evaluation possible -- and simple. * Careful abstraction makes functions broadly reusable, in contrast with OO 'concretion' which aspires to encapsulation but mostly just protects your data from being reused.

The Best of OO: Clojure Types and Protocols by Stuart Halloway

The best abstraction in OO is the interface: a pure specification that can be implemented polymorphically by many different callees. Clojure's types and protocols generalize this idea to solve the expression problem. In this session you will learn: * What the expression problem is. * How the expression problem hurts your code every day, through wrappers, mixins, adapters, facades, open classes, and monkey patches. * How protocols solve the expression problem. * Ease of use matters: using protocols and types are as simple as classes and interfaces. * Performance matters too: protocols take advantage of the host environment's support for fast polymorphic dispatch.

Clojure by Stuart Halloway

In recent years, the Java community has embraced a variety of new languages that target the JVM, but also offer productivity advantages over traditional Java coding. In this half-day session you will explore Clojure, a language with many compelling features; * Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. * Clojure's sequence library turns the tables on OO, providing a powerful set of verbs that can work with a small, standard set of nouns. * Clojure is a Lisp and gracefully supports Lisp's signature feature: code as data. * Clojure's functional style and support for software transactional memory make it an appealing option for taking advantage of massively parallel hardware. * Clojure's datatypes and protocols generalize the best idea in OO (interfaces) to handle the expression problem. Say goodbye to wrappers, mixins, adapters, and monkey-patches.

Clojure by Stuart Halloway

In recent years, the Java community has embraced a variety of new languages that target the JVM, but also offer productivity advantages over traditional Java coding. In this half-day session you will explore Clojure, a language with many compelling features; * Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. * Clojure's sequence library turns the tables on OO, providing a powerful set of verbs that can work with a small, standard set of nouns. * Clojure is a Lisp and gracefully supports Lisp's signature feature: code as data. * Clojure's functional style and support for software transactional memory make it an appealing option for taking advantage of massively parallel hardware. * Clojure's datatypes and protocols generalize the best idea in OO (interfaces) to handle the expression problem. Say goodbye to wrappers, mixins, adapters, and monkey-patches.

Agility as a Tool: Getting Ready to Iterate by David Hussman

Many people simplistically apply agile recipes, assuming a one size fits all approach. This may lead to naive use beliefs like collocation breeds instant success. While sitting together always helps, it does not mean that people spontaneously collaborate to create sustainable value. Instead of approaching agile methods like a recipe, this session will teach you to design agility that is a useful tool for your project community. We will cover practice selection ideas, tools for creating healthy development eco-systems and product discover tools. If you would like to improve the stickiness of your agility, stop in learn a pile of techniques to use before holding your first planning session.

Products and People over Process and Dogma by David Hussman

The time has come to shift our focus away from process to products and people. 10 years into the agile movement, the fresh, lightweight process once created are gaining weight and often calcifying to a dangerous degree. Where meaningful and lasting agility thrives, agile practices are powerful tools but not the focus of daily discussion.

Redesigning Agility: Incorporating Design Thinking by David Hussman

Design tends to mean one thing to developers and another thing to designers. The later group are product designers and are not aware of the structure of the code. The hipsters in the agile community are trying to blend product design into the process of coding and delivering software. This session discusses what to do after your agility is flowing, or post agilism: imagine you using agile methods successfully, what's next?

Groovy Power Features by Paul King

Groovy is a dynamic language for the JVM; it's like a super version of Java. For Java programmers, it offers a syntax that closely resembles (in some cases exactly resembles) Java, but offers many improvements that not only greatly simplify code but also provide an enriched environment with many productivity features. In many cases, such features are promised in Java versions 7 and later, but they are available today in Groovy. **Prerequisite:** *This is an intermediate to advanced workshop. It assumes that attendees have some previous exposure to Groovy or are accomplished at Java.*

