Trends affecting the Future of Enterprise Java

Floyd Marinescu

Enterprise Software Development Community

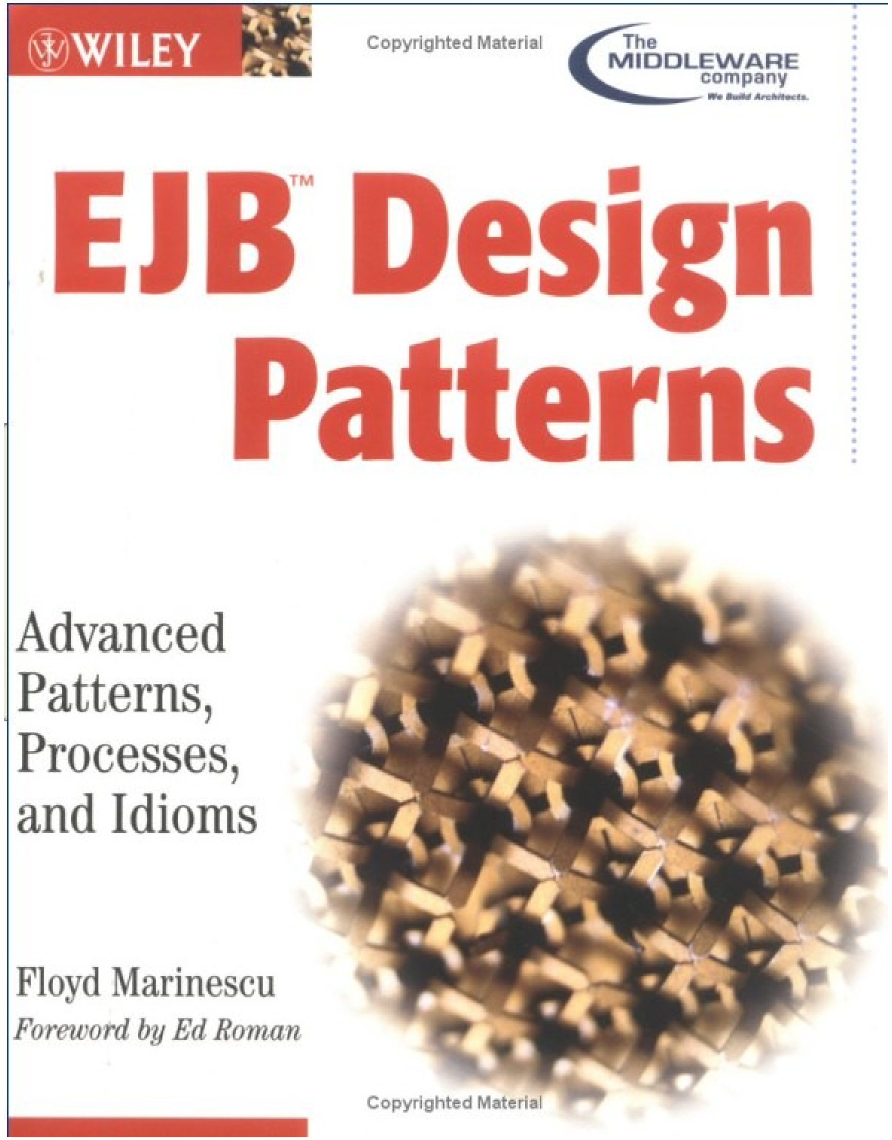
CEO & co-founder, InfoQ.com
Founder, TheServerSide.com
Author, EJB Design Patterns

Goals for this presentation

Bring you up to date on important news and trends, both technical and cultural, that are affecting the future of Java today and 2 years out

About Floyd Marinescu

- **Wrote the book *EJB Design Patterns (2002, Wiley)***
- **Co-founder, CEO InfoQ.com**
- **Designed, implemented, and managed TheServerSide.com J2EE Community for 6 years**
- **Sun appointed Java Champion**
- **I am** 🍁



WILEY

Copyrighted Material

The MIDDLEWARE company
We Build Architects.

