

Talks

Abstraction Distractions

Neal Ford

Computer science is built on a shaky tower of abstractions, but we've been distracted by other things until we believe it is reality. This talk teases apart some of the tangled abstractions that have become so common they are invisible yet impact important decisions. I cover languages, tools, platforms, and burrow all the way down to fundamental concepts. This wide-ranging keynote answers these questions and more:

- Which language should I learn next? And should I bother?
- Why is the iPad the most revolutionary device in the last 30 years?
- Why do some people hate Maven so much?
- How can I choose technologies with long shelf lives?
- Is hiding always a good thing?
Oh, and some jokes too.

Deception and Estimation

Linda Rising

Cognitive scientists tell us that we are hardwired for deception. It seems we are overly optimistic, and, in fact, we wouldn't have survived without this trait. With this built-in bias as a starting point, it's almost impossible for us to estimate accurately. That doesn't mean all is lost. We must simply accept that our estimates are best guesses and continually re-evaluate as we go, which is, of course, the agile approach to managing change. Linda Rising has been part of many plan-driven development projects where sincere, honest people with integrity wanted to make the best estimates possible and used many “scientific” approaches to make it happen—all for naught. Re-estimation was regarded as an admission of failure to do the best up-front estimate and resulted in a lot of overhead and meetings to try to “get it right.” Offering examples from ordinary life—especially from the way people eat and drink—Linda demonstrates how hard it is for us to see our poor estimating skills and helps work with the self-deception that is hardwired in all of us.

It could be heaven or it could be hell: On being a Polyglot Programmer

Venkat Subramaniam

This keynote will focus on the power and perils of working with multiple languages.

How Debuggers Work

Karl Rehmer

Karl Rehmer gives us an overview of how debuggers work, set breakpoints, execute programs, display machine code, display and modify memory, and provide for symbolic debugging. Cross debugging methods such as serial communication with a debug monitor and JTAG debugging will be touched on. He will briefly explain common debugging features and some of the ways that they are implemented.

State of Scala

Venkat Subramaniam

Scala, the hybrid functional, fully object-oriented language has evolved over the years. In this presentation we will talk about what has changed in this language in the recent release and look at some cool things you can do with this very powerful language.

Key enhancements and changes in Scala.

Agile UI

Nathaniel Schutta

Some developers assume that agility and usability are mutually exclusive - in reality, they are extremely complimentary; if you squint, you might have a hard time telling the difference between agile practices and good user interface design. This usability talk is aimed squarely at developers giving you the tools you need to develop UIs that won't make your users yack. We'll discuss the importance of observation, personas, paper prototyping, usability testing and the importance of good moderators. In addition, we'll map the various aspects of user interface design to a typical agile iteration.

Some developers assume that agility and usability are mutually exclusive - in reality, they are extremely complimentary; if you squint, you might have a hard time telling the difference between agile practices and good user interface design. This usability talk is aimed squarely at developers giving you the tools you need to develop UIs that won't make your users yack. We'll discuss the importance of observation, personas, paper prototyping, usability testing and the importance of good moderators. In addition, we'll map the various aspects of user interface design to a typical agile iteration.

Java, .Net, Ruby and PHP integration

Matthias Hrynyszak

Over the decades the developer's landscape has changed from one tool for all possible jobs (the almighty machine code) to a number of managed platforms supported by thousands of libraries to do pretty much everything you can imagine. Our job is less and less about inventing new low-level building blocks and becomes increasingly more about putting together existing parts into a business solution.

In this talk we're going to take a look at some of the possibilities to break the monogamy of using just one platform and/or language. We'll look at ways to combine Java with other major players (.NET, PHP and Ruby) and see how it can help us solve business problems in today's world.

Git Going with a Distributed Version Control System

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.

In this talk, we discuss the team changes that liberate you from the central server, but still conform to the corporate expectation that there's a central master repository. You'll get a cheat sheet for Git, and a trail-map from someone who's actually experienced the Subversion to Git transition.

Lastly, we'll even expose how you can leverage 75% of Git's features against a Subversion repository without ever telling your bosses you are using it. Be forewarned that they may start to wonder why you are so much more effective in your checkins than other members of your team.

What's new in Java EE 6

Sang Shin

In the past years, the Java EE platform has grown and matured, and is now able to cover a wide range of enterprise and web application development needs. In addition, the Java EE platform has fostered a vibrant community and marketplace for additional technologies, frameworks, and applications that work with the platform. Some of these provide facilities that are missing from the platform. Others provide alternatives to platform facilities. A major theme for Java EE 6 release is to embrace and support those technologies as part of the overall Java EE landscape, while also continuing to simplify the platform to better target a wider range of developers. To that end Java EE has two major goals - extensibility and profiles. In addition, the theme of "ease of development" is still an important goal.

This session covers these core features of Java EE 6 platform. The concrete set of technologies that make up Java EE 6 platform - Servlet 3.0, EJB 3.1, JSF 2.0, JPA 2.0, Context and Dependency Injection (JSR 299), JAX-RS 1.0, and more - are covered at the high level from the standpoint of these features.

The Future of the Java Platform: Java SE 7 and Java SE 8

Simon Ritter

As it reaches its fifteenth birthday Java has become the most popular programming language in use today, being not just robust, secure and high performant, but suitable for all types of applications. This talk will explain how the Java Standard Edition will evolve over the next two years through the release of Java SE 7 and Java SE 8. Which features will be included and how these should be used will be discussed.

