

Pacific Northwest Software Symposium

Redmond Marriott Town Center

September 18 - 20, 2009

<http://www.nofluffjuststuff.com/conference/seattle/2009/09/home>

Fri, Sep. 18, 2009

	Salon 3	Salon 4	Bear River	Lewis River	Elk River
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	On Being a Software Architect Mark Richards	The Amazing Groovy Weight-loss Plan Scott Davis	Iteration Management: What's in your Toolkit? Tiffany Lentz	Effective Java Venkat Subramaniam	JSF 2.0: An Introduction David Geary
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Common AntiPatterns and How To Avoid Them Mark Richards	Groovy XML Ninja Skills Scott Davis	Metrics Madness! Tiffany Lentz	Cleaning up Code Smell Venkat Subramaniam	JSF 2.0: Advanced Topics David Geary
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Transaction Pitfalls and Strategies Mark Richards	Groovy Testing Scott Davis	Agile Project and Management Metrics: Measuring Success Downward and Upward Tiffany Lentz	Programming Scala Venkat Subramaniam	Flex for Java Developers David Geary
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: Venkat Subramaniam				

Sat, Sep. 19, 2009

	Salon 3	Salon 4	Bear River	Lewis River	Elk River
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Architecture and Scaling Ken Sipe	Building External DSLs Venkat Subramaniam	Dim Sum Grails: A Sampler of Practical Non Database- Driven Grails Applications Scott Davis	Introduction to JMS Mark Richards	The Agile Mindset: Applying Agile in Non-Technical Areas of an Organization Tiffany Lentz
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	What's New in Spring 3 Ken Sipe	Design Patterns in Java and Groovy Venkat Subramaniam	Web 2.0 Checklist: Deconstructing Modern Websites Scott Davis	Advanced Topics in JMS Mark Richards	Project Bootstrapping Nathaniel Schutta
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Security Boundaries Ken Sipe	MOPping Up Groovy Venkat Subramaniam	GWT: An Introduction David Geary	This is not your father's Von Neumann Machine: A crash course in modern CPU architecture Brian Goetz	Open Source Debugging Tools for Java Matthew McCullough
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	The Busy Java Developer's Guide to Java7 Ted Neward	Mastering Maven 2.0 Matthew McCullough	GWT: Advanced Topics David Geary	Stupid JIT Tricks Brian Goetz	Seven Habits of Highly Dysfunctional Teams Nathaniel Schutta
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSION				

Sun, Sep. 20, 2009

	Salon 3	Salon 4	Bear River	Lewis River	Elk River
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	REST : Information-Driven Architectures for the 21st Century Brian Sletten	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	Brew up a rich web application with Cappuccino Howard Lewis Ship	The Java Memory Model Brian Goetz	Iteration 0 Ken Sipe
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	SPARQL: Querying the Data Web Brian Sletten	Making Web Apps Suck Less Nathaniel Schutta	Git Going with Distributed Version Control Matthew McCullough	Are All Web Applications Broken? Brian Goetz	7 Habits of Highly Effective Developers Ken Sipe
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa Brian Sletten	Cloud Computing Boot Camp on the Google App Engine Matthew McCullough	Clojure: Functional Concurrency for the JVM Howard Lewis Ship	Garbage-collector-friendly programming Brian Goetz	The Busy Java Developer's Guide to Collections Ted Neward
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Semantic SOA : Meaningful Service Strategies Brian Sletten	iPhone Objective-C with Java Web Services Matthew McCullough	Clojure Deep Dive Howard Lewis Ship	Hacking Your Brain for Fun and Profit Nathaniel Schutta	The Busy Java Developer's Guide to Advanced Collections Ted Neward

Pacific Northwest Software Symposium

Redmond Marriott Town Center

September 18 - 20, 2009

<http://www.nofluffjuststuff.com/conference/seattle/2009/09/home>

The Amazing Groovy Weight-loss Plan by Scott Davis

"The central enemy of reliability is complexity." (Dr. Daniel Geer) Java is a powerful programming language. A smart developer can do nearly anything with Java. So the next question is, "How quickly can it be done? How many lines of code does it take to do common tasks?" Groovy greases the wheels of Java by decreasing the complexity of the language while preserving the raw power. At first glance, you might think that this talk is simply about how Groovy drastically reduces the lines of code you need to write. What this talk is really about is bringing simplicity, clarity, readability, and yes, beauty to your source code.