EJB™ Design Patterns

Advanced Patterns, Processes, and Idioms

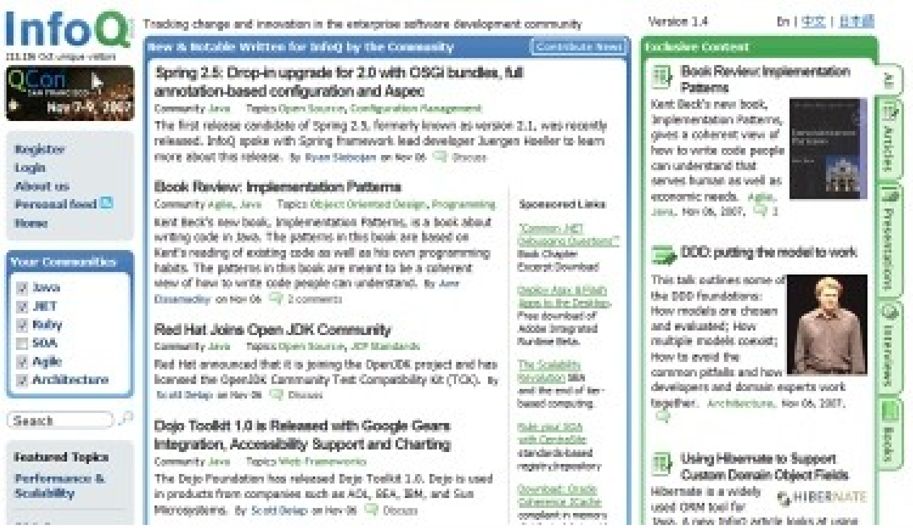
Floyd Marinescu

Foreword by Ed Roman

Copyrighted Material

About InfoQ.com

- Online News Site/Community for senior developers, Team Leads, Architects, Project Managers
- Mission: *Tracking Change and Innovation in Enterprise Software Development*
- 250,000 unique visitors in Jan 2008, launched June 2006
- Java, .NET, Ruby, SOA, Agile, Architecture
- Chinese & Japanese Editions
- News, Free Books, Articles, Video Presentations & Interviews
- Over 31 editors like you involved



InfoQ Tracking change and innovation in the enterprise software development community Version 1.4 En | 中文 | 日本語

QCon San Francisco Nov 7-9, 2007

Register
Login
About us
Personal feed
Home

Your Communities
☑ Java
☑ .NET
☑ Ruby
□ SOA
☑ Agile
☑ Architecture

Search

Featured Topics
Performance & Scalability

New & Notable Written for InfoQ by the Community — Contribute News

Spring 2.5: Drop-in upgrade for 2.0 with OSGi bundles, full annotation-based configuration and AspectJ
Community Java Topics Open Source, Configuration Management
The first release candidate of Spring 2.5, formerly known as version 2.1, was recently released. InfoQ spoke with Spring framework lead developer Juergen Hoeller to learn more about this release. By Ryan Slobojan on Nov 06 Discuss

Book Review: Implementation Patterns
Community Agile, Java Topics Object Oriented Design, Programming
Kent Beck's new book, Implementation Patterns, is a book about writing code in Java. The patterns in this book are based on Kent's reading of existing code as well as his own programming habits. The patterns in this book are meant to be a coherent view of how to write code people can understand. By Amr Elssamadisy on Nov 06 2 comments

Red Hat Joins Open JDK Community
Community Java Topics Open Source, JCP Standards
Red Hat announced that it is joining the OpenJDK project and has licensed the OpenJDK Community Test Compatibility Kit (TCK). By Scott Delap on Nov 06 Discuss

Dojo Toolkit 1.0 is Released with Google Gears Integration, Accessibility Support and Charting
Community Java Topics Web Frameworks
The Dojo Foundation has released Dojo Toolkit 1.0. Dojo is used in products from companies such as AOL, BEA, IBM, and Sun Microsystems. By Scott Delap on Nov 06 Discuss