Groovy Power Features by Paul King

Groovy is a dynamic language for the JVM; it's like a super version of Java. For Java programmers, it offers a syntax that closely resembles (in some cases exactly resembles) Java, but offers many improvements that not only greatly simplify code but also provide an enriched environment with many productivity features. In many cases, such features are promised in Java versions 7 and later, but they are available today in Groovy. **Prerequisite:** *This is an intermediate to advanced workshop. It assumes that attendees have some previous exposure to Groovy or are accomplished at Java.*

Dynamic Languages Practices by Paul King

Developer practices for traditional and agile Java development are well understood and documented. But dynamic languages (Groovy, Ruby, and others) change the ground rules. Many of the common practices, refactoring techniques, and design patterns we have been taught either no longer apply or should be applied differently and some new techniques also come into play.

Agile Tools - Taking Your Agile Practices To The Next Level by Paul King

Tools and practices as subscribed by the XP methodology are reasonably well known and used by the majority of agile project teams. As agile teams become more mature, so does their thirst for tools to push them to the next level of productivity.

XML and Web Services with Groovy by Paul King

Groovy provides excellent facilities for parsing and creating XML. As well as providing syntactic sugar on top of traditional Java-based parsing approaches (e.g. SAX, DOM, StAX), it has its own XmlParser and XmlSlurper libraries which support XPath-like expressions at the object level (akin to LINQ in the .Net world). In addition, Groovy's markup builders provide an elegant and efficient way to create and modify XML. Groovy also has various options available for SOAP and RESTful web services. We'll examine the most popular of these.

Grails: Bringing Radical Productivity to the JVM by Dave Klein

The goal of this 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. Rather than try to learn Grails feature by feature, we'll let it unfold as we build the application. We'll begin with a simple application structure that runs right out of the box, then we'll gradually build our application while building our knowledge of Grails.

Grails: Bringing Radical Productivity to the JVM by Dave Klein

The goal of this 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. Rather than try to learn Grails feature by feature, we'll let it unfold as we build the application. We'll begin with a simple application structure that runs right out of the box, then we'll gradually build our application while building our knowledge of Grails.

Grails Integration Strategies by Dave Klein

It's amazing how quickly one can build web applications with Grails in a greenfield environment, but most of us do not have that luxury. We have existing infrastructure and applications that we have to maintain and extend. We have legacy databases (or legacy database administrators) to deal with. Does this mean we cannot benefit from the magic of Grails? No way! The ease of use and productivity of Grails is matched by its power and flexibility. In this session we will discuss some of the ways that Grails can be integrated with the enterprise; EJB/JSF applications, Spring/Hibernate, legacy databases, and even non-Java applications.

Patterns of Modular Architecture by Kirk Knoernschild

Attempts to architect more flexible software often results in the opposite - brittle software fraught with complexity. Something is missing. Complexity is the beast we must tame, and modularity is part of the answer. While modularity is not a new concept, until recently, major platform support for modularity has been lacking.

Turtles and Architecture by Kirk Knoernschild

A little old lady once challenged a well-known scientist's explanation on the structure of the universe, countering that the world is really a flat plate supported on the back of a giant tortoise. The scientist rebutted the little old lady's challenge with one of his own by asking what the tortoise was standing on. The little old lady's sly reply was that it's, "turtles all the way down." So too is software architecture "turtles all the way down".

An in depth look at Apache Wicket by Andrew Lombardi

The model supplied by Java Web Frameworks is broken. As software engineers break away from the shackles of Struts and the false promises of JSF, a new model based on object oriented programming and a clean separation of concerns has emerged with Apache Wicket. The framework has a simple component hierarchy allowing for reusability without pain.

Hands on with Apache Wicket by Andrew Lombardi

Apache Wicket injects fun back into your web application development. The in-depth look went over the components and concepts of Wicket while showing the clear alternative that it provides. Our advanced talk took you through the simple process of interactivity using Wicket's AJAX support and proved that reuse while often promised with other frameworks, is a reality here.

