

Salt Lake Software Symposium

Marriott Salt Lake City University Park

July 17 - 18, 2009

http://www.nofluffjuststuff.com/conference/salt_lake_city/2009/07/home

Fri, Jul. 17, 2009

	Bonneville 1	Bonneville 2 & 3	Theatre	Connor	Fort Douglas
7:30 - 8:30 AM	REGISTRATION/BREAKFAST				
8:30 - 8:45 AM	WELCOME				
8:45 - 10:15 AM	Emergent Design & Evolutionary Architecture Neal Ford	Designing for Ajax, part 1 Nathaniel Schutta	Effective Java Venkat Subramaniam	The Amazing Groovy Weight-loss Plan Scott Davis	Agile, Relevance Style Stuart Halloway
10:15 - 10:45 AM	BREAK				
10:45 - 12:15 PM	Real-world Refactoring Neal Ford	Designing for Ajax, part 2 Nathaniel Schutta	Programming Scala Venkat Subramaniam	Groovy XML Ninja Skills Scott Davis	Taking Agile From Tactics to Strategy Stuart Halloway
12:15 - 1:15 PM	LUNCH				
1:15 - 2:45 PM	Git control of your source Stuart Halloway	Cleaning up Code Smell Venkat Subramaniam	The Busy Java Developer's Guide to Java7 Ted Neward	Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications Scott Davis	Project Bootstrapping Nathaniel Schutta
2:45 - 3:00 PM	BREAK				
3:00 - 4:30 PM	Test Driven Design Neal Ford	Building External DSLs Venkat Subramaniam	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Lizard Brain Web Design Scott Davis	Seven Habits of Highly Dysfunctional Teams Nathaniel Schutta
4:30 - 4:45 PM	BREAK				
4:45 - 6:15 PM	How to Fail with 100% Code Coverage Stuart Halloway	Design Patterns in Java and Groovy Venkat Subramaniam	The Busy Java Developer's Guide to Advanced Platform Security Ted Neward	Web 2.0 Checklist: Deconstructing Modern Websites Scott Davis	Visualizations for Code Metrics Neal Ford
6:15 - 7:00 PM	DINNER				
7:00 - 8:00 PM	Keynote: Venkat Subramaniam				

Sat, Jul. 18, 2009

	Bonneville 1	Bonneville 2 & 3	Theatre	Connor	Fort Douglas
7:30 - 8:15 AM	BREAKFAST				
8:15 - 9:45 AM	Architecture and Scaling Ken Sipe	The Productive Programmer: Mechanics Neal Ford	Open Source Debugging Tools, A 90 Minute Bootcamp Matthew McCullough	JSF 2.0: An Introduction David Geary	Agile Retrospectives Stuart Halloway
9:45 - 10:00 AM	BREAK				
10:00 - 11:30 AM	So you want to be an Architect Ken Sipe	Hands-on Agile Development Neal Ford	The Busy Java Developer's Guide to ClassLoaders Ted Neward	JSF 2.0: Advanced Topics David Geary	Introduction to NetKernel : Software for the 21st Century Brian Sletten
11:30 - 12:15 PM	EXPERT PANEL DISCUSSION				
12:15 - 1:00 PM	LUNCH				
1:00 - 2:30 PM	Architecture: Non-Functional Requirements Ken Sipe	Unit Testing that Sucks Less: Small Things that Make a Big Difference Neal Ford	The Busy Java Developer's Guide to Collections Ted Neward	Flex for Java Developers David Geary	Advanced NetKernel : Software for the 21st Century Brian Sletten
2:30 - 2:45 PM	BREAK				
2:45 - 4:15 PM	What's New in Spring 3 Ken Sipe	Java.next #1: Common Ground Stuart Halloway	Mastering Maven 2.0 Matthew McCullough	GWT: An Introduction David Geary	SPARQL: Querying the Data Web Brian Sletten
4:15 - 4:30 PM	BREAK				
4:30 - 6:00 PM	Security Boundaries Ken Sipe	Programming Clojure Stuart Halloway	iPhone Objective-C integration to Java Web Services Matthew McCullough	GWT: Advanced Topics David Geary	Semantic SOA : Meaningful Service Strategies Brian Sletten

Sun, Jul. 19, 2009

Sun, Jul. 19, 2009					
--------------------	--	--	--	--	--

Salt Lake Software Symposium

Marriott Salt Lake City University Park

July 17 - 18, 2009

http://www.nofluffjuststuff.com/conference/salt_lake_city/2009/07/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.

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.

Lizard Brain Web Design by Scott Davis

"There's an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone." (Bjarne Stroustrup) The "lizard brain" is the oldest part of the human brain -- the part responsible for autonomic functions like breathing, heart rate, and navigating websites. OK, maybe not that last part, but your website should be easy to use. Stupid easy. Lizard brain easy. Any time your user spends figuring out how to do something -- even for a split second -- is wasted time due to poor design. Inspired by Steve Krug's book "Don't Make Me Think", this talk answers the question, "Why is that website so hard to use?"

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?

Emergent Design & Evolutionary Architecture by Neal Ford

Most of the software world has realized that BDUF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent design & evolutionary architecture, including both proactive (test-driven development) and reactive (refactoring, composed method) approaches to discovering design. The goal of this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed, keeping you code in sync with the problem domain.

Real-world Refactoring by Neal Ford

Refactoring is a fine academic exercise in the perfect world, but we don't really live there. Even with the best intentions, projects build up technical debt and cruffy bad things. This session covers refactoring in the real world, at both the atomic level (how to refactor towards composed method and the single level of abstraction

principle) to larger project strategies for multi-day refactoring efforts. This talk provides practical strategies for real projects to effectively refactor your code.