Sponsored Links
"Common .NET Outsourcing Questions" Book Chapter Excerpt Download
Deploy Ajax & RIA Apps to the Desktop. Free download of Adobe Integrated Runtime Beta.
The Scalability Revolution: SBA and the end of tier-based computing
Build your SOA with CentraSite standards-based registry/repository
Download: Oracle Coherence 3Cache compliant in memory

Exclusive Content

Book Review: Implementation Patterns
Kent Beck's new book, Implementation Patterns, gives a coherent view of how to write code people can understand that serves human as well as economic needs. Agile, Java, Nov 06, 2007, 1

DDD: putting the model to work
This talk outlines some of the DDD foundations: How models are chosen and evaluated; How multiple models coexist; How to avoid the common pitfalls and how developers and domain experts work together. Architecture, Nov 06, 2007,

Using Hibernate to Support Custom Domain Object Fields
Hibernate is a widely used ORM tool for Java. A new InfoQ article looks at using

InfoQ 时刻关注企业软件开发领域的变化与创新 中文站1.4版 En | 中文 | 日本語

QCon San Francisco Nov 7-9, 2007

注册
登录
关于我们
个性化RSS
首页

我的社区
☑ Java
☑ Ruby
□ SOA
☑ .NET
☑ Agile
☑ Architecture

Search

InfoQ中文站10月份精彩内容回顾
社区 .NET, Architecture, Agile, Java, Ruby 主题 InfoQ中国
InfoQ中文站 发布于 11月7日 参与讨论

简讯: Visual Studio 2008将于本月底发布
社区 .NET 主题 .NET框架
作者 Jonathan Allen 发布于 11月5日 1 条评论

JSLoader提供了JavaScript库共享资源
社区 Java 主题 JavaScript
作者 Rob Thornton 发布于 11月5日 1 条评论

InfoQ中文站第44周精彩新闻回顾
社区 .NET, Architecture, Agile, Java, Ruby 主题 InfoQ中国
InfoQ中文站 发布于 11月5日 参与讨论

Java・.Net纵横

Joe Walker谈DWR

InfoQ Japanese Version Powered by CSQ 最新技術を追い求めるデベロッパのための情報コミュニティ Version 1.4 En | 中文 | 日本語

QCon San Francisco Nov 7-9, 2007

会員登録
ログイン
InfoQについて
パーソナルRSS
ホーム

コミュニティー欄
☑ Java
☑ .NET
☑ Ruby
☑ SOA
☑ Agile
☑ Architecture

Search

JBossがGroovyサポートとJSF強化を伴ってSeam 2.0をリリース
コミュニティ Java トピック Webフレームワーク
Scott Delap 翻訳者 投稿日 11月6日 ディスカッション

Adobe MAX Japan 2007: アドビが自社のオープンソース戦略を語る
コミュニティ Java トピック オープンソース, RIA
日本時間の11月1日から2日にかけて行われたAdobe MAX Japan 2007にて、アドビのSteven Heintz氏が自社のオープンソース戦略を説明した。 投稿日 11月6日 ディスカッション

JJUG Cross Community ConferenceにJames Gosling氏が登場
コミュニティ Ruby, Java トピック オープンソース, コミュニティ

デメテルの法則を守るRailsプラグイン
コミュニティ Ruby トピック Ruby on Rails

JSF/DWR/DOJOを使ってWebアプリケーションを作成する

Java・グリッド、なぜわれわれはそれを必要としているのか

OpenTerracottaの紹介

Catch-up: The return of OO Development in Java

- **J2EE development was not OO**
  - **1999-2004: Applications incorrectly designed around remoting (EJB)**
  - **Business logic moved into session beans instead of domain objects**
  - **Lack of ORM tools, persistent components over objects**
  - **Mis-marketing by Sun (EJB), young inexperienced developers**
  - **The rise of web-based development (new problem domain)**

…The return of OO Development in Java

- **What has changed**
  - **Dependency Injection, Aspects, Annotations**
  - **Component market place never happened**
  - **Lightweight POJO containers vs. appservers**
  - **Object relational mapping now free and standardized**
  - **Open source has fixed lacking issues in the specs**
  - **JAVA EE 5 now supported by most appserver vendors**
  - **The problem domain of web & server side Java development is now well understood**

Aspect Oriented Programming – Just beginning!

