

# Northern Virginia Software Symposium

Sheraton Reston

November 6 - 8, 2009

<http://www.nofluffjuststuff.com/conference/reston/2009/11/home>

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Northern Virginia Software Symposium on November 6 - 8, 2009. NVSS2009FE will be held at the Sheraton Reston.

Since 2002, the No Fluff Just Stuff Java Symposium has been regarded as the premier Java/Agility event series anywhere serving over 21,000 attendees with some 130 events. The popularity of the NFJS symposium series can be traced to the following:

- 1). Exceptional Speakers
- 2). Limited Attendance - capped at 250 people
- 3). No Vendors, No Sales Pitches, No Marketecture
- 4). Excellent networking opportunity with speakers and fellow attendees because of small size.
- 5). The Best Value in the Java conferencing space period.

You will have the opportunity to attend multiple sessions on variety of interesting topics covering core Java, Enterprise Java, Web 2.0, Dynamic Languages, Architecture, Security, Testing, CI and Agility.

## **The Registration Fee Includes:**

3 Day All Access Pass to NVSS2009FE

All Meals/Snacks - duration of the symposium

Complimentary copy of "Career 2.0" by Jared Richardson

90 Day IntelliJ license compliments of JetBrains

Session Materials

Custom NFJS Binder

Great Giveaways @ NFJS including \$200 and \$400 Apple Gift cards

**Early Bird Registration:** \$850/person good thru 10/19/09 after \$950

**Excellent Group Discounts Available** - bring your entire development team to the show - no travel required!! Rate good thru 10/19/09

## **Registration Fees**

Attendees	Before Oct. 19, 2009	After Oct. 19, 2009
5-9	\$750	\$850
10-14	\$725	\$825
15-24	\$700	\$800
25+	\$675	\$775

Go to <http://www.nofluffjuststuff.com/conference/reston/2009/11/home> and register today!

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

### Friday, Nov. 6

12:00 - 1:00 PM : REGISTRATION

1:00 - 1:15 PM : WELCOME

1:15 - 2:45 PM - Sessions

#### **Session #1 : 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.

#### **Session #2 : 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.

#### **Session #3 : 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.

#### **Session #4 : 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 repeatedly 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

#### **Session #5 : 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

#### **Session #6 : Preproduction: Everything You Need to Do Before Iteration 1 by David Hussman**

Many agilists take little time to prepare for the first planning session of their first iteration on a new project. They dive right into the "work" and, sometimes, ultimately deliver software that lacks much value. Some newly formed teams believe that collocation breeds instant success and altogether ignore early planning. While sitting together always helps, it does not mean that people spontaneously collaborate to create sustainable value. Before holding the first planning session, preproduction helps communities learn about each other, the value they will deliver, and their newly forming ecosystem.

2:45 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

#### **Session #7 : 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.

#### **Session #8 : 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.

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

### Session #9 : 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.

### Session #10 : 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

### Session #11 : 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.

### Session #12 : Discovering Real Value with Story Maps and Personas by David Hussman

While actors and use cases often left users behind, personas and story maps bring the users to life and help mine real value. This session will teach you how to craft personas and use them to drive value into your development stream. The tools presented will help you better understand your buyers and users and build strong product backlogs and product road maps.

4:45 - 5:00 PM : BREAK

5:00 - 6:30 PM - Sessions

### Session #13 : 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*

### Session #14 : 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.

### Session #15 : 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.

### Session #16 : 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.

### Session #17 : 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

### Session #18 : Architecture and Agility Are Not Enemies by David Hussman

Being agile does not mean living life one iteration at a time. Agile projects without a long view can run into the common design problems of the past. Planning iteration by iteration is often foolish and feeds the myth that agile projects do not think beyond a few weeks.

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

Successful agile projects plan within iterations and across iterations. The later planning is called release planning and it is the forum where agility first engages architecture and other cross cutting concerns.

6:30 - 7:15 PM : DINNER

Keynote: The Busy Developer's Guide to Iconoclasm - Ted Neward

## Saturday, Nov. 7

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### Session #19 : 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

### Session #20 : 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.

### Session #21 : RESTful Grails by Scott Davis

"Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction." (Albert Einstein) REST and Resource-Oriented Architecture (ROA) are popping up in technical discussions more and more frequently. Here, you'll see practical examples of adding RESTful web services to your Grails application.

### Session #22 : Spring 3.0 Overview by Scott Leberknight

The Spring framework has simplified Java enterprise and web development since 2003, and has been a major innovator in improving and simplifying Java server-side programming since its inception. This session will look at the new features in Spring 3.0 as well as what's being removed from the Spring core.

