

# New England Software Symposium

Sheraton Ferncroft

March 20 - 22, 2009

<http://www.nofluffjuststuff.com/conference/boston/2009/03/index.html>

Fri, Mar. 20, 2009					
	1	2	3	4	5
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Real-world Refactoring Neal Ford	The Busy Java Developer's Guide to Java7 Ted Neward	Java Memory, Performance and the Garbage Collector Ken Sipe	On Being a Software Architect Mark Richards	Agile Testing Venkat Subramaniam
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Effective Java Venkat Subramaniam	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Debugging your Production JVM Ken Sipe	Common AntiPatterns and How To Avoid Them Mark Richards	The Productive Programmer: Mechanics Neal Ford
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Know your Groovy? Venkat Subramaniam	The Busy Java Developer's Guide to Advanced Platform Security Ted Neward	What's New in Spring 3 Ken Sipe	Transaction Pitfalls and Strategies Mark Richards	Hands-on Agile Development Neal Ford
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Jared Richardson				

Sat, Mar. 21, 2009					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Programming Scala Venkat Subramaniam	The Busy Java Developer's Guide to Collections Ted Neward	7 Habits of Highly Effective Developers Ken Sipe	Emergent Design & Evolutionary Architecture Neal Ford	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Cleaning up Code Smell Venkat Subramaniam	The Busy Java Developer's Guide to the OpenJDK Ted Neward	Architecture and Scaling Ken Sipe	Meta-programming JRuby for Fun & Profit Neal Ford	Techniques 2009 Jared Richardson
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Creating a real world application in Grails - A Case Study Mark Johnson	REST : Information-Driven Architectures for the 21st Century Brian Sletten	Test Driven Design Neal Ford	Introduction to JMS Mark Richards	Agile Anti-Patterns Jared Richardson
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Git control of your source Stuart Halloway	RESTlet for the Weary Brian Sletten	The Busy Java Developer's Guide to Hacking with the JDK Ted Neward	Advanced Topics in JMS Mark Richards	Code Metrics & Analysis for Agile Projects Neal Ford
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSION				

Sun, Mar. 22, 2009					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Pragmatic Domain Specific Languages Neal Ford	Introduction to NetKernel : Software for the 21st Century Brian Sletten	Groovy Closures - The way to cleaner code Mark Johnson	Developer Essentials Srivaths Sankaran	Agile, Relevance Style Stuart Halloway
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Architecture Rules Enforcement using Aspects Sri Penchikala	Advanced NetKernel : Software for the 21st Century Brian Sletten	Groovy and Webtest - Produce verifiable tests without the expense Mark Johnson	Shining a Light on Eclipse Srivaths Sankaran	Taking Agile From Tactics to Strategy Stuart Halloway
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Domain Driven Design & Development with Spring Portfolio Sri Penchikala	Semantic SOA : Meaningful Service Strategies Brian Sletten	High Performance Projects Srivaths Sankaran	Java.next #1: Common Ground Stuart Halloway	10 Principles for Software Estimation : It Does not have to be that hard! Mark Johnson
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Application Architecture Trends - Where We Have Been, Where We Are Going Sri Penchikala	XMPP For the People : Smack and OpenFire Brian Sletten	Become a Maven maven Srivaths Sankaran	Programming Clojure Stuart Halloway	The Software Manager's Dashboard: Getting the information you really need Mark Johnson

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

### Friday, Mar. 20

12:00 - 1:00 PM : REGISTRATION

1:00 - 1:15 PM : WELCOME

1:15 - 2:45 PM - Sessions

#### **Session #1 : 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 cruddy 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 #2 : 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.

#### **Session #3 : Java Memory, Performance and the Garbage Collector by Ken Sipe**

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

#### **Session #4 : 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 #5 : Agile Testing by Venkat Subramaniam**

How is testing done on agile projects? Do we need testers when programmers can write tests? When do we do test? If the requirements are evolving, should we postpone testing until they stabilize?