Groovy XML Ninja Skills by Scott Davis

"XML is like violence: if it doesn't solve your problem, you aren't using enough of it." (Anonymous) XML is everywhere. Whether you are dealing with local configuration files (web.xml, struts-config.xml) or remote web services (SOAP, REST, RSS, Atom), the modern software developer needs to be able to request, slice, and dice XML with ease. That requires a set of razor-sharp tools that reduce the inherent complexity of the problem, not multiply it. Once you see XML tremble in fear at the awesome power of Groovy, you'll wonder what you ever did without it.

Groovy Testing by Scott Davis

"Tests don't break things; they dispel the illusion that it works." (Anonymous) In this era of "Test-First" and "Test-Driven" development, the modern software engineer knows that testing is no longer an optional part of the process. You need to have the best tools at your fingertips: a set of utilities that maximize your results with a minimum of effort. Groovy offers Java developers an optimal set of testing tools.

Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications by Scott Davis

"The proof of the pudding is in the eating. By a small sample we may judge of the whole piece." (Miguel de Cervantes Saavedra) Most Grails tutorials demonstrate how easy it is to build simple CRUD (Create/Retrieve/Update/Delete) applications. While skinning a database with a web front-end is undeniably one beneficial aspect of Grails, it isn't the only thing Grails is good for. As you'll see here, Grails can be used to build a wide variety of web applications. You won't see a single HTML table with "edit" and "delete" links, I promise.

Web 2.0 Checklist: Deconstructing Modern Websites by Scott Davis

"The challenge of modernity is to live without illusions and without becoming disillusioned." (Antonio Gramsci) There are plenty of sarcastic "Web 2.0" checklists out there -- be perpetually in BETA, when in doubt add rounded corners, etc. While we can all laugh at the superficial aspects of the Web 2.0 revolution, there are plenty of serious aspects to it as well. Is your website mash-up friendly or hostile? Do you tell your visitors when things change (via RSS or Atom syndication), or do you expect them to check in daily for updates? Is your website a silo or a part of a larger ecosystem?

JSF 2.0: An Introduction by David Geary

This session introduces JSF 2.0 fundamentals, with emphasis on new features in JSF 2.0. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks*

JSF 2.0: Advanced Topics by David Geary

This session covers two of the most important features of JSF 2.0: composite components and built-in Ajax. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks. Familiarity with Ajax. This session builds on the JSF 2.0 Introduction talk, so it is helpful, although not required, if you attend the intro talk before coming to this session.*

Flex for Java Developers by David Geary

An introduction to Flex for Java developers. **Prerequisite:** *Familiarity with Flex and at least one other web application framework*

GWT: An Introduction by David Geary

An introduction to Google Web Toolkit. **Prerequisite:** *Familiarity with a component-based framework, preferably a desktop application framework*

GWT: Advanced Topics by David Geary

Learn to do really cool stuff with GWT. **Prerequisite:** *The GWT: Introduction session is not a prerequisite for this session, but it will help if you have some familiarity with GWT.*

This is not your father's Von Neumann Machine: A crash course in modern CPU architecture by Brian Goetz

Do software developers need to know anything about CPU architecture? They do if they aspire to be performance experts. Modern CPUs behave almost nothing like the sequential Von Neumann machine model we know and love. This session provides an overview of the architecture of modern CPUs, how this has changed in recent years, and what the implications are for software development and performance management.

Stupid JIT Tricks by Brian Goetz

Ever wondered what happens to your bytecodes when they're executed by a Java Virtual Machine? This talk provides a peek "under the hood" of modern JVMs, exploring dynamic compilation, speculative optimization, garbage collection, and some hardware-specific optimizations.

The Java Memory Model by Brian Goetz

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

Are All Web Applications Broken? by Brian Goetz

Many developers believe that web frameworks "take care of" the details of concurrency, but this is only because most web applications make limited use of state. Stateful web applications also need to be careful about hazards like races. This talk will use the Java Memory Model to analyze common patterns of state management in web applications. **Prerequisite:** *The Java Memory Model*

Garbage-collector-friendly programming by Brian Goetz

To many developers, garbage collection is black magic. Accordingly, there are is a lot of conflicting advice about what is good or bad for the garbage collector. In this talk, I look at how garbage collection is implemented in the HotSpot VM, and techniques for writing programs that exhibit good garbage collection behavior. Surprisingly, many of these techniques coincide with writing good, clean code.