### Session #23 : 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.

### Session #24 : Connecting Companies with Acceptance Testing by David Hussman

While more companies are not waiting to test, testing is still something that is too often owned by the testers. This session provides tools and techniques which draw on the value automated testing provides while also using it as a core tool to help connect community. From idea to implementation, we will discuss and practice various ways to connect business, development and testing around the value captured in the acceptance tests and their use.

10:30 - 11:00 AM : BREAK

11:00 - 12:30 PM - Sessions

### Session #25 : 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

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

### **Session #26 : Tackling Concurrency on the JVM by Venkat Subramaniam**

In this presentation we will take a quick walk through the issues with concurrency and how the solutions provided in Scala and Clojure help address those.

### **Session #27 : 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?

### **Session #28 : Groovier Spring (More Flexible Applications With Spring and Groovy) by Scott Leberknight**

Spring provides a solid foundation for web and enterprise applications. Its support for dynamic languages like Groovy adds interesting capabilities that can make your application architecture more flexible and dynamic.

### **Session #29 : 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.

### **Session #30 : What Is Lean and Why Should You Care? by David Hussman**

Whether it was intentional or not, the agile community has been borrowing successful ideas from the lean manufacturing for years. Lean practices, like finding and removing wasteful work, can be applied without needing special permission or certification. Ideas like kanban (visual planning aids) and kaizen (continuous learning) are simple, helpful tools that are easily applied and produce great results.

12:30 - 1:30 PM : LUNCH

1:30 - 3:00 PM - Sessions

### **Session #31 : 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.

### **Session #32 : Easy BDD with Groovy by Andrew Glover**

The Manifesto for Agile Software Development essentially focuses on meeting customer needs through reducing wasteful activities. For example, Agile developmental practices push for reducing repetitive documentation and for a rapid acceptance of change; yet, achieving these goals is by no means easy. While a process can enable increased collaboration, for instance, there are various tools that can effectively implement Agile principles. Once such tool is easyb ([www.easyb.org](http://www.easyb.org)), which is a Groovy based domain specific language, which facilitates collaboration by bridging those that define requirements (i.e. customers) and those who turn requirements into code (i.e. development). With easyb, collaborative teams can develop stories in a specific format which are then implemented as tests through a framework which marries the underlying application. This test suite enables change and produces accordance among Agile teams in short order.

### **Session #33 : GWT fu, Part 1 by David Geary**

Learn to implement web applications with GWT. *Prerequisite: Familiarity with a component-based framework, preferably a desktop application framework*

### **Session #34 : Real World Hibernate Tips (Reloaded) by Scott Leberknight**

Hibernate is a very powerful object/relational mapping framework. This session contains a new set of Hibernate tips, tricks, and pitfalls.

### **Session #35 : 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.

### **Session #36 : 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

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

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.

3:00 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

### **Session #37 : 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*

### **Session #38 : 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*

### **Session #39 : GWT fu, Part 2 by David Geary**

Learn to do amazing stuff with GWT. **Prerequisite:** *GWT fu, Part 1 is not a prerequisite for this session, but it will help if you have some familiarity with GWT.*

### **Session #40 : Polyglot Persistence by Scott Leberknight**

Polyglot persistence is all about considering your persistence requirements and selecting a persistence mechanism that best meets those requirements, as opposed to selecting an RDBMS as the default choice. In this session we'll look at some of the persistence alternatives that are available like Amazon SimpleDB, CouchDB, Google Bigtable, and more.

### **Session #41 : 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.

### **Session #42 : 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.

4:45 - 5:30 PM : BIRDS OF A FEATHER SESSION

## **Sunday, Nov. 8**

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### **Session #43 : Architecting Code for Concurrent Execution: Theory and Practice by Robert Fischer**

The power of multicore machines and cloud computing is all dependent on an application's ability to successfully leverage concurrency. Although concurrency has traditionally been considered fatally difficult in Java, a few simple architecture principles can make all the difference. This session will review some of those principles in both theory and practice.

### **Session #44 : 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

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

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.

### **Session #45 : Flex and Java integration by Shashank Tiwari**

Flex is a leading rich internet application development framework and Java is the most pervasive of enterprise computing environments. In this session you will learn to combine the two effectively and leverage a robust server side with a highly interactive user interface.

### **Session #46 : REST : Web Architecture for Rich Clients by Brian Sletten**

The failure of a Service-Only Architecture (SOA) is that it fails to highlight the data that flows through it. We must embrace a software architecture that puts information first. Who wants what? How do they want to use it? This blended vision that handles data, documents, services and concepts. There is tremendous interest in REpresentational State Transfer (REST) as an architectural style for building scalable, flexible, information-driven architectures in the Enterprise. The success of the Web has caught our attention in the face of increased complexity and many failures with more traditional Web Services technologies. The problem is that it is difficult to sell a way to do things. Managers do not want to feel like they are innovating in the middleware space. They want to understand why they should deviate from the blue prints laid down by the industry leaders. They want to understand when they should use REST, when they should use SOAP and when they might fallback to regular old Java-based messaging. They want to make business-based technology decisions that lay a path to forward progress rather than paying for technological flux.

