

Twin Cities Software Symposium

Hilton Minneapolis/Bloomington Hotel

October 10 - 12, 2008

<http://www.nofluffjuststuff.com/conference/minneapolis/2008/10/index.html>

(event schedule as of October 9, 2008)

Fri, Oct. 10, 2008					
	Salon A	Salon B	Salon C	Washington	Jefferson
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	A Thorough Introduction To Groovy Jeff Brown	Caring about your Code Quality Venkat Subramaniam	JavaServer Faces: A Whirlwind Tour David Geary	Agile, Smagile: What's Working? - What's Not? David Hussman	The Busy Java Developer's Guide to Annotations Ted Neward
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Agile Test Driven Development With Groovy Jeff Brown	Know your Java? Venkat Subramaniam	Facelets David Geary	Agile Management & Managing Agility David Hussman	The Busy Java Developer's Guide to ClassLoaders Ted Neward
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Powerful Metaprogramming Techniques With Groovy Jeff Brown	Acceptance Testing Application Behavior Venkat Subramaniam	Using Ajax4jsf David Geary	Coaching and Leading Agile Projects David Hussman	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Ted Neward				

Sat, Oct. 11, 2008					
	Salon A	Salon B	Salon C	Washington	Jefferson
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Grails - Agile Web 2.0 The Easy Way Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward	Architecture and Scaling Ken Sipe	10 Tips for Getting Your Project Back on Track Jared Richardson	Refactoring JavaScript Stuart Halloway
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Advanced Web Development With Grails Jeff Brown	The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward	Intro to Seam David Geary	Techniques 2008 Jared Richardson	How to Fail with 100% Code Coverage Stuart Halloway
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Git control of your source Stuart Halloway	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Spring 2.5 - Spring without XML Ken Sipe	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Java.next #1: Common Ground Stuart Halloway	Build Teams, Not Products Jared Richardson	Boosting Programmer productivity with Mylyn Brian Sam-Bodden
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION				

Sun, Oct. 12, 2008					
	Salon A	Salon B	Salon C	Washington	Jefferson
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	Design for Operations Michael Nygard	Iteration 0 Ken Sipe
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Testing the Web Layer Nathaniel Schutta	10 ways to use Hibernate effectively Brian Sam-Bodden	SOAs Challenges Ken Sipe	Automating Customer Acceptance David Hussman
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Clouds, Grids, and Fog Michael Nygard	Designing for Ajax, part 1 Nathaniel Schutta	Rapid Web Development with Grails and Ajax Scott Davis	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	Architecture and Agility Are Not Mutually Exclusive David Hussman
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	The 90-Minute Startup Michael Nygard	Designing for Ajax, part 2 Nathaniel Schutta	Groovy, Grails and Google Maps: Mashups 101 Scott Davis	Advanced Rules Programming with Drools Brian Sam-Bodden	Agile Takes More than Developers Stevie Borne

Twin Cities Software Symposium

Hilton Minneapolis/Bloomington Hotel

October 10 - 12, 2008

<http://www.nofluffjuststuff.com/conference/minneapolis/2008/10/index.html>
(event schedule as of October 9, 2008)

Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden

Learn how to build featured rich applications using the Eclipse Rich Client Platform. The Eclipse platform is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

Boosting Programmer productivity with Mylyn by Brian Sam-Bodden

Mylyn is a task-focused toolkit for the Eclipse IDE that allows developers to focus on tasks in a way that they never been able to do before. Mylyn eliminates the constant context switching produced by typical ways IDEs are used. No more scrolling/browsing/searching/tagging/sending emails with progress updates... Mylyn provides a new way of working that allows you to focus on specific tasks by reducing information overload. Mylyn also provides a framework for integrating with the most commonly usage task tracking systems and version control systems. In this talk you'll learn how Mylyn can boost your productivity as a Java developer by letting you get the most out of your IDE.

Beginning Object-Relational Mapping with Hibernate by Brian Sam-Bodden

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

10 ways to use Hibernate effectively by Brian Sam-Bodden

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

Beginning Drools - Rule Engines in Java by Brian Sam-Bodden

Drools is an open source pure-Java implementation of a forward chaining rules engine. Drools can be used in a J2SE or J2EE application and allows you to express rules programatically or by building domain specific rule languages. Learn how Business Rules with Drools can make your Java applications more flexible and robust.

Advanced Rules Programming with Drools by Brian Sam-Bodden

In this session you'll learn some of the more advanced features of Drools; a pure-Java Rule Engine. This session will walk through the construction of an advanced Rules application covering such topics as: - Fine control and monitoring of a Working Memory session - Using Decision Tables - Advanced Rule Language Features - Building Domain Specific Languages - Managing your Rules **Prerequisite:** *Beginning Drools*

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

Facelets by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

Using Ajax4jsf by David Geary

Ajax4jsf makes it very easy to add Ajax to your JSF applications. Come to this presentation to see how.

Intro to Seam by David Geary

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0.

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

Agile, Smagile: What's Working? - What's Not? by David Hussman

With the growth of agile comes the need to add a new line to the Agile Manifesto: Success over Dogma. The number of people who can say agile is growing faster than the number of people benefiting from agile practices. There are now many successful agile projects, yet there are also a growing number of projects claiming to be agile but not seeing any of the benefits agile methods provide. This session will discuss successful adoptions of agile, dumb things you can do to muck it up, and more.

Agile Management & Managing Agility by David Hussman

Management and agility are not mutually exclusive. Many managers are already working in an agile manner as a means to improve, produce, or simply survive. Other managers hear about projects using agile methods and struggle to find a place in the project community. This session provides a new way to think about managing projects. Some managers will find that their existing practices and skills are supported and enhanced by the forums and metrics provided within an agile project while others will be challenged by some of the principles and practices.