Iteration Management: What's in your Toolkit? by Tiffany Lentz

Is your Agile team running smoothly? How do you know? This answer is found in your iteration and your toolkit for constant team improvement! Comparing Iteration Management skills and tools to those of the Agile Project Manager, Scrum Master and Technical Leader roles, you will see that Iteration Management encompasses end to end activities of the iteration, which are crucial to unblocking your software production line and making your team a success.

Metrics Madness! by Tiffany Lentz

This interactive session allows the audience to drive the discussion by putting forward current team struggles for the group and the facilitator to address using Agile metrics. Good metrics properly applied are the key to visibility and highly functioning team, complete with problem ownership and business-driven solutions.

Agile Project and Management Metrics: Measuring Success Downward and Upward by Tiffany Lentz

This session focuses on visibility of team progress and correcting team "bad smells" using Agile metrics. Since we use metrics to self-correct and sharpen the team, they are an integral part of each iteration and flow into release planning. We will discuss some of the How's and Why's of team metrics and review the risks that are inevitable when it comes to gathering metrics.

The Agile Mindset: Applying Agile in Non-Technical Areas of an Organization by Tiffany Lentz

Agile techniques are often pigeon-holed as just applying to software projects and IT organizations. Agile techniques are a mindset more than a list of rules to follow and can bring efficiency and improvements to all areas of an organization.

Brew up a rich web application with Cappuccino by Howard Lewis Ship

Cappuccino is a rich web framework that all but redefines what's possible in the web browser. It brings the heritage and richness of the Apple Objective-C language and runtime directly into the browser, as Objective-J (Objective JavaScript). Cappuccino apps look and act like Desktop Mac OS X Applications -- a very high bar to set! In many ways comparable to Google Web Toolkit, Cappuccino is simpler and less obtrusive.

Clojure: Functional Concurrency for the JVM by Howard Lewis Ship

Talk about strange bedfellows: what happens when you mix one part Lisp (one of the oldest computer languages), one part Java (so young, yet so well adopted), a healthy serving of functional programming, and add a state-of-the-art concurrency layer on top? That's Clojure, which "feels like a general-purpose language beamed back from the near future."

Clojure Deep Dive by Howard Lewis Ship

Clojure packs a tremendous amount of power into a very small package. Clojure inherits the power and elegance of Lisp, and in this session we'll explore the features that make Clojure so attractive: macros, multimethods, meta-data, and an overview of Clojure's rich built-in library. **Prerequisite:** *Clojure: Functional Concurrency for the JVM.*

Open Source Debugging Tools for Java by Matthew McCullough

This session will survey a wide range of tools across the Java space. We'll look at utilities such as VisualVM, jstatd, jps, jhat, jmap, Eclipse Memory Analyzer, jtracert, btrace and more. Open Source is not just a suite of libraries you consume within your application, but now reaches into the space of tools to help you troubleshoot and improve your applications. The price of these tools eliminates barriers to their use and their open source nature allows you to mix and match them into compositions that work well for your application's unique debugging needs.

Mastering Maven 2.0 by Matthew McCullough

Maven has been on the Java build tools scene for quite a number of years, but the adoption rate in enterprises is now going through the roof. Maven can seem daunting, but this presentation will equip existing Maven users with more efficient techniques and tools to overcome the biggest perceived Maven hurdles and build issues with ease. We'll examine tools to help you find artifacts in central repositories, manage your corporation's internal Maven artifacts with a proxy tool such as Nexus, view and override dependency graphs, dependency management and multi-module best practices, create OS specific profiles, and leverage the latest Maven plugins for the top Java IDEs. **Prerequisite:** *Basic Maven knowledge*

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*

Cloud Computing Boot Camp on the Google App Engine by Matthew McCullough

Cloud this, cloud that. It's all we are hearing about these days. And whether buzz-worthy or not, you need to get in-the-know so that you can talk effectively about how this could fit into the application strategy on your next project. Leverage 100s of hours of research distilled into a 90 minute presentation. Get bootstrapped with what cloud computing is and isn't, who the players are in this space, what unique features each offers, and then how Google is completely changing the game.

iPhone Objective-C with Java Web Services by Matthew McCullough

iPhone development is all the rage both in the mobile entertainment, social networking, and productivity application spaces. As a Java developer, prepare yourself to be a participant in aspects of this new breed

and platform of development. Hop on board with a quick start to iPhone application coding in Objective-C and integration with some of our favorite Java web service back-ends such as RESTful Grails.