Pro Groovy

Dierk König

Many Java developers have come to appreciate Groovy for its conciseness and scripting capabilities but not many use the language to its full extend. In this talk we will go through some of Groovy's lesser known features by means of live coding. If you seek to make your life as a Java programmer easier by improving your Groovy Kung-Fu, this talk is for you.

Architecture and programming model for NOSQL web (Polish)

Jarosław Pałka

Presentation in Polish.

This time we will take a deep dive into vast ocean of NOSQL and will sail for island of Web development, carefully watching for Maelstroms of Reliability, underwater Volcanoes of Stability and Reefs of Scalability.

During presentation I will show refactoring of simple shopping site, written with typical technology stack Wicket/Spring/Hibernate, and will replace some parts of persistence layer with NOSQL databases. To make my presentation more challenging for me and more fun for participants I am going to use 3 different NOSQL approaches to refactor 3 different parts of web site functionality. This way I would like to show where specific NOSQL models fit in your website, where SQL still rocks and what architecture, design and infrastructure problems can be solved with NOSQL. I will also show that NOSQL databases are not "silver bullet", "holy grail" or "swiss army knife" and they have really limited but well defined functionality. The magic ingredients of our trip will include key value stores, graph and document database with a pinch of endless complaining on enterprise architecture.

At the end of presentation we will end up with with web site that connects to 4 different data stores and we will still feel good about it as all architecture choices will be driven by customers needs and strong instinct to "keep it simple".

I hope that after presentation you will feel more confident with NOSQL and see that there is no big paradigm shift this time and NOSQL is something we were thinking about for many years but were to afraid to say it loud.

HTML 5 Fact and Fiction

Nathaniel Schutta

For the last few years, the web has been all a-twitter about web 2.0 (and even the occasional reference to web 3.0.) Yes, the days of static web applications are officially over and while libraries like jQuery and Prototype make it easier to build modern applications, ultimately they are papering over issues in the web standards (and the browsers that implement them.) Today we're building to standards that are from the paleolithic era of web design but that's changing - and HTML 5 is a large part of that. In this talk, we'll discuss just what HTML 5 is and why it matters. We'll show how you can build to HTML 5 today and which browsers support what. Thankfully, after many years of stagnation, the future of web applications looks bright!

Functional Thinking

Neal Ford

Learning the *syntax* of a new language is easy, but learning to *think* under a different paradigm is hard. This session helps you transition from a Java writing imperative programmer to a functional programmer, using Java, Clojure and Scala for examples. This session takes common topics from imperative languages and looks at alternative ways of solving those problems in functional languages. As a Java developer, you know how to achieve code-reuse via mechanisms like inheritance and polymorphism. Code reuse is possible in functional languages as well, using high-order functions, composition, and multi-methods. I take a variety of common practices in OOP languages and show the corresponding mechanisms in functional languages. Expect your mind to be bent, but you'll leave with a much better understanding of both the syntax and semantics of functional languages.

Hadoop: Divide and Conquer Gigantic Datasets

Matthew McCullough

Moore's law has finally hit the wall and CPU speeds have actually decreased in the last few years. The industry is reacting with hardware with an ever-growing number of cores and software that can leverage "grids" of distributed, often commodity, computing resources. But how is a traditional Java developer supposed to easily take advantage of this revolution? The answer is the Apache Hadoop family of projects. Hadoop is a suite of Open Source APIs at the forefront of this grid computing revolution and is considered the absolute gold standard for the divide-and-conquer model of distributed problem crunching. The well-travelled Apache Hadoop framework is currently being leveraged in production by prominent names such as Yahoo, IBM, Amazon, Adobe, AOL, Facebook and Hulu just to name a few.

In this session, you'll start by learning the vocabulary unique to the distributed computing space. Next, we'll discover how to shape a problem and processing to fit the Hadoop MapReduce framework. We'll then examine the incredible auto-replicating, redundant and self-healing HDFS filesystem. Finally, we'll fire up several Hadoop nodes and watch our calculation process get devoured live by our Hadoop cluster. At this talk's conclusion, you'll understand the suite of Hadoop tools and where each one fits in the aim of conquering large data sets.

Failures Come in Flavors

Michael Nygard

The typical JEE application does not reach the fabled "five nines" of availability. Far from it. It's more like "double eights". Come see why enterprise applications and web sites are only serving users 88% of the time instead of 99.999%.

The bad news: applications are more complex and error-prone than ever. Site development projects are really enterprise application integration projects in disguise. SOA portends far-flung interdependencies among unreliable services. Failures will spread wider and wider, reaching across your company and even crossing boundaries between companies.

How do monumentally costly failures begin, develop, and spread?

Can they be averted?

Once you hit Release 1.0, your system will be living in the real world. It has to survive everything the messy, noisy real world can throw at it: from flash mobs to Slashdot. Once the public starts beating on your system, it has to survive--without you.

Did you know that just having your database behind a firewall can bring down your system? I'll show you that and many other risks to your system. You will learn the biggest risks to your system and how to counter them with stability design patterns. We'll talk about the best way to define the term "availability" and why the textbooks get it all wrong.

In this session, you will learn why the path to success begins with a failure-oriented mindset. I'll talk about numerous antipatterns that have caused and accelerated millions of dollars worth of system failures. I'll share some of my scars and war stories with you (don't worry, they're all suitable for polite company) in the hopes that you can avoid some of these costly disasters.

Architect for Scale

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.

In this talk, Michael will discuss the notion of "design envelopes" for architectures. He will explore several common scaling strategies and map them to different system scales.