Hands on with Apache Wicket by Andrew Lombardi

Apache Wicket injects fun back into your web application development. The in-depth look went over the components and concepts of Wicket while showing the clear alternative that it provides. Our advanced talk took you through the simple process of interactivity using Wicket's AJAX support and proved that reuse while often promised with other frameworks, is a reality here.

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

For so many Java developers, the `java.util.*` package consists of `List`, `ArrayList`, and maybe `Map` and `HashMap`. But the `Collections` classes are so much more powerful than many of us are led to believe, and all it requires is a small amount of digging and some simple exploration to begin to "get" the real power of the `Collection` classes.

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

Much noise has been made in recent years about functional languages, like `Scala` or `Haskell`, and their benefits relative to object-oriented languages, most notably `Java`. Unfortunately, as wonderful as many of those benefits are, the fact remains that most `Java` developers will either not want or not be able to adopt those languages for writing day-to-day code. Which leaves us with a basic question: if I can't use these functional languages to write production code, is there any advantage to learning about them? The short answer is yes, for the fundamental premise--"I can't use functional code on my `Java` project"--is flawed. `Java` developers can, in fact, make use of functional ideas, and what's better, they don't even have to reinvent them for `Java`--thanks to the `FunctionalJava` library, many of the core primitives--interfaces that serve as base types for creating function values, for example--already exist, ready to be used.

The Busy Java Developer's Guide to Advanced Collections by Ted Neward

Once you've learned the core `Collections` classes, you're done, right? You know everything there is to know about `Collections`, and you can "check that off" your list of `Java` packages you have to learn and know, right?
Prerequisite: Busy Java Developer's Guide to Collections

BJDG-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.

BJDG-ParallelizationPatterns by Ted Neward

Getting the most out of your hardware historically was a question of optimizing native code to take advantage of CPU instructions, and setting the right optimization flags on the compiler. Then, when `Java` came around, it became a matter of simply allowing `Hotspot` to do its thing. But as the rise of multicore CPUs has made its presence felt in the developer mindspace, so has the need to start writing code in a more parallel/concurrent fashion. Unfortunately, this is an area that most `Java` developers have historically tried to avoid (like the `Plague`), so it represents a huge "black hole" for most of them.

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.

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.

Architect for Scale by Michael Nygard

Is your system small, medium, large, or super-size? Is traffic on it's way up? Architecture patterns and structures that work at one scale seldom work across all of them. A communication style that's appropriate for small websites will probably fail badly if you apply it to world-wide networks of computers. Likewise,

structures that work for large-scale systems are probably too complex and expensive to be worth it for small sites.

High Performance Persistence with Redis by Michael Nygard

Redis is one of the fresh crop of "NoSQL" storage solutions. It's a distributed key-value store that knows how to deal with data structures. Oh, and it happens to also be incredibly fast. Like, microseconds per write.

When the Fur Flies: Dev and Ops Cooperation when the Worst Happens by Michael Nygard

There's nothing like a crisis to remove artificial barriers. In this session, Michael will present an experience report about development and operations coming together after a failed launch.

Design for Operations by Michael Nygard

If your software fails in production, nobody will care how great the development project was, or how well the system passed QA. Production operations, the domain of your systems' least-appreciated stakeholders, is where the rubber meets the road. Come learn how to build your systems to thrive in Operations.

DSLs in Scala: Internal and External by Michael Nygard

We're no longer working in a single language. Programming today is about both consuming and creating languages. We've all heard a lot about domain specific languages (DSLs). So much so, in fact, that it seems like more people are talking about it than doing it. In this session, Michael will present a real domain with familiar problems. He'll then demonstrate both an internal DSL and an external DSL that solve the same problem. Along the way, we'll cover fluent interfaces, composing multiple DSLs, and the very cool parser combinators. If you've ever left a DSL talk wondering when we're ever going to get past the "coffee ordering DSL" or the "Waffle House breakfast DSL", then you will want to see this session. *Prerequisite: None.*

Stability Antipatterns by Michael Nygard

In this workshop, you will learn how to create applications that survive the rigors of life in production. Too often, project teams aim to pass QA instead of aiming for success in production. Testing is not enough to prove that your software is ready for continuous availability in the corrosive environment of the Internet.