- **Refactor repeating (cross cutting) code into one place**
    - **Intercepting method calls, field calls, constructor calls**
    - **Adding interfaces & implementation to existing Java classes**
- **OO = nouns and verbs, AO is the adverbs and adjectives**
- **Look for adverbs and adjectives in your requirements**
    - **a SecureBankAccount**
    - **or a *Secure* BankAccount**

...AOP

- **Practical applications of AOP in use today:**
  - **Transactions**
  - **Security**
  - **Event handling (replace observer pattern)**
  - **Annotation Handling**
  - **Exception Handling / Mapping**
  - **Forcing a Singleton**
  - **Instrumentation / Metrics recording / Diagnostics**
  - **Testing your applications**
  - **Assertions**
  - **Architecture Enforcement**
  - **Adding state and behavior / multiple inheritance alternative**
  - **Logging**

Domain Annotations viable due to AOP?

Dependency Injection

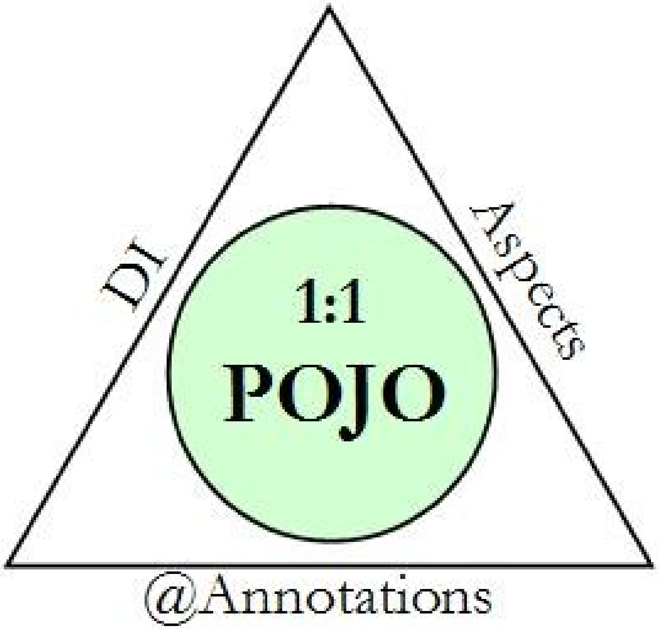
- **Refactor configuration and dependency code out of business logic**
- ***Inject* dependencies into objects using plain java setter or constructors**
  - **As opposed to explicit look up, or explicit call backs**
- **A simple POJO with a setter for injection:**

```
public class MovieListener {
   private MovieFinder _finder;
   public void setMovieFinder(MovieFinder finder) { _finder =
   finder; }
...
}
```

- **Already a mainstream trend – incorporated into the Java specs too!**

Foundations of todays application design

- **Community moving back to OO, POJO applications**
- **Dependency Injection, Aspects, and Annotations the foundations of such a modern application**
  – Adrian Colyer



Declining importance of Java EE standards

- **1998-2001: a fragmented and voluminous ecosystem drove a need for standards - J2EE creates a market**
- **2001-2002: real project experience drives backlash to the standards, need for more innovation**
- **2003+: Spring, Hibernate, light weight solutions. Key programming models open source + portable vs. standards based**
- **2000 – 2008: major appserver consolidation, now only TWO VENDORS**