During this session, Michael will present reference architectures for systems at a variety of scales. It's sometimes possible to scale smoothly from range to range, but it can be extremely disruptive if you don't plan for it.

Along the way, he'll also address the different dimensions of scalability problems: traffic, content, data volume, and operations.

Command-query Responsibility Segregation - nowe, bardziej racjonalne podejście do warstw. (Polish)

Sławomir Sobótka

Presentation in Polish.

Z biegiem kariery, tworząc kolejny system lub aplikację, stajemy przed problemami innej klasy. Inna klasa problemu to inne wymagania odnośnie technologii, metodyki i architektury. Popularne architektury warstwowe mogą być nieodpowiednie do pewnych klas problemów - mogą np. narzucać nieracjonalne wykorzystanie ORM lub promować naiwne modelowanie domeny biznesowej.

Podczas prezentacji przedstawię nowe ujęcie warstw - architekturę Command-query Responsibility Segregation, która wspiera aplikacje średniej i dużej skali. Architektura CqRS promuje zdobycze nowoczesnej inżynierii oprogramowania takie jak: Domain Driven Design, otwiera system na skalowanie oraz w naturalny sposób rozwiązuje typowe problemy z mapowaniem relacyjno-objektowym.

Tłem technicznym dla przykładów będzie Spring i JPA.

Rich Grails UIs with Canoo RIA Suite

Dierk König

Learn how to easily write a Grails web application that appears to your users as if it was a desktop application. To this end, we will use the Grails ULC plugin and the UlcBuilder in a live programming experience.

Effective Java

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. In this presentation we will look at various concepts that you use in general programming with Java. We will discuss the issues with those and how you can improve your code. We will look at concepts you can do better and those you should outright avoid.

Patterns and Anti-Patterns in Hibernate

Patrycja Węgrzynowicz

Hibernate is the most popular object-relational mapping library for Java, providing a framework for mapping an object-oriented domain model to a traditional relational database. An application of hibernate to simple models is straightforward and almost effortless. However, in case of more complex models we usually run into some issues related to performance or correctness.

We will show several flaws in the demo from “Java Persistence with Hibernate” - CaveatEmptor (yes, it has several bugs including a serious locking-related issue!) and other open-source projects, which illustrate transactional and performance anti-patterns in hibernate.

The hibernate-related flaws will be accompanied by alternative solutions and best practices, which help to improve performance and quality of both database and object-oriented models. We will explore patterns and practices mainly in the context of object-oriented model, specifically how to meet object-oriented principles, yet to ensure correctness and efficiency of hibernate mappings.

Additionally, we will present Yonita, a static analysis tool that helps in automated discovery of concurrency-related issues with hibernate and database transactions.

Upon completion of this presentation, attendees should better understand the potential hibernate transactional and performance issues along with patterns to use hibernate in a correct and elegant way. Moreover, attendees will learn how to automatically discover a certain class of hibernate-related bugs.

The talk is a revised version of the talks presented at Jazoon 2010, Strange Loop 2010, and TheServerSide Java Symposium 2011.

Hacking Your Brain for Fun and Profit

Nathaniel Schutta

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

EJB 3.1 vs Contexts and Dependency Injection (CDI) and Dependency Injection for Java in Java EE 6

Jacek Laskowski

There seem to be some overlap between EJB 3.1, CDI and DI. They all are part of Java EE 6 and as far as dependency injection - the design concept - is concerned they are indeed the same. The question you may be tempted to ask yourself is why they all became part of Java EE 6 and what are the main benefits of using one over the other. It may come as a surprise that there's not much difference, and during the presentation, we'll strive to find them by a means of extensive set of examples.

Play! framework: a revolution in the Java world

Nicolas Leroux

The Play! framework is more than just yet another web framework in the Java eco-system: it is also another way of developing web applications. Like RoR, Django, it encourages fast and expressive development, with a clean and pragmatic API. Most importantly, perhaps, it brings simplicity to the Java world. It is the first web application framework in Java that is built by web developers for web developers. Because it is a simple, stateless framework that promotes a RESTful approach it offers the possibility of building complex web applications and, by extension, complex systems.

So you think agile software development and building houses have nothing in common... (Polish)

Wojciech Seliga

We are often told: building software is completely different than building houses. Rules applicable there do not work here and vice versa.

Really?

I am about to finish a house. Built from scratch. That's my single biggest personal project I have ever managed. Two years, around one hundred of people involved, strict budget and deadline, a lot of constraints and even stakeholders (bank...). All risk was mine. Also mine was my agile experience.

This presentation shows how agile principles work when applied in a completely different world - world of concrete, bricks, wood, hardware, plumbers, sand. World having as little in common with software as possible. You will hear maybe obvious statements, but you will learn things quite unexpected. You will hear which lessons, we, software guys, may learn from such experience and from such people - theoretically living in a different world - but facing surprisingly similar problems.

If you are a project manager or a developer interested in end-to-end view on a software project, if you are not afraid of crazy metaphors or comparisons and if you are interested in viewing your daily job at a little bit different angle - this session is for you.

Java Boilerplate Busters

Hamlet D'Arcy

Java has a reputation for boilerplate code: ubiquitous getters and setters, a verbose anonymous class syntax, and redundant declarations to name a few. It doesn't have to be this way! There are many ways to bust the boilerplate and this session provides a solid understanding of the most modern techniques. Come learn about inversion of control idioms, Proxy objects, code generation tools, and the latest libraries that both create and manage boilerplate code so you don't have to. A leaner, meaner codebase is yours for the taking.