Stability Antipatterns by Michael Nygard

In this workshop, you will learn how to create applications that survive the rigors of life in production. Too often, project teams aim to pass QA instead of aiming for success in production. Testing is not enough to prove that your software is ready for continuous availability in the corrosive environment of the Internet.

Solr - Case Study by Eric Pugh

So you've decided that you need great search, and Solr is going to be the solution? Well implementing a successful search solution based on Solr is a lot more than just downloading some code and firing it up, although Solr does pass the 15 minute test with flying colors! In this case study we'll look at the various pitfalls and challenges involved in migrating a content driven site from Lucene to Solr for the nation's fourth largest retailer. We'll look at some of the decisions we made, from technical, to networking, to operational, and how they impacted the final solution. You'll leave this session know what challenges await a successful migration effort, and what questions to ask to make sure you are successful.

Solr by Eric Pugh

So you're building a website and want a terrific search experience for your users. Do what your database can't: faceted navigation, result highlighting, fuzzy queries, ranked scoring, spell correction, and more. Solr, the open-source enterprise search server is the answer. Solr bridges the technology divide between databases and document/web search engines (e.g. Google). Each has its uses but do not overlap. Chances are you have some structured data, probably in a database, and perhaps some related text documents. When you bring this data into Solr, you'll be able to deliver amazing features. Users will be able to navigate search results by filtering on aggregated attributes (so-called "faceted search"). Furthermore, various features like spell-correcting, auto-completing of search text, boosting records based on various rules, become possible. Solr does not tie you to a particular programming language or computing platform. And whether you have a thousand records or millions and many requests per second, Solr can scale to meet your performance needs. Furthermore, as an open-source solution, Solr doesn't ask you for more money when

you want more out of it. You'll learn the basics of getting started with Solr, and an understanding of what solutions are available to simplify adding great search to your site!

Solr by Eric Pugh

So you're building a website and want a terrific search experience for your users. Do what your database can't: faceted navigation, result highlighting, fuzzy queries, ranked scoring, spell correction, and more. Solr, the open-source enterprise search server is the answer. Solr bridges the technology divide between databases and document/web search engines (e.g. Google). Each has its uses but do not overlap. Chances are you have some structured data, probably in a database, and perhaps some related text documents. When you bring this data into Solr, you'll be able to deliver amazing features. Users will be able to navigate search results by filtering on aggregated attributes (so-called "faceted search"). Furthermore, various features like spell-correcting, auto-completing of search text, boosting records based on various rules, become possible. Solr does not tie you to a particular programming language or computing platform. And whether you have a thousand records or millions and many requests per second, Solr can scale to meet your performance needs. Furthermore, as an open-source solution, Solr doesn't ask you for more money when you want more out of it. You'll learn the basics of getting started with Solr, and an understanding of what solutions are available to simplify adding great search to your site!

Test Driven Development in Java: Live and Uncensored by Ben Rady

One of the barriers to wider adoption of TDD is that it is best taught from within a team, and the technical challenges of writing tests frequently thwart those looking to teach themselves.

Continuous Testing on the JVM by Ben Rady

Continuous Testing (CT) is a developer practice that shortens the feedback loop established by Test Driven Development. It gives you near instant feedback about the correctness of your code, and helps you find bugs as quickly as syntax errors.

Iteration-less, Value-Based Planning by Ben Rady

One of the hallmarks of a healthy Agile team is its ability to introspect and adapt. Many mature Agile teams have started to experiment with iteration-less development, in which releases occur as each new feature or story is completed. This practice allows for more frequent releases and smaller batch sizes, but can create problems when teams are forced to re-evaluate how they plan. In addition, the technical practices necessary to support this style of development are difficult to master, and some teams find themselves releasing more often at the expense of quality. |

ATDD/BDD with Cucumber 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.