2:45 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

#### **Session #6 : 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.

#### **Session #7 : 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 #8 : Debugging your Production JVM by Ken Sipe**

So your server is having issues? memory? Connections? Limited response? Is the first solution to bounce the server? Perhaps change some VM flags or add some logging? In today's Java 6 world, with its superior runtime monitoring and management capabilities the reasons to the bounce the server have been greatly reduced.

#### **Session #9 : 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

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

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 #10 : 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!

4:45 - 5:00 PM : BREAK

5:00 - 6:30 PM - Sessions

### Session #11 : Know your Groovy? by Venkat Subramaniam

Groovy brings the dynamic productivity to the Java platform. One of the strengths of Groovy is the seamless integration with Java—it preserves the Java semantics. However, Groovy does have some differences that can surprise you if you're not expecting.

### Session #12 : 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 #13 : 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

### Session #14 : 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 #15 : 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.

6:30 - 7:15 PM : DINNER

Keynote: Career 2.0: Take Control of Your Life - Jared Richardson

## Saturday, Mar. 21

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### Session #16 : 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 #17 : 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.

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

### **Session #18 : 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.

### **Session #19 : 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 #20 : Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson**

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

10:30 - 11:00 AM : BREAK

11:00 - 12:30 PM - Sessions

### **Session #21 : 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 #22 : The Busy Java Developer's Guide to the OpenJDK by Ted Neward**

With the release of the OpenJDK source code, Java developers have been given a unique opportunity to peer inside the hood of the JVM, see what's there, and how Java code actually executes.

### **Session #23 : Architecture and Scaling by Ken Sipe**

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

### **Session #24 : Meta-programming JRuby for Fun & Profit by Neal Ford**

Ruby is the revenge of the Smalltalkers. Not since Smalltalk has a language had such powerful meta-programming facilities. While this may seem like a minor feature, it turns out that surgical meta-programming allows solutions to problems that are clearer, more concise, more maintainable, and take orders of magnitudes fewer lines of code.

### **Session #25 : Techniques 2009 by Jared Richardson**

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

12:30 - 1:30 PM : LUNCH

1:30 - 3:00 PM - Sessions

### **Session #26 : Creating a real world application in Grails - A Case Study by Mark Johnson**

You have probably heard about Grails, a Rails based development framework which claims to make software development a breeze. You have heard the hype and are now wondering about the reality, will it really work in your environment? Has it really lived up to the hype? This presentation reviews the results using Grails to develop the New England Java Users Group event management web site to answer these questions.

### **Session #27 : 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.

### **Session #28 : 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.

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

### Session #29 : 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 #30 : Agile Anti-Patterns by Jared Richardson

Agile is wildly popular in some circles and hated in others. How can the same ideas cause such different reactions? Sometimes it's the definition of "agile" and other times it's company culture, but there's usually a good reason when Agile ideas are thrown out on their collective ears.

3:00 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

### Session #31 : 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.

### Session #32 : RESTlet for the Weary by Brian Sletten

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community. **Prerequisite:** REST (unless you are very comfortable with REST)

### Session #33 : The Busy Java Developer's Guide to Hacking with the JDK by Ted Neward

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

### Session #34 : 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 and batching, transacted sessions, client-based acknowledgement, message correlation alternative, 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

### Session #35 : Code Metrics & Analysis for Agile Projects by Neal Ford

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

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

## Sunday, Mar. 22

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### Session #36 : Pragmatic Domain Specific Languages by Neal Ford

This talk covers practical concerns for both writing and consuming DSLs, covering internal DSLs, DSL building patterns and antipatterns, and how to leverage language features to make writing DSLs easier.

### Session #37 : 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

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

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.

### **Session #38 : Groovy Closures - The way to cleaner code by Mark Johnson**

The factory patterns and callbacks have been around for a long time as a technique to provide flavor specific code variations. But they are awkward and hard to update. Enter Groovy closures. Imagine having the ability to inject different coding flavors using code closures. If you need a different flavor, then just pass a different code block. Now imagine that all of this works on the JVM!