The Busy Java Developer's Guide to Java7 by Ted Neward

Even though the Java 7 JSR has yet to be formed, some interesting things are beginning to emerge from Sun about what Java7 may include when its formal release contents are finally made public.

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 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*

On Being a Software Architect by Mark Richards

One way to stop a conversation dead while at a party or gathering is to mention you are a software architect. Why? Because it takes about an hour (complete with Powerpoint slides) to explain what you do for a living. By then the person you are talking to is so bored they would rather sit in a corner licking nine-volt batteries. The problem is that no one inside or outside our industry really knows what a software architect is or what they do. In this highly interactive (and slightly humorous) session we will take a deep dive into the role a software architect plays in the IT industry. We will explore the characteristics an architect needs to have, and the elements that make a good architect and a bad architect. Through amusing antidotes and real-world examples, we will see how to become an effective software architect and help shape the industry in terms of the role and title of software architect. **Prerequisite:** *None*

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*

Transaction Pitfalls and Strategies by Mark Richards

In previous years I have given sessions related to my book "Java Transaction Design Strategies", where I have reviewed the basics of programmatic and declarative transactions and outlined the basic patterns described in the book. In this new session for 2009 I will focus on some of the pitfalls encountered while dealing with transactions and then how to develop an effective transaction strategy. I will start this session by describing and illustrating some of the common pitfalls I continue to see in both Spring and EJB. I will then describe four common transaction strategies you can use and implement, including a transaction strategy for high-speed transactions, a transaction strategy for client orchestration, a transaction strategy for use with API's, and finally a strategy for highly concurrent environments. Note: This session assumes you know a little bit about transactions and have been using them in either Spring or EJB. It is not intended to be an introductory session on how transactions work. You can obtain a free PDF download of my transaction book at <http://www.infoq.com/minibooks/JTDS> to quickly come up to speed with transactions. **Prerequisite:** *Java, Spring or EJB; some knowledge of transactions and JTA.*

Introduction to JMS by Mark Richards

There's no doubt about it - messaging is quickly becoming a standard part of most application architectures, particularly as more and more companies struggle to find ways to integrate heterogeneous environments due to mergers, acquisitions, or to streamline existing application portfolios. The Java Message Service (JMS) API allows Java applications to implement messaging using a standard API, therefore removing the

dependency of any particular messaging provider. In this introductory session we will take a look at the basics of messaging and the JMS API. I will start by discussing the different messaging models, the structure of a basic JMS message, and the JMS API interfaces and how they interrelate. Then through interactive coding I will show the basics of sending and receiving messages using the point-to-point messaging model and how to do request/reply processing. NOTE: this session is meant to be an introduction to messaging and JMS - no prior JMS or messaging experience is needed for this session. **Prerequisite:** None

Advanced Topics in JMS by Mark Richards

This session covers some of the more advanced features of JMS messaging, and is intended for those who are familiar with JMS and messaging in general. Some of the topics I will be covering in this session include message grouping (where I will demonstrate sending a large JPG image using messaging), transacted sessions, client-based acknowledgement, and some various messaging design considerations and things to watch out for from a design and coding perspective. I will be doing live coding demonstrations to illustrate the techniques described in this session. Although this session is entirely JMS provider agnostic, I will be using ActiveMQ, a popular open source JMS provider, during the live coding demonstrations. **Prerequisite:** Some knowledge of messaging and JMS would be helpful

Project Bootstrapping by Nathaniel Schutta

Ah, that new project smell, it's intoxicating! Full of hope, we trek off in pursuit of technical greatness. In this talk, we'll cover some of the important first steps of a new project including continuous integration, creating a testing culture and establishing low ceremony process.

Seven Habits of Highly Dysfunctional Teams by Nathaniel Schutta

Being on a high performing team is a transcendent experience - unfortunately, many of us find more dysfunction than function. In this talk, we'll take a look at some of the common issues that face teams and discuss some ways of working towards a happy crew.

JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language.

Making Web Apps Suck Less by Nathaniel Schutta

We've all used web applications that had us screaming at their creators - unfortunately sometimes we're the ones being cursed. Believe it or not, there are some simple steps we can take to ensure that our users have a great experience.