ATDD/BDD with Cucumber 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.

Common AntiPatterns and How To Avoid Them by Mark Richards

In the book "97 Things Every Software Architect Should Know" (O'Reilly, 2009) I wrote about the importance of design patterns as a useful means of communication between architects and developers. Equally important to patterns is an understanding of AntiPatterns - things that we repeatably do that produce negative results. AntiPatterns are used by developers, architects, and managers every day and are one of the main factors that prevent progress and success. In this session we will look at some of the more common and significant development and architecture antipatterns. Through coding and design examples, you will see how these antipatterns emerge, how to recognize when the antipattern is being used, and most importantly, how to avoid them. By attending this session, you will be part of a movement to reduce the AntiPattern catalog from hundreds of entries to only a few. **Prerequisite:** None

The Art of Messaging by Mark Richards

Messaging is both a science and an art. Messaging is a science with respect to the mechanics of the JMS API and the syntax for sending and receiving messages. However, messaging is also an art when it comes to applying the JMS API to solve real-world problems. In this session I will review some of the more common use cases for messaging and show techniques for significantly increasing both the performance and scalability of messaging-based applications. Using ActiveMQ, you will see how to create embedded brokers, solve internal application bottleneck issues, how to use asynchronous logging with Log4J and JMS, and how to significantly speed up your messaging applications. In this session I will also describe and demonstrate some emerging trends in RESTful JMS (that is, JMS over HTTP). Come to this session to find out how much fun messaging can really be! **Prerequisite:** *Some knowledge of JMS and Messaging in general*

Using Apache Camel by Mark Richards

Apache Camel is a robust open source integration framework that handles routing and mediation tasks associated with enterprise integration. Camel allows you to quickly and easily route messages and integrate components in a distributed, decoupled manner. For example, using the Camel Java DSL, you can send and receive JMS messages in just a couple of lines of Java code. In this live coding session I will describe what Camel is, describe the overall architecture, show why it is useful, and demonstrate through live coding examples how to use the Camel Java DSL to write simple (and complex) routing logic. By attending this session you will learn Camel well enough to use it at work the next day.

Hydras and Hypermedia by Ian Robinson

Do you know what your enterprise apps get up to in their time off? Fighting fantasy, pick-your-path, hypermedia-driven, RESTful Web application adventures – of course. With techniques drawn from the forthcoming O'Reilly book REST in Practice, this session challenges the notion that REST is suitable only for simple CRUD-based data services, suggesting instead that the Web's architecture provides everything we need to model and implement sophisticated business processes in Web-based applications. **Prerequisite:** *This talk is suitable for distributed systems developers and architects, and anyone curious about REST's place in the enterprise. It assumes a familiarity with HTTP and Web application/service development.*

The Counterintuitive Web by Ian Robinson

The Web doesn't care for your finely-honed application architecture principles - for your orthodox tell-don't-ask, information hiding dictums, separated concerns, and guaranteed and reliable delivery strategies. It's an irresponsible place, where exposing your data, polling for results and making your errors the client's problem are considered acceptable behaviour. If it wasn't so successful, it'd be dismissed as an architectural clown. **Prerequisite:** *This talk is suitable for distributed systems developers and architects, and anyone curious about the Web's place in the enterprise.*

Business Architecture Foundations of IT by Ian Robinson

Many organisations today are frustrated by success: their rapid, ad hoc growth has resulted in a "tube map" systems estate that is costly to operate, and which inhibits business agility. In this talk, I show how an organisation can take its first steps to resolve this impasse by teasing apart strategy and execution. I discuss how our models of what the business does and what it's trying to achieve influence the progress of large software initiatives, and describe a number of practices and activities that together create a useful representation of a firm's operating model - one that can be used to identify, prioritise and plan strategic IT investments, and guide the evolution of the systems estate in an incremental, sustainable fashion.