Comparing JVM Web Frameworks

Matt Raible

The rise of Java Web Frameworks has come and gone, yet many are still widely used today. This session looks at the top JVM web frameworks and compares their pros and cons. Frameworks included: Struts 2, Spring MVC, Wicket, JSF, Tapestry, GWT, Grails and Rails

Don't code - create software!

Paweł Lipiński

Many programmers just focus on writing code. They learn new languages, use bleeding-edge technologies, or check performance of for vs. while loops. Not that it's not important, but many times during a project they forget that the ultimate goal is to create software that satisfies the customer, and they completely neglect things like maintainability, alignment with requirements, or more generally just quality. During the presentation I'll be talking about using Behaviour-Driven Development, Test-Driven Development and Domain-Driven Design to produce high-quality software. I'll briefly tell you what they are, how they complement each other and show you how they can help you with your projects.

JPA beyond essentials

Paweł Szulc

“Requirements: Good knowledge of JPA/Hibernate” - does it seem familiar? Well, it should, since this sentence happens to be on almost every Java related job description. Majority of applications (especially web applications) use database as back-end and those databases are accessed using ORM technologies. Therefore it seems pretty obvious that developers should have rather deep knowledge of ORMs they use. This is however not always the case.

The biggest advantage of JPA is also its biggest flaw - simplicity. Learning curve is really, really short. You quickly learn how to map your entities with tables from the database, persist/merge/delete operations work out of the box. It takes maybe a few hours to grasp the cascade concept, but after that you are ready to go... or so it seems.

Does `LazyInitializationException` ring any bell? Have you ever persisted your entity, but it never actually showed up in the actual database? Ever worked on an application that was running smoothly on development environment, but deployed to a production was facing major performance issues?

Are any of these familiar? Do you constantly fear that one day they might strike right at you, causing major damage on the most vulnerable part of your code? In this talk I will help you finally stand up for yourself and face those issues once and for all. We will enter a vast, dangerous world of JPA and travel way beyond simple entity mappings. I will guide through fundamentals of advanced topics of JPA programming and even though this will be only a short trip, before this talk finishes, you will have a better understanding of your favorite ORM standard.

Are you ready to dive in?

Do you really get Memory?

Jevgeni Kabanov

That's a stupid question, isn't it? Memory is just where you keep things. Nevertheless the majority of performance problems I have seen were in the end connected to memory issues. In modern computers memory is a complex multi-layered thing affected by the hardware, operating system and the JVM. And considering that it's shared among multiple CPUs we get a lot of classical distributed problems without an actual network anywhere in sight.

You should attend this talk if any of the following questions give you pause:

- How slow is memory?
- What does volatile really do?
- How pagefaults happen and what do they mean?
- Why swapfile and Java don't work together well?
- Why GC doesn't scale well (so far)?

- What good alternatives to Java Memory Model are there?

To answer those questions I'll build a simple CPU model in Java to illustrate some of the concepts and do live simulations for the rest. As a bonus I'll also talk about some improvements in the modern hardware and what it could mean for the next generation JVM.

Lost Chapters of Divine Code: 7 Deadly Sins

Szczepan Faber, Bartosz Bańkowski

A proper software reverends would preach on Cardinal Virtues, confession and atonement. We are inquisitors and we focus on Deadly Sins and the penance. However, if you show the true understanding we may be merciful. If you realize that privacy is godless and you promise to behave then maybe generous amounts of Holy Water and Incense is enough. If you promise that static is evil then perhaps a mild Exorcism followed by an eager Prayer will do.

You will get the chance to realize other sins as well. Join us and be reborn as a Pious Programmer.

REST in Java

Stefan Tilkov

REST, the architectural style of the Web, is continuously gaining popularity, whether it's external Web APIs or internal integration scenarios. This talk will show the different options a Java developer has for building both RESTful client and server applications, including the servlet API, Restlet, JAX-RS and its various extensions as well as the Spring framework. We'll also address some of the doubts regarding the ease with which both human-facing Web apps and Web Services can be developed, and show how RESTful HTTP not only leads to a better overall architecture, but also to increased developer productivity.

Practical Guide to using REST for SOA

Stefan Tilkov

Whether you've already managed to convince your higher-ups that REST is the way to go, can decide to do so for yourself, or are still looking for ways to do the convincing, at one point in time you'll have to actually start doing stuff as opposed to talk about it. In this talk, we'll take a look at some practical recipes on how to use RESTful approaches for SOA, including options for mixing it with WS-*, picking the right data formats, (re-)introducing infrastructure, dealing with RESTful SOA governance and other aspects deemed "enterprisey" by the cool kids.

Rethinking "Enterprise"

Ted Neward

The era of the big, heavy, transactionally-oriented client/server topology, as best described by the J2EE Specification and Blueprints document, seems to be over. The era of the lightweight,

transactionally-oriented client/server topology seems to be at its zenith. Is this really where we want to be when building enterprise systems? And how did we end up here, anyway? What's the new "enterprise" developer supposed to do?

CDI extensions: Backport Java EE from the future

Dan Allen

Java EE 6 offers significant and compelling improvements over previous versions of the platform. But regardless of how sweeping the change, it's not long before we want more. In this talk, you'll discover that you don't have to wait for Java EE 7 to get the features you crave thanks to the portable extension integration SPI introduced into the platform by JSR-299.