### **Session #47 : 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.

### **Session #48 : IZero: Starting Projects Right by Stuart Halloway**

If an iteration is the heartbeat of an agile development process, then Iteration Zero (IZero) creates the heart. While you can (and should) retrospect and adjust throughout the software lifecycle, few things are as valuable as a good start. In this talk, you will learn how we run Iteration Zero at Relevance.

10:30 - 11:00 AM : MORNING BREAK

11:00 - 12:30 PM - Sessions

### **Session #49 : The Concurrency Toolset: JConch, Google Collections, and java.util.concurrent by Robert Fischer**

JConch is a library that provides a few high-level tools for high-concurrency environments on the JVM. The java.util.concurrent package in the Java standard library provides low-level structures for managing concurrent communication. Learn here how to use both of them in order to produce clean, highly-concurrent, and highly-tunable code.

### **Session #50 : Open Source Debugging Tools for Web Apps by Matthew McCullough**

This session will survey a wide range of tools across the Web application debugging space, covering the REST, HTML, SOAP, CSS, TCP, Filesystem and JavaScript facets of an app. We'll look at utilities such as tcpdump, curl, Wireshark, JMeter, Firebug, JASH, Poster, SoapUI, Firediff, Isof, fs\_usage, iwatch 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.

### **Session #51 : Tactical Continuous Integration with Hudson by Andrew Glover**

This session will walk attendees through a series of iterations on a fictional Java project where an automated build system is created that facilitates compilation, testing, inspection, and deployment. This build system is then plugged into the Hudson CI server and as features are coded using Agile techniques like developer testing, attendees will ultimately see firsthand how a Continuous Integration process reduces risk and improves software quality.

### **Session #52 : 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 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. **Prerequisite:** *The Semantic Web: The Future Now would be helpful, but not required*

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

### **Session #53 : Java.next: Clojure, Groovy, JRuby, and Scala by Stuart Halloway**

In this talk, we will explore and compare four of the most interesting 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/2008/9/24/java-next-overview>.

### **Session #54 : Estimating vs. Guessing - How Agile Teams Estimate Their Work by David Bock**

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

12:30 - 1:15 PM : LUNCH

1:15 - 2:15 PM : EXPERT PANEL DISCUSSION

2:15 - 3:45 PM - Sessions

### **Session #55 : Grails for the Enterprise by Robert Fischer**

The Grails web application is an innovative hybrid of best-of-breed Java technologies and dynamic/convention-based development. The result is a powerful, flexible, exciting framework that still fits comfortably into enterprise stacks. This session introduces Grails, but approaches it from the perspective of an enterprise web development stack, in order to see how Grails works well in mid-size and mature development shops.

### **Session #56 : 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*

### **Session #57 : Flex and Hibernate by Shashank Tiwari**

A complete journey into the challenges and solutions for effective integration of Flex and Hibernate.

### **Session #58 : 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*

### **Session #59 : 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.

### **Session #60 : Maintaining Source Code Quality (The Project Integrity Series) by David Bock**

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? Have you ever been the 'new guy' on a project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

3:45 - 4:00 PM : BREAK

4:00 - 5:30 PM - Sessions

### **Session #61 : Leveraging the cloud with Amazon Web Services by Andrew Glover**

Amazon's S3 and EC2 offerings are publicly available services that enable you to run virtual machines and store (and retrieve) digital assets (i.e. images, music, documents, etc). In this session, we'll cover the ins and outs of S3 & EC2 and see first hand how to leverage them for various purposes -- either personal or in the enterprise.

### **Session #62 : 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

# Northern Virginia Software Symposium

## -Session Schedule-

(event schedule as of November 2, 2009)

start to iPhone application coding in Objective-C and integration with some of our favorite Java web service back-ends such as RESTful Grails.

### **Session #63 : Collaborative real-time RIA by Shashank Tiwari**

In this session, learn to craft and create collaborative rich internet applications, that are responsive and updated in real-time for streamlined decision making and business intelligence harnessing. Understand how in-time communication can smoothen information exchange, reduce errors and increase productivity.

### **Session #64 : 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*

### **Session #65 : 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.

### **Session #66 : Managing Complexity (The Project Integrity Series) by David Bock**

How many times have you started a new project only to find that several months into it, you have a build process that mysteriously fails, a bunch of 'TODO' and 'FIXME' comments in the source, and problems that come and go because "it works on my machine"? Does your project have a little bit of 'folk wisdom' that isn't well-known, but is necessary to get things done? How easily could you recreate your development environment if you got a new machine today?