Successful Software Management: 17 Lessons Learned by Johanna Rothman

Many software managers came to management through the technical ranks. Although they may have had plenty of technical training and mentoring, they frequently have to learn management skills the hard way, through trial and error. Johanna will describe some technical management tips and tricks learned through trial and error, focusing on software managers and their particular issues. You'll learn about a manager's job, how to create an effective work environment, and how you can help people do their best work.

Manage Your Project Portfolio: A Lean and Agile Approach by Johanna Rothman

Feel as if you can't get anything done? More projects than time to do them? Crises out the wazoo? Then it's time to consider another approach: managing your project portfolio. You can organize your projects and evaluate them without getting buried under a mountain of statistics.

Agile Program Management: Another Approach to Large Projects by Johanna Rothman

Have you ever waited weeks for one piece of functionality so you could release a large project? Have you been in the situation where the software is waiting for the hardware? Or, where the database admin held up the entire release because his work wasn't coordinated with the feature-based teams? Program management is the art of coordinating several sub-projects to a common objective. Until the parts are assembled into the whole, the parts have no value to the organization.

Coaching as a Two-Way Relationship by Johanna Rothman

Many of us have experienced sports coaches, where they helped us stretch for the crawl, turn in dance, or catch a ball. But sports coaching tends to be just one kind of coaching—and not necessarily the kinds of coaching you need to provide or hear at work. Coaching is one of the most important--and most difficult--responsibilities of leaders. Too often, people struggle and fail when it comes to coaching others. Coaches may try to impose their style on someone else for whom it doesn't fit. Some coaches try to transfer "best practices" regardless of how well they fit the organization. Other coaches talk too much. Coaching is a two-way relationship between a coach and the coachee. But not all the learning is on the coachee's side. If a coach is not learning as he or she proceeds with the coaching, the coach shortchanges the coachee.

Coaching as a Two-Way Relationship by Johanna Rothman

Many of us have experienced sports coaches, where they helped us stretch for the crawl, turn in dance, or catch a ball. But sports coaching tends to be just one kind of coaching—and not necessarily the kinds of coaching you need to provide or hear at work. Coaching is one of the most important--and most difficult--responsibilities of leaders. Too often, people struggle and fail when it comes to coaching others. Coaches may try to impose their style on someone else for whom it doesn't fit. Some coaches try to transfer "best practices" regardless of how well they fit the organization. Other coaches talk too much. Coaching is a two-way relationship between a coach and the coachee. But not all the learning is on the coachee's side. If a coach is not learning as he or she proceeds with the coaching, the coach shortchanges the coachee.

End-to-End Performance Optimization by Aleksandar Seovic

Performance matters. If your web site or application is unresponsive, users will either go to competitor's site or hate the application they are forced to use.

Performance and Scalability Revisited: In-Memory Data Grids by Aleksandar Seovic

Building scalable, highly-available applications that perform well is not an easy task. These features cannot be simply "bolted" onto an existing application – they have to be architected into it. Unfortunately, the things we need to do to achieve them are often in conflict with each other, and finding the right balance is crucial.

Agile Velocity by Ken Sipe

The agile development process is all about early and often feedback. One aspect of feedback is how is the team doing... Are we accurate in our estimates? Are we consistent in our velocity? As velocity varies, what is it telling me?

Architecture: Non-Functional Requirements by Ken Sipe

The agile focus of software development puts heavy focus on user requirements through user stories. However we can not lose sight of the non-functional requirements as well. The software could be written to the exact specification and desire of the user, however if it takes 5 minutes for a request response, or it only supports 2 users or it isn't secure, then we still haven't done our jobs as developers.

So you want to be an Architect by Ken Sipe

This session is a quick look at all aspects of being a corporate software architect. Whether you are a developer looking to move into the role of architect, needing to have an understanding of what is expected or already in the role of software architect looking for new and interesting ideas, this session is for you.