### **Session #39 : Developer Essentials by Srivaths Sankaran**

Creating great code gets harder every day. But using the right tool, and embracing tried & tested habits can greatly improve your productivity. This session will focus on such tools and other quality enhancing techniques. You will also learn specific habits that will help you shine and elevate you above the masses.

### **Session #40 : 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.

10:30 - 11:00 AM : MORNING BREAK

11:00 - 12:30 PM - Sessions

### **Session #41 : Architecture Rules Enforcement using Aspects by Srinu Penchikala**

Even though many companies have some kind of application architecture standards, they don't usually have a mechanism to enforce those standards. As a result of this lack of architecture governance, the Implementation (Code) often doesn't match the Requirements (Reference Architecture). Enforcing Reference Architecture guidelines promotes consistency and modularity in the System. It also helps in detecting structural complexity and preventing it earlier in the software development process. As a result, the application code is modifiable, portable, and testable. In this presentation, I will talk about the significance of enforcing the architecture rules and standards and how to actually enforce them in software development projects. I will start off the discussion explaining various types of architecture rules that range from simple module dependencies to enforcing how APIs and specific idioms should be used. All these rules usually fall into categories such as Layered Architecture, Separation Of Concerns, Domain Driven Design, and Infrastructure. Some of the rules examples will include basic validations like DAO (Persistence) layer shouldn't depend on any other layers, Presentation layer should not use DAO classes directly, and Service layer should never call web layer. Separation of concerns rules are no transaction management in DAO classes and Service layer should be transactional. More sophisticated rules like that are used in a Domain Driven Design project such as Service object creation through a Factory object (i.e. no Service instantiation using the new keyword), No direct access to DAO's except from the Domain classes (go through Domain or Repository objects), and Business service that fails with a concurrency related failure can be retried.

### **Session #42 : 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)*

### **Session #43 : Groovy and Webtest - Produce verifiable tests without the expense by Mark Johnson**

The problem with testing is you need to make certain that the test is validating a "true" business requirement. Most business folks can't read code, so they have to trust that we are testing the right stuff. The reports generated as part of the webtest process are actually human readable, so you can validate with the business stakeholders that your tests properly cover "true" requirements. Throw in Groovy and closures into the mix with WebTest and you have the ability to implement many high powered, clearly understandable tests taking full advantage of re-usable code allowing for many different tests with the same code block.

### **Session #44 : Shining a Light on Eclipse by Srivaths Sankaran**

The Eclipse IDE has matured over several years to become quite a powerful tool with a robust plugin-based architecture. It gives commercial IDEs such as IDEA a run for their money. Yet, most developers barely scratch the surface of its capabilities. This talk will expose you to features that will greatly enhance your productivity. You will learn tips and tricks that will make you more efficient in your development activities and allow you to focus on solving the business problem at hand. Along the way you will be introduced to improvements introduced by the recent Ganymede release.

### **Session #45 : 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.

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

12:30 - 1:15 PM : LUNCH

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

2:15 - 3:45 PM - Sessions

### **Session #46 : Domain Driven Design & Development with Spring Portfolio by Srini Penchikala**

In this technical session, I will discuss Domain Driven Design and Development from a practical implementation stand-point. The presentation looks at various architectural and design factors, best practices, frameworks and tools that affect the design of a business domain implementation project. It also covers the impact of various design concerns like Dependency Injection (DI), Aspect Oriented Programming (AOP), Persistence, Caching, Transaction Management and Application Security in Domain Modeling and Domain Driven Design problem space. The session also provides the domain model and code examples based on a real world DDD implementation project that utilized agile software development techniques such as SCRUM, Test Driven Development and Refactoring to realize a great success in completing the project on time and under budget.