Test Driven Design by Neal Ford

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

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.

The Productive Programmer: Mechanics by Neal Ford

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

Hands-on Agile Development by Neal Ford

BRING YOUR LAPTOP WITH YOU, BUT A LAPTOP ISN'T REQUIRED! Reading and hearing about agile practices is one thing, but actually doing it is completely different. This session puts you to work in an agile fashion, applying agile developer practices.

Unit Testing that Sucks Less: Small Things that Make a Big Difference by Neal Ford

Unit testing seems to a lot of managers and developers like pure overhead, but professionally responsible developers know that it is one of the keys to quality. This session covers a bunch of small tools that makes testing easier & faster. I talk about tools like Infinitest, Jester, MockRunner, Hamcrest, Groovy, RSpec/EasyB, Selenium, and others. While none of these tools is elaborate enough for it's own session, together they add up to more than the sum of the parts. This session shows tools and strategies to streamline testing, making easier and more palatable for both managers and developers.

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 kickass 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.*

Agile, Relevance Style by Stuart Halloway

The Agile Manifesto, like any good scripture, admits of many interpretations. There is no one "right path." What works for us may not work for you. At Relevance we have tried many paths, and learned many lessons. Join us to see dozens of ideas that have worked for us, plus some that haven't.

Taking Agile From Tactics to Strategy by Stuart Halloway

Teams adopting agile should begin at a tactical level, but they shouldn't end there. The Agile Manifesto operates at many different levels. Learn to apply the principles of agile at a strategic level. Otherwise you can have a great agile ground game and still lose.

Git control of your source by Stuart Halloway

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

How to Fail with 100% Code Coverage by Stuart Halloway

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

Agile Retrospectives by Stuart Halloway

Agile teams manage change and risk by apapting. But to adapt, you must identify opportunities for change and take them. Retrospectives are a fun, cost-effective way for your team to learn and change.

Java.next #1: Common Ground by Stuart Halloway

In this talk, we will explore and compare four of the most interesting new JVM languages: Clojure, Groovy, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all of them succeed in this mission. However, these languages have very different design goals. We will explore these differences, and help you decide when and where these languages might fit into your development toolkit. For more information see <http://blog.thinkrelevance.com/2008/8/4/java-next-common-ground>.

Programming Clojure by Stuart Halloway

Find out why Clojure is Java.next: * Clojure provides clean, fast access to all Java libraries. * Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. * Clojure includes Lisp's signature feature: Treating code as data through macros. * Clojure's emphasis on immutability and support for software transactional memory make it a viable option for taking advantage of massively parallel hardware.

Open Source Debugging Tools, A 90 Minute Bootcamp by Matthew McCullough

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. This session will quickly survey a wide range of tools across the Java, Networking, Filesystem, SOAP, REST, HTML, CSS and JavaScript realms. We'll look at applications such as VisualVM, which help you analyze your heap and garbage collection cycles of both local and remote applications. Performance and load testing tools such as JMeter will expose bottlenecks, threading, and scalability concerns of everything from Java modules to Web Apps (even ones that don't use any Java). **Prerequisite:** *Basic knowledge of web services, core Java, web application design.*

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*

iPhone Objective-C integration to 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 JSR311 Jersey and 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 Java Platform Security by Ted Neward

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

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

So you know the platform security model, and now you want to use it in new and interesting ways, like creating a custom Policy implementation, a custom Permission, or create a custom security context in which code will execute. Perhaps you even wish to make certain objects accessible only to those with the right permissions, or cryptographic key. Nothing could be easier, despite Java security's reputation as a dark and arcane place. **Prerequisite:** *The Busy Java Developer's Guide to Platform Security*

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

If you've ever gotten a ClassCastException and just knew the runtime was wrong about it, or found yourself copying .jar files all over your production server just to get your code to run, then you probably find the Java ClassLoader mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--ClassLoaders aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose ClassLoader problems. And once you've got that, and hear about ClassLoaders' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like ClassLoaders after all... maybe.

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.

Designing for Ajax, part 1 by Nathaniel Schutta

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decide where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

Designing for Ajax, part 2 by Nathaniel Schutta

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

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.

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?

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.

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.

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.

Introduction to NetKernel : Software for the 21st Century by Brian Sletten

Imagine the simplicity of REST married to the power of Unix pipes with the benefits of a loosely-coupled, logically-layered architecture. If that is hard to imagine, it may be because the architectures available to you today are convoluted accretions of mismatched technologies, languages, abstractions and data models. NetKernel is a disruptive technology that changes the game. It has been quietly gaining mind share in the past several years; people who are exposed to it don't want to go back to the tired and blue conventions of J2EE and .NET. Not only does 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. Come see how this open source / commercial product can change the way you think about building software.

Advanced NetKernel : Software for the 21st Century by Brian Sletten

If you have come to the NetKernel overview talk and came away compelled but unsure how to proceed, this talk will jump right in to building real resource-oriented systems with NetKernel. We will move away from the theoretical mind-melting right into the applied mind-melting. It is difficult to make the shift away from an object-oriented model, but this talk will demonstrate several examples of how and why you may want to. It will also include a preview of what is coming in NetKernel 4. This is kind of a REST + Polyglot Programming + SOA + Architecture talk all rolled up into one. *Prerequisite: Intro to NetKernel : Software for the 21st Century, Give it a REST (if unfamiliar with REST)*

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*

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.

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.

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.

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.

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.