Hacking - The Dark Arts by Ken Sipe

A live Hacking demonstration exposing the tools and techniques used by Hackers.

Security Boundaries by Ken Sipe

Security is a large concern in today's world of software development. Security is a multi-dimensional problem requiring skills at a number of different levels. This session is a security overview of a typical Java web development stack.

Security Code Review by Ken Sipe

Security concerns abound... According to Gartner 75% of all attacks are at the web application tier. There has never been a more urgent time to understand the security concerns and how to apply solutions to our web applications.

Enterprise Security API library from OWASP by Ken Sipe

When it comes to cross cutting software concerns, we expect to have or build a common framework or utility to solve this problem. This concept is represented well in the Java world with the loj4j framework, which abstracts the concern of logging, where it logs and the management of logging. The one cross cutting software concern which seems for most applications to be piecemeal is that of security. Security concerns include certification generation, SSL, protection from SQL Injection, protection from XSS, user authorization and authentication. Each of these separate concerns tend to have their own standards and libraries and leaves it as an exercise for the development team to cobble together a solution which includes multiple needs.... until now... Enterprise Security API library from OWASP.

RDFA : Weaving Richness and Meaning in the Web by Brian Sletten

The human web is reasonably well in hand by now. We are getting pretty good at building systems that people find valuable and entertaining. We have not spent as much time concerned about our software friends. There is a ton of rich content available on the web that is too difficult to extract in automated ways using just XHTML, the meta tag and microformats. This talk will introduce you to some emerging technologies from the Semantic Web camp to enrich your web pages with useful information for both automated extraction and improved browsing experiences.

NetKernel: Making IT Matter Again by Brian Sletten

The premise of Nicholas Carr's "Does IT Matter?" book was that if everyone uses the same tools, processes, products, etc., is there any competitive advantage to be had from the average IT organization? NetKernel represents a fundamentally different approach to building systems. It takes what we like about Unix, REST and SOA and mixes it together. It inexplicably changes everything while allowing you to reuse existing code, services and libraries. Not only can it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. **Prerequisite:** *Introduction to NetKernel : Software for the 21st Century will be helpful but not strictly required for the workshop.*

NetKernel: Making IT Matter Again by Brian Sletten

The premise of Nicholas Carr's "Does IT Matter?" book was that if everyone uses the same tools, processes, products, etc., is there any competitive advantage to be had from the average IT organization? NetKernel represents a fundamentally different approach to building systems. It takes what we like about Unix, REST and SOA and mixes it together. It inexplicably changes everything while allowing you to reuse existing code, services and libraries. Not only can it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. **Prerequisite:** *Introduction to NetKernel : Software for the 21st Century will be helpful but not strictly required for the workshop.*

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. **Prerequisite:** *Semantic Web : The Future Now would be a useful introduction but is not required*

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. **Prerequisite:** *Semantic Web : The Future Now would be a useful introduction but is not required*

Automated deployment with Maven and friends - going the whole nine yards by John Smart

Automating your build process with Continuous Integration is certainly a great idea, but why stop there? Why not go the whole nine yards and automate the deployment process as well? Staging and production deployments are typically more complicated and more involved than a simple development deployment, but doing them by hand can be time-consuming, tricky and error-prone.

Zen and the Art of Build Script Maintenance by John Smart

Build scripts are an essential art in any software project. And yet they are so often fragile, brittle and unportable things, hard to understand and harder to maintain. In this talk, we cover what constitutes a good build script, and look at a few of the essential rules in writing one.

Coding Dojo with John Smart by John Smart

A Coding Dojo is a place where programmers come to improve their skills, by following a pattern similar to the martial arts dojo. Participants meet for at a pre-arranged time in a room that has one computer attached to a screen. The aim of the exercise will be to add features to an existing (working) web application, using a variety of technologies, such as Hibernate, JUnit 4, JWebUnit, Selenium, easyb and more. Participants will take turns to code, using pair programming and TDD.