This talk introduces JSR-299: Contexts and Dependency Injection for the Java EE platform (CDI), the new Java standard for dependency injection and contextual lifecycle management. The talk covers the core programming model, explains its relationship to EJB 3.1 and JSF 2.0, and clarifies how it unifies and enhances the Java EE platform as a whole (extending to JPA, JAX-RS and JMS). You are then introduced to Weld, the Apache-licensed JSR-299 reference implementation, OpenWebBeans--another Apache-licensed implementation--and Resin, along with their servlet container add-ons.

Most important, we look ahead at portable enhancements provided by CDI extensions that help extension writers today and will become candidates for the next spec revision. To prove that the Java EE platform is truly extensible, we'll focus on examples rather than promises. We'll go over the options you have for enhancing the application, from registering custom beans, interceptors and decorators, to customizing dependency injection points, to augmenting the annotation-metadata on registered beans, to introducing custom scopes. The talk will cover when and how you weave these enhancements into the container lifecycle using the CDI eventing system. Having looked at some standard examples, we'll unlock some shortcuts and shorthands that are provided by the portable Weld Extension library. Examples we'll be taken from the growing ecosystem of open-source portable extensions.

While many of the features provided by CDI (dependency injection, contextual lifecycle, configuration, interception, event notification) are familiar, the innovative use of meta-annotations is uniquely expressive and typesafe. This talk emphasizes the value in this approach. This is a great opportunity to learn about the this new, extensible programming model that forms the foundation of a unified, extensible Java EE 6 platform.

Arquillian: Real Java enterprise testing

Dan Allen

This talk unveils the missing link in enterprise Java development: simple, portable integration tests. For many, this void has long made enterprise Java development an arduous undertaking.

While development life is simple with unit tests and mocks, they only take you so far. Eventually, you need to validate how your components interact and operate in their intended environment-- you need real integration tests. Yet, writing integration tests has meant assuming the burden of bootstrapping all or part of your infrastructure. That's time lost and it places a mental barrier on testing.

Arquillian, a container-oriented testing framework layered atop TestNG and JUnit, tears down this barrier. It does so by bringing your test to the runtime rather than requiring you to manage the runtime from your test. Picking up where unit tests leave off, Arquillian enables you to test real components that rely on real enterprise services in a real runtime. You'll see tests that run on JBoss AS, Weld Embedded, GlassFish and more. Attend this talk to learn about the future of Java enterprise testing.

Busy Developer's Guide to Scala: Patterns

Ted Neward

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files. Sporting the usual object-oriented concepts as classes and inheritance, Scala also offers a number of powerful functional features, such as algebraic data types, immutable objects by default, pattern matching, closures, anonymous functions and currying, and more. Combined with some deep support for XML generation and consumption, Scala offers Java programmers an opportunity to write powerful programs with concise syntax for a new decade of Java programming. In this presentation, we'll start looking at some of the O-O community's favorite design patterns (from the Gang-of-Four book as well as others) and examine their applicability to Scala, including those patterns that work well here, those that don't, and a few new ones that wouldn't work well anywhere but here.

Using Spring with non relational databases

Costin Leau

The needs of many enterprises have stretched traditional RDBMS based solutions to the breaking point and as a result, a plethora of new non-relational storage options have appeared: key-value stores, column storages, document and graph databases. This talk will show how the Spring Framework is evolving to support 'NoSQL' databases by bringing traditional Spring values such as portability, productivity and a unified POJO based programming model to this style of data access. During the presentation, the main features of Spring Data will be discussed by taking a tour of the various offerings out there such as Redis, Mongo and Neo4J just to name a few. Basic NoSQL knowledge is required.

Eclipse Orion: Introduction

Szymon Brandys, Tomasz Żarna

Orion is a browser-based open tool integration platform which is entirely focused on developing for the web, in the web. The project is currently in the early stages of development, with a code editor, file and folder navigation and Git integration already created, and a debugger in the works.

This talk will demo the current state of Orion, and explain the vision of moving software development to the web. It will also explain the design based on well-proven web architectures - using hyperlinks to navigate between different tools and resources, making data accessible in a RESTful way, and using web technologies like HTTP, JSON, OAuth, OpenID and others.

See wiki.eclipse.org/Orion for more information about Orion.

Programming Clojure

Venkat Subramaniam

Clojure is an exciting language on the JVM for several reasons. It brings the power of functional programming and the characteristics of dynamic typing into a single fold. It is a language that has done quite a few things right. It is lightweight, highly expressive, concise, and offers quite an interesting, simple, yet very powerful solution to handle concurrency. In this session you will learn the key strengths of this language and how you can integrate it with your Java projects.

Fractal TDD: Using tests to drive system design

Steve Freeman

We present our experience applying Test-Driven Development (TDD) at all levels of the development process. TDD at the class level is now well understood (if not always well practiced). We find that the benefits we get from writing tests first and using them to drive design apply at the system level too. That's why we try to address integration and system testing as early as possible. The sooner the system is in a deployable state, the better equipped we are to react to changing business needs by delivering new features. Our experience is that putting testing at the front of our development process, and paying attention to what we learn from them, flushes out architectural issues such as concurrency and distribution. The result is systems that are easier both to maintain and to support. We can also avoid some of common testing pitfalls, such as unreliability, slow execution, and brittleness.

Five years of change, no outages

Steve Freeman

Our team built a static data system for a leading investment bank. It went into production about 5 years ago, since when it has grown significantly and been through two generations of developer. It has never failed in production. We followed XP as an approach with a strong emphasis on driving features with acceptance tests.

Alexis Moussine-Pouchkine

GlassFish OSGi - from modular runtime to hybrid applications

You don't need to choose between Java EE and OSGi, you can use both in a number of ways to address your modularity and enterprise needs. Here's how.