Coaching and Leading Agile Projects by David Hussman

Successfully coaching agile communities involves using a wide variety of skills. Coaches help guide coding and design, collaboration and communication, the writing and telling of user stories and much more. The coach needs to continuously show and teach the varied interactions that connect and support the entire project community. This session will explore and teach coaching skills. The session will reference a wide variety of agile coaching as well as drawing from cross disciplinary techniques like those used by music producers to help foster creativity while helping to ensure products are delivered and challenges confronted.

Automating Customer Acceptance by David Hussman

Why should the value of test driven development (TDD) stay stuck in the realm of coding? The ideas behind TDD are now being successfully applied to the automation of business value. While this has been going on for some time within the agile community, it is not starting to spread to main stream development. There are more tools coming available everyday which allow developers, testers, and customers (or product owners) to work together to automate acceptance tests. This process helps clarify the needs of the end user before development begins and removes more of the wasteful work based on incorrect assumptions from vague requirements.

Architecture and Agility Are Not Mutually Exclusive 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. 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.

10 Tips for Getting Your Project Back on Track by Jared Richardson

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

Techniques 2008 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.

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.

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Powerful Metaprogramming Techniques With Groovy by Jeff Brown

Metaprogramming is a key component in building truly dynamic and flexible applications with Groovy. Groovy's metaprogramming capabilities bring great new possibilities to the table that would be very difficult or just plain impossible to write with Java alone. This session will demystify a lot of the magic that seems to be going on inside of a Groovy application. **Prerequisite:** *A Thorough Introduction To Groovy*

Grails - Agile Web 2.0 The Easy Way by Jeff Brown

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

Advanced Web Development With Grails by Jeff Brown

Grails makes web application development both fun and easy. This session dives beyond the basics to cover advanced details of Grails that bring the really exciting features to your applications. **Prerequisite:** *Grails - Agile Web 2.0 The Easy Way*

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?

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none

make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

Iteration 0 by Ken Sipe

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

SOAs Challenges by Ken Sipe

SOA... Is it hype? What's real... and what's not? What is the right abstraction level?

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.

Clouds, Grids, and Fog by Michael Nygard

Servers, storage, networking, backups... they're all vanishing into the "clouds". Cloud Computing is the emerging architecture for massive, scalable infrastructure that your company doesn't have to own or operate. From the "zero servers" web startup to the corporate IT department battling server-sprawl, cloud computing has many manifestations. This session will differentiate among the various types of cloud computing and describe applicable use cases.

The 90-Minute Startup by Michael Nygard

Cloud computing is taking the world by storm. Amazon's Web Services, EC2, and S3 provide completely virtual infrastructure, letting startup and existing companies create sites and web applications faster than ever before. In this session, Michael will use cloud computing to create and deploy a fully-functional web site. You will learn how to create and run your own virtual infrastructure in the clouds.

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

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

Testing the Web Layer by Nathaniel Schutta

While your project might have nearly 100% code coverage on the server tier, many projects ignore testing the web layer. With more and more code being pushed to the browser, a lack of tests for the client code begs for trouble.

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.

Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

Rapid Web Development with Grails and Ajax by Scott Davis

Grails is a Java- and Groovy-based web framework that is built for speed. First-time developers are amazed at how quickly you can get a page-centric MVC web site up and running thanks to the scaffolding and convention over configuration that Grails provides. Advanced web developers are often pleasantly surprised at how easy it is to break out of that coarse-grained navigation model using the native Ajax support baked into the framework.

Groovy, Grails and Google Maps: Mashups 101 by Scott Davis

Groovy is a new dynamic language that dramatically speeds up Java development. Grails is a complete web framework in a box, including a web server and a database. Google Maps allows you to add maps to your webpage in a few lines of code. Put all three together and you are built for speed.

Agile Takes More than Developers by Stevie Borne

In agile teams, are testers needed if the developers write automated tests? Is there a need for BA's if the developers are supposed to talk directly to the customer/business? How do PM's fit into an agile team?

Refactoring JavaScript by Stuart Halloway

The rise of Ajax and Rich Web Applications, plus the success of dynamic languages, has caused people to revisit the JavaScript language. Now that we take JavaScript seriously as a language, it is time to get serious about the quality of JavaScript code, through refactoring. In this talk, we will approach refactoring JavaScript in three phases: Test first, then refactor. Bring JavaScript code under test, so that you can refactor with confidence. Refactoring 101. Explore some important refactorings: composed method, extract method, introduce named parameter, and extract object Common problems. Work through three problems endemic to legacy JavaScript code: making JavaScript unobtrusive, refactoring to prototype-based inheritance, and refactoring to functional style.

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.

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.

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>.

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

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

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 Hacking (on) 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.

Why the Next Five Years Will Be About Languages by Ted Neward

Thanks to the plateau of per-chip performance increases and the resulting need to work better with multi-core CPUs, the relative difficulty of mapping user requirements to general-purpose programming languages, the emergence of language-agnostic "virtual machines" that abstract away the machine, the relative ceiling of functionality we're finding on the current crop of object-oriented languages, and the promise and power of productivity of dynamically-typed or more loosely-typed languages, we're about to experience a renaissance of innovation in programming languages.

The Busy Java Developer's Guide to Concurrency (Part 1: Threads) by Ted Neward

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the java.util.concurrent packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown. **Prerequisite:** *The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

Caring about your Code Quality by Venkat Subramaniam

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.

Know your Java? by Venkat Subramaniam

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a

look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.

Acceptance Testing Application Behavior by Venkat Subramaniam

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.