Continuous Integration - Hudson by John Smart

Continuous Integration is a fundamental best practice of modern software development. In this workshop, you will learn how to set up an effective Continuous Integration environment using Hudson, a popular open source Continuous Integration tool.

Continuous Integration - Hudson by John Smart

Continuous Integration is a fundamental best practice of modern software development. In this workshop, you will learn how to set up an effective Continuous Integration environment using Hudson, a popular open source Continuous Integration tool.

Transforming to Groovy by Venkat Subramaniam

Groovy is a elegant, dynamic, agile, OO language. I like to program in Groovy because it is fun and the code is concise and highly expressive. Writing code in a language is hardly about using its syntax, however. It is about using the right idioms. Come to this section to pick up some nice Groovy idioms. **Prerequisite:** *Some knowledge of Groovy is helpful but not required.*

Scala for Java Programmers by Venkat Subramaniam

Scala is a very powerful, statically typed, hybrid functional, pure OO language. The strengths of Scala is in its expressiveness, support for XML, powerful pattern matching, and elegant solution to concurrency. In this workshop, you will deep dive into understanding the strengths of Scala from a very practical point of view.

Scala for Java Programmers by Venkat Subramaniam

Scala is a very powerful, statically typed, hybrid functional, pure OO language. The strengths of Scala is in its expressiveness, support for XML, powerful pattern matching, and elegant solution to concurrency. In this workshop, you will deep dive into understanding the strengths of Scala from a very practical point of view.

How to Approach Refactoring by Venkat Subramaniam

You can't be agile if your code sucks. You know that you have to constantly refactor your code and design. But the questions is how? In this presentation, instead of looking at a laundry list of refactoring techniques,

we will instead look at how to effectively approach refactoring and along the way discuss some core principles to look for.

TDD Regular Code/Multithreaded Code! by Venkat Subramaniam

There are two reasons to do TDD. One is to ensure the code meets and continues to meet the expectations at the unit of code level. The second, and equally significant, reason is to drive the design of the code. Realizing the first benefit is rather mechanical and comes with rigor and discipline. Realizing the second benefit, however, requires quite some hard work and rethinking. It takes a bit of *unlearning* to achieve this goal.

TDD Regular Code/Multithreaded Code! by Venkat Subramaniam

There are two reasons to do TDD. One is to ensure the code meets and continues to meet the expectations at the unit of code level. The second, and equally significant, reason is to drive the design of the code. Realizing the first benefit is rather mechanical and comes with rigor and discipline. Realizing the second benefit, however, requires quite some hard work and rethinking. It takes a bit of *unlearning* to achieve this goal.

Developing a Complex External DSL by Vaughn Vernon

The use of a domain-specific language, or DSL, is becoming a realistic and even necessary solution for software developers on all sorts of projects. You've heard about DSLs, and you may know that DSLs are divided into a few different styles, internal and external. But what is an internal DSL and external DSL? When would you decide to use one or the other? And, primarily, how would you go about developing a complex external DSL? This presentation answers these questions, with a focus on developing a complex external DSL.

Implementing Domain-Driven Designs by Vaughn Vernon

Eric Evans' masterful book, Domain-Driven Design, is deep in theory and pattern language, but light on implementation guidance. With so much to offer, DDD should be put to use by many more teams seeking to tackle complexity in the heart of their software systems. However, without clear examples of how behavior-rich domain model implementations are achieved, many who could benefit greatly, instead shy away. Here's your chance to grasp DDD as Vaughn shares his experience through three practical example domain models implemented in Java.

The Domain-Driven Design Metamodel by Vaughn Vernon

Is there a metamodel behind domain-driven design? Is it possible to blend a complex external DSL and the concepts of domain-driven design patterns to produce a tool to rapidly implement advanced domain models that strictly adhere to Eric Evans' DDD pattern language? Absolutely, and Vaughn demonstrates how this has been achieved and why the repeatable method is important to you.