GlassFish v3 was built with modularity in mind using 200+ OSGi bundles running on Apache Felix (the default), Equinox and other runtimes. It can be installed on existing OSGi, explicitly deploy OSGi bundles, re-use existing OSGi tooling yet without ever exposing this modules technology to the Java EE developer. All this offers great modularity which we will discuss and demonstrate in this session.

While hiding OSGi as an underlying implementation detail offers GlassFish developers and users many valuable features, you may also be interested in tapping the OSGi nature of GlassFish in the following ways:

- extend GlassFish using OSGi Declarative Services
- expose JPA, WebServices, or EJB's as OSGi services
- use Web Application Bundles (WAB)
- CDI-injected dynamic OSGi services

This session will cover these different approaches of using OSGi and discuss their relative merits.

Hamlet D'Arcy

Get 'em before they get you

Let's face it - the code we write contains bugs. These bugs impact stability, reliability, correctness, performance or maintainability of our applications as well as our own reputation as developers. I'll show you a few handy tools that help a lot in finding and eliminating bugs, tools that enable you to start fixing reported exceptions in seconds, tools that highlight repetitive pieces of code, analyze flow of values through your program, visualize dependencies in code and highlight possible dependency problems. During the presentation I'll show you an alternative view on IntelliJ IDEA, a Java IDE with more than a 10 year history in the field. You'll get a chance to see lots of hands-on practical demos revealing the analysis aspect of the IDE and gain more insight into how you can fight for greater code quality in your projects, today.

Martijn Dashorst

Introducing Apache Wicket

Apache Wicket is a component oriented Java web framework. With proper mark-up/logic separation, a POJO data model, and a refreshing lack of XML, Wicket makes developing web-apps simple and enjoyable again.

This talk provides a short introduction to the framework's concepts: components, behaviors and models. We'll take a short look at integrating with Spring and integrating with JQuery. For the test infected we'll cover testing your web pages with Wicket's internal test framework. But most importantly we'll take a look at the new Wicket release 1.5 and see what has changed for the better and worse.

Where is my table now? On adapting Java and SQL code to ever-changing database model.

Tomasz Skutnik

Changing requirements are a fact of life. No matter how small or big our IT system might be, sooner or later it will need to change. A new business requirement, a new tax rule, a new integration point --- anything may force us to change our application in nontrivial way.

Nearly all Java web applications use relational databases. We store a lot of important business information in them. We trust them that our data won't get lost or be corrupted. We expect them to be high performing and reliable. And in fact they do a pretty good job of it.

However, when we need to modify the system functionality they also pose the biggest challenge for us. Single change to a database schema may involve multiple changes throughout the whole application. Java code and SQL statements need to be updated and keeping all this information in sync is difficult and error prone.

This talk surveys the common issues that may arise during database schema updates, their impact on application code and how can we prepare for and manage those changes. It covers these topics:

- How do I keep track of changes in my database?
- What impact on my code will have this particular change?
- How can I keep my SQL synchronized with database schema?
- How can I detect inconsistencies as early as possible?

If you have ever had a hard time fixing bugs that blown up after database upgrade --- this talk is for you.

HTML5 WebSockets - the Web Communication revolution, making the impossible, possible

Brad Drysdale

HTML5 WebSockets is rapidly becoming the standard for better connecting business data with Web-based clients, and also for Web and Cloud enabling parts of the business that perhaps today are not taking advantage of these trends. However the vision for WebSockets is vastly bigger than this. What today constitutes the best choice for delivering low-latency, real time data to Rich

Internet Applications is now also proving to be the new standard communication technology for all Web interaction going forward. With the limitations of HTTP lifted, what sorts of applications can now be built? What new ways can you best connect your business with your customers over the Web, Cloud or Mobile with low latency and high scale? Come to this session to hear about how WebSocket technology is delivering on the dreams of many large organisations as they better serve their customers demands for business data over the Web, and hear about some of the revolutionary things that were once deemed impossible, made possible.

Going Mobile with jQuery

Nathaniel Schutta

The word just came down from the VP - you need a mobile app and you need it yesterday. It needs to be polished and have that design stuff too. Oh and it needs to be on all the major platforms in time for the big marketing push next month. After a moment of panic, you wonder if it's too late to become a plumber but don't worry, there's hope! More and more developers are falling in love with the "write less do more" library and for good reason; it simplifies the job of today's front end engineer. But did you know jQuery could also help you with your mobile needs as well? That's right, jQuery Mobile is a touch optimized framework designed to provide a common look and feel across a wide variety of today's most popular platforms. In this session, we'll take a look at all that jQuery Mobile has to offer and we'll convert a native application to an HTML5, jQuery Mobile masterpiece.

In this session, we'll take a look at:

- pages
- toolbars
- buttons
- form elements
- list views

How to solve unsolvable problems in projects

Marcin Kokott, Martin Chmelar

These days everyone is looking for an excellent methodology to help us bring the project to the desired success. But the history of IT has taught us that there is no ideal approach, but only the ones that make the project easier to manage by rapid visualization of the coming problems. Great leader, stands out from the rest not only with the fact that they are able to notice a problems, but, among other things, whether and how well they are able to solve them with their teams. The fact is, that we are investing a lot of time into improving the techniques of project management, but are we really able with this knowledge to deal with solving problems in an appropriate manner? Isn't that too often familiar to us, that we find a workaround just because the problem seems to be unsolvable? How often in one moment, with one decision we can decide whether the

project will end up with another success or maybe a complete failure? Join us to learn how to be a better leader in the IT world... and your daily life.