### **Session #47 : 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 #48 : High Performance Projects by Srivaths Sankaran**

Unsuccessful projects dominate the landscape of IT projects by a wide margin. Retrospectives are common occurrences and the advent of newer project delivery techniques hasn't proven to be the miracle cure. Some projects have lemon written all over them at the time of arrival and others just wither away for want of proper care and feeding. This talk introduces you to a variety of project problems and how to address them. You will learn techniques that will get you off on the right foot and habits that will ensure smooth progress throughout the project's lifecycle. The net result will be a project that hits its target and a customer that is happy.

### **Session #49 : 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>.

### **Session #50 : 10 Principles for Software Estimation : It Does not have to be that hard! by Mark Johnson**

As developers we dread when management requests a project estimate. Typically, you do not have the opportunity to understand all the requirements, the team composition is unknown, and you have been given until tomorrow end of day to produce an estimate. Several months later everyone is yelling at you about the software estimation errors encountered during the project.

3:45 - 4:00 PM : BREAK

4:00 - 5:30 PM - Sessions

### **Session #51 : Application Architecture Trends - Where We Have Been, Where We Are Going by Srini Penchikala**

Java Application Architecture is going through a major paradigm shift in terms of design techniques, technologies, and frameworks that are used to build and deploy Java applications. Enterprise JavaBeans (EJB), traditional Message Queues (JMS), and even Application Servers as we know them are being replaced by light-weight POJO based frameworks such as Spring, ActiveMQ, and OSGi compatible containers. This technical session will give an overview of Java application architectures of the past where EJB's, verbose EAR files and heavy-weight J2EE application servers were the only choice a Java developer had to develop and implement Java applications to the current pragmatic architectures where the concepts like POJO's and Domain Driven Design (DDD) have become the core design and development concerns like they should be. The presentation will also include a discussion on how concerns like Persistence, Transaction Management, Application Security and Asynchronous Messaging have become the infrastructure concerns that are managed by the frameworks (like Spring) out-of-the-box instead of developers having to spend a lot of time and effort in programming or dealing with complex configuration files and deployment descriptors for implementing these concerns. The presentation will talk about the emerging design techniques like Domain Driven Design, Domain Specific Languages (DSL), Custom Annotations, Dependency Injection (DI), Aspect-Oriented Programming (AOP) and OSGi. I will also discuss the use cases where these techniques add value to the architecture and where they may be just an overkill.

### **Session #52 : XMPP For the People : Smack and OpenFire by Brian Sletten**

Communication is a key piece of organizational success. We are constantly trying to find new ways to improve how we share ideas, concepts, issues and needs. E-mail has lost its luster. Phone calls are too disruptive. We want a combination of presence-aware and asynchronous. Instant messaging has started to fill this void but using public services are often inappropriate for sensitive

# New England Software Symposium

## -Session Schedule-

(event schedule as of March 18, 2009)

conversations. This talk will focus on two open source offerings: the OpenFire XMPP (Jabber) server and the Smack client library. Come hear how you can adopt these technologies in your daily activities more fully.

### **Session #53 : Become a Maven maven by Srivaths Sankaran**

Wouldn't it be nice if you didn't have to worry about mundane project administrivia? What if all your build, deployment, configuration and documentation problems just disappeared? We will see how you can do just that with Maven. This session will introduce you to Maven. It will teach you how to use it to set up your project & effortlessly manage dependencies. You will learn how easy it is to tailor it for your specific needs. Using Maven, developing professional documentation including reports from several popular quality and metrics gathering tools is a snap. By the end of this session, you will be a true Maven maven.

### **Session #54 : 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.

### **Session #55 : The Software Manager's Dashboard: Getting the information you really need by Mark Johnson**

When we start a project, our management hands us a copy of MS Project and using this tool we are expected to accurately track the project to completion. What often ends up happening is many of the project tasks are listed as 90% complete and you don't have a clear understanding of the blocking reasons. This presentation will explore various vendor independent time efficient dashboard options you can pursue to properly track your project.