Hacking Your Brain for Fun and Profit by Nathaniel Schutta

The single most important tool in any developers toolbox isn't a fancy IDE or some spiffy new language - it's our brain. Despite ever faster processors with multiple cores and expanding amounts of RAM, we haven't yet created a computer to rival the ultra lightweight one we carry around in our skulls - in this session we'll learn how to make the most of it. We'll talk about why multitasking is a myth, the difference between the left and the right side of your brain, the importance of flow and why exercise is good for more than just your waist line.

Architecture and Scaling by Ken Sipe

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

What's New in Spring 3 by Ken Sipe

The Spring Framework has led the industry in innovation for years. Starting with dependency injection and promoting testing through removal of framework dependencies. Spring 3.0 continues that innovation in a way that takes full advantage of the Java 5 platform. There are a number of significant changes to the framework. So whether you are new to the framework or an experience Spring developer, this is a great session to come up to speed on the latest from SpringSource. **Prerequisite:** Java 5

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.

Iteration 0 by Ken Sipe

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

7 Habits of Highly Effective Developers by Ken Sipe

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

REST : Information-Driven Architectures for the 21st Century by Brian Sletten

There is a shift going on in the Enterprise. While still used and useful, the promises of the SOAP/WSDL/UDDI Service-Oriented Architecture (SOA) stack have failed to live up to their promise. A new vision of linked information is enveloping online and Enterprise users. The REST architectural style is squarely behind this thinking as a way of achieving low-cost, flexible integration, increased data security, greater scalability and long-term migration strategies. If you have dismissed REST as a toy or are unfamiliar with it, you owe it to yourself to see what is so interesting about this way of doing things.

SPARQL: Querying the Data Web by Brian Sletten

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There is an emerging Data Web that relies on technologies from the Semantic Web stack to link increasingly rich connections between various data sources. SPARQL and RDF are the main tools for expressing and using this connectivity. This talk will introduce you to one of the practical and accessible aspects of employing these ideas on the Web and in the Enterprise. **Prerequisite:** *The Semantic Web: The Future, Now and Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa would both be helpful but are not required*

Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa 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 a 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. **Prerequisite:** *The Semantic Web: The Future Now would be helpful, but not required*

Semantic SOA : Meaningful Service Strategies by Brian Sletten

The goal for web services was always to reduce our burden by increasing the potential for reuse of business functionality. Somehow, we got lost along the way in a morass of confusing, unfulfilling and downright broken technologies. While we are interested in pursuing REST-based systems for managing information, we need some strategies for tying it all together sensibly. If we abandon WSDL, SOAP and UDDI, what do we replace them with? This talk will walk you through combining resource-oriented strategies with technologies from the Semantic Web to describe, find, and bind to services in dynamic, flexible and extensible ways.

Prerequisite: *The Semantic Web: The Future Now, Give it a REST and SPARQL : Querying the Data Web would all be helpful talks to have attended*

Effective Java by Venkat Subramaniam

Java is a well established language, that has been around for more than a decade. Yet, programming on it has its challenges. There are concepts and features that are tricky. When you run into those, the compiler is not there to help you.

Cleaning up Code Smell by Venkat Subramaniam

Projects often start out simple, but soon become complex and turn into a lose cannon. Organizations are struggling to maintain and evolve software. Poor code quality is a significant part of that problem. Improving the quality of code is critical to success of enterprise projects.

Programming Scala by Venkat Subramaniam

Scala is a static fully object-oriented, functional language on the JVM. While taking advantage of the functional aspects, you can continue to make full use of the powerful JVM and Java libraries.

Programming Scala by Venkat Subramaniam

Scala is a static fully object-oriented, functional language on the JVM. While taking advantage of the functional aspects, you can continue to make full use of the powerful JVM and Java libraries.

Building External DSLs by Venkat Subramaniam

Domain Specific Languages (DSLs) are languages targeted at a particular problem and domain. They have context and are fluent. They help users of applications at various levels to easily communicate with your application. Developing DSLs, however, are not easy. You could easily get dragged into using parsers and tools with steep learning curve.

Design Patterns in Java and Groovy by Venkat Subramaniam

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

MOPping Up Groovy by Venkat Subramaniam

Groovy's dynamic capabilities makes it an attractive language for Meta Programming. There are several facilities to intercept method calls to do AOP kind of operations. You can also inject methods dynamically. You can also easily perform method synthesis as well. In this presentation we will take a look at techniques that make Groovy pretty Hip for MOP.