“Same Data, Any API”, making sure your application scales

Guy Korland

A “normal” Java application uses many different data APIs to answer different requirements - for example, JPA for data access, JMS for messaging, Memcached for caching and REST for interoperability. Each API has its own flavor, limitations, implementation, transactional model and scaling model. With the same data flowing back and forth between these APIs, development is problematic and difficult.

XAP’s “Same Data, Any API” capability promotes openness and interoperability, supporting all common interfaces for accessing data, such as Memcached, JCache, JDBC, JPA, JMS, REST, Document, and the highly efficient POJO API. XAP ensures you can choose the best API for the use case at hand and operate on the same underlying data regardless of which API you use. This, in turn, significantly decreases the integration process, eliminates the data interoperability torture and promises a uniform transactional & scaling model.

Monitoring 10 Critical Code Quality Metrics with Sonar

Matthew McCullough

You’re serious about improving the quality of your code base, but with 10,000 lines of code, where do you start and how do you ensure the greatest ROI for the re-work your team members will perform?

Sonar is an open source tool that brings together the best of breed static and dynamic analysis of Java projects. The result is a unified view of problematic areas of your code on a time-line basis, allowing the team to attack the problems with the best ROI, and maintain a more watchful eye for positive and risky trends in the codebase in the future.

This talk will show you Sonar from the ground up and explain 10 critical metrics that affect your code’s flexibility, stability, and durability.

Workshop: The Productive Programmer

Neal Ford

The Productive Programmer consists of two parts: mechanics & practice. In the mechanics section, I identify 4 principles of productivity: acceleration, focus, automation, and canonicity. This session defines the principles and describes their use, but the primary focus of this workshop is real-world examples of how you can use these principles to make yourself a more productive programmer. The second part of this workshop covers 10 ways to improve your code, derived from

the practices section. This workshop includes tons of examples, all culled from real-world projects.

Workshop: The Power of Retrospection

Linda Rising

Project Retrospectives are an important part of any software development process. The Principles Behind the Agile Manifesto state that, "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly." How can this be done? By taking the time to reflect and learn and proactively determine what should be done differently in the next iteration, release, or project. Linda's presentation will introduce techniques for project retrospectives, whether they are agile or not. The techniques help teams discover what they're doing well so that successful practices can continue and identify what should be done differently to improve performance. Retrospectives are not finger pointing or blaming sessions, but rather a highly effective process in which teams reflect on the past to become more productive in the future. Linda will share her experiences with leading retrospectives of several kinds for dozens of projects—successful and unsuccessful, small and large, in academia and industry. Her lessons learned can be applied to any project to enable teams and organizations to become learning organizations.

Workshop: Android programming

Sang Shin

Android is becoming a popular platform for building mobile applications. The workshop is made of two parts. In the first part of the workshop, you are going to build a few simple Android applications. The goal of this part is to get you exposed to all the steps involved in building Android applications. In the second part, you are going to explore two key concepts of Android programming - "Activity" and "Intent".

The attendees are expected to come with their laptops with Android development tools pre-installed as described

in http://www.javapassion.com/tmp/android_stepbystep/#Exercise_0 Attendees are also expected to download the hands-on lab zip files that are available from the "Android programming" course website - Go to <http://www.javapassion.com> and then click "Android programming". And you will find 5 hands-on lab zip files. There is no need to register to JavaPassion.com in order to get the hands-on lab zip files.

Workshop: Introduction to Scala

Hubert Plociniczak

In this course we will give you a gentle introduction to the Scala language. Scala is a statically typed language developed in order to increase the productivity of the programmers as well as

coherence and maintainability of the code. Since it runs on a standard JVM, it can easily be integrated with existing Java projects, hence enabling gradual migration.

Due to its extensibility and support of a complex type system Scala enables the programmers to focus on the process of solving problems rather than thinking on how they fit into the frames of the language. In other words the language helps to reduce the amount boilerplate code we would normally have to write. In this presentation we will describe the fundamentals that already enable you to write more elegant code.

Some parts of the language that the workshop will cover include:

- Objects, classes, traits
- Variables, method definitions, expressions
- Functional style of Scala including closures, higher-order functions, immutability, call-by-name and currying
- Pattern-matching
- Scala Collections
- For-comprehensions

Each of the concepts will be introduced through a number of motivating examples in the form of code snippets, ensuring both a theoretical and practical presentation.

No previous knowledge of Scala is needed to follow the workshop. Scala examples will often be confronted with Java equivalents in order to ease the transition between the languages.

Workshop: Git

Luca Milanesio

GIT is the leading edge distributed SCM written by Linus Torvalds and born from the Linux Kernel development. It is now widely used in many OpenSource projects and it is starting replacing SVN in some Enterprises. Distributed SCM is quite different from traditional Server-centric version controls and GIT introduces a full set of new concepts and mechanisms. Explore GIT from his roots and learn how to use it effectively in your open-source projects and in your enterprise, with simple examples, hands-on in a very interactive workshop, from developer to developer.

Full agenda is available [here](#).

BOF: Automated Bug Hunting

Patrycja Węgrzynowicz

Ever since the invention of computers, software has been all about automation and more efficient execution of manual tasks. Today, software itself is becoming more and more complex. There is a growing need for automation in the area of software development, particularly software quality.

In this talk we will present various techniques that support automated detection of bugs in software. We will focus on static and dynamic code analysis as well as automated test generation. We will discuss strengths and weaknesses of each technique along with their applicability and the classes of detectable bugs. Finally, we will picture the IDE of the future that combines different techniques to support developers in automated bug hunting.

BOF: Dokąd zmierza Software Craftmanship (Polish)

Sławomir Sobótka, SSEPP members

Presentation in Polish.

Ruch Software Craftmanship jest odpowiedzią na pojawiające się potrzeby problemy.

Jest to jednak ogólna idea. Warto jednak zastanowić się nad konkretnymi:

- jakimi cechami powinien charakteryzować się profesjonalista
- jaki zakres bazowych kompetencji powinien posiadać profesjonalista
- ew. jakie poziomy kompetencji możemy próbować określać
- być może zależy to od klasy problemów
- czy nadszedł już czas na krystalizację profesji - czy mamy właściwą ilość doświadczeń
- czy i kiedy warto mówić o profesjonalizmie - kiedy jest on potrzebny i kiedy wnosi wartość

BOF: Agile Contracts in Practice

Wojciech Seliga

Agile development is hot. But how do we execute whole contract in agile manner: iteratively, without fixed scope, allowing changing requirements? Developing something in an agile fashion has limited sense if whole business is framed in a traditional fixed price / fixed scope way.

This BoF is to share experience and ideas about:

- how to convince the customer to go for an agile contract - main selling points
- how to address main concerns around agile contracting
- how to communicate and iteratively deliver software to the customer
- how to charge for such contract
- how to structure the contract document itself

This session is a continuation of an incredibly popular session at Javarsovia 2010 - we hope to learn together best practices and pitfalls to avoid.

BOF: Future of Java EE

Alexis Moussine-Pouchkine

By many accounts Java EE 6 has been judged as a very successful release adding RESTful web services, unified injection, vastly improving existing specifications to the extent that one could think there isn't really a need for Java EE 7. This BOF will introduce the main theme for Java EE 7 as well as the current list of features and discuss the plans and status of this major new release now under Oracle stewardship.

BOF: Hack your company: what a developer can do, to get fun and money from learning

Jakub Nabrdalik

Feeling bored? Tired? Burned out? Not everybody works for Google, not every company gets its income from advertisements. For most software houses, the work of a developer brings the money in, and so the pressure on efficiency and deadlines, makes it hard to learn and innovate. Chances are, after work you are already so tired, you don't want to do any programming anymore.

I'll show you a few tricks a developer can do to change this situation, in spite of pointy haired bosses and fossilized management. I'll tell you what works for my company, what does not, why, and what we are still too afraid to try out.

I'll show you the Agile Skills Project, how to build your motivation yourself and how to get fun and money from learning. And hopefully, I'll get you to share your experience and success stories, even though you may not be aware of them yet.

And yes, I know it'll be the 10th hour of the conference, so I'll make it easy on you. The goal is to have you find and remember at least one thing, that you can use right away and change your life for better.

BOF: Web frameworks shootout

Błażej Bucko, Tomasz Dziurko, Wojciech Erbetowski, Łukasz Kuczera, Paweł Szulc

What web framework should I choose?

Do you want an answer for this question? Come and join us to see 4 emerging web frameworks in action: Play, Wicket, Lift and Grails.

This BOF will contain 2 exercises for each framework:

- Obligatory: Create simple web form
- Free style: Present the best of the web framework

BOF: Those broken, broken class loaders

Jevgeni Kabanov

What web framework should I choose?

Hackergarten

Hamlet D'Arcy, Dierk König

Hackergarten is an open source programming space. Our goal is to create something that others can use; whether it be working software, improved documentation, or better educational materials. Our intent is to end each session with a patch or similar contribution submitted to an open and public project. If you have energy to learn and code, then come spend a few hours making the community better. If you have your own open source project, then come and introduce others to your software and make some new friends. Hackergarten is self-organizing and drop-in based. Come and go as you please.

Drinks are provided by Canoo Engineering, AG.

Training: EJB 3.0, JPA, Spring, Hibernate, Java EE 6

Sang Shin

This is a rare opportunity to learn how to learn how to build enterprise applications using advanced Java enterprise technologies - EJB 3.0, JPA, Spring framework, Hibernate, and Java EE 6) - from a long-time industry expert, Sang Shin. The codecamp is designed with heavy emphasis on hands-on development. Please expect an intensive 3-day training.

As a bonus. Every participant will get 1 year free access to all trainings on JavaPassion.com.

When

11th - 13th April 2011

Where

Krakow, Poland

Price

799 PLN + VAT

Who Should Attend

Anybody who need to learn EJB 3.0, JPA, Spring framework, Hibernate, Java EE 6

Topics covered

- **Part 1: Java EE 5 & EJB 3.0**
- Introduction to Java EE 5
- EJB 3.0 overview

- **Part 2: JPA**

- JPA basics
- JPA mapping
- JPQL

- JPA advanced

- **Part 3: Hibernate**

- Hibernate step by step
- Hibernate basics
- Hibernate mapping
- Hibernate query language
- Hibernate transaction and concurrency
- Hibernate interceptors and events
- Hibernate join fetch
- Hibernate caching

- **Part 4: Spring framework**

- Spring framework basics
- Spring MVC
- Spring Webflow
- Spring AOP
- Spring Security

- **Part 5: Java EE 6**

- Servlet 3.0
- CDI
- JPA 2.0
- EJB 3.1
- JSF 2.0

What you should bring to the codecamp

Your own laptop

Pre-Requisites

- Some Java programming experience
- Some Web application programming experience