Declining of importance of Java EE standards

- **Do standards matter when there are only 2 vendors?**
- **Is lock-in a factor when using open source?**
- **Community-driven emergent standards via open source vs. design by committee**
- **Portable runtimes vs. standard API’s**
  - **Spring vs. J2EE, Apache vs. HTTP, Flash vs. Ajax**
- **IBM and BEA increasingly distant from J2EE, JCP**
- **Rod Johnson: the JCP is like the communist party**
- **OSGi the final ‘nail in the coffin’?**
- **Mainstream projects not so concerned about Java standards anymore**

Open Source is now mainstream

- **Used to be driven by developers (bottom up)**
- **Now often being driven top down**
- **Execs: ‘Open source more reliable’**
- **Support and indemnity commonly available**
- **Death of the in-house framework**
- **Used to be psychology of standards use, now psychology of portable framework reuse**

Modularity: OSGi Service Platform

- A specification of a service platform / component model for Java (like an appserver)
- Defines ***bundles***(typically JAR files) that contain
  - Java classes, Resources, Files, Metadata
- Bundles are independent (eliminates inter-module dependency problems), self-describing, dynamically manageable (start/stop/install/uninstall)
- Ideal base for long running systems composed of ‘modules’
  - IDE, Appserver, etc
- An ideal micro-kernel architecture

OSGi ecosystem

- **OSGi Alliance**
  - **Specs for runtime environments, traditionally embedded**

Equinox

  - **Eclipse's OSGi implementation and OSGi R4 reference impl**
  - **One of many open source OSGi implementations**

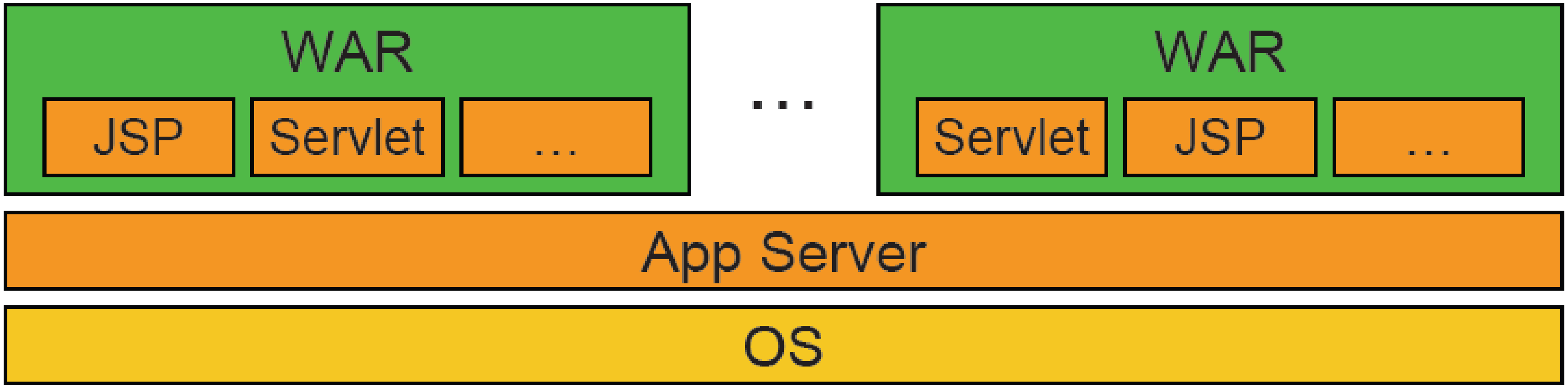
- **Eclipse – OSGi based for 5 years**
  - **Eclipse RCP – needed open, modular, dynamic, modular runtime**
- **OSGI officially part of Java SE**
  - **Sun developing competing JSR 277 – module system for Java**

OSGi – recent uses & stated benefits

- **Oracle, BEA, IBM, JoNas appservers built on OSGi**
- **JBoss beginning a re-architecture**
- **Abobe Creative Suite**
  - **Used for a plugin architecture like Eclipes**
  - **Isolate dependencies beteween installed plugins**
  - **Keep minimal components running (more efficient use of system resources)**
  - **dynamic loading and unloading of plug-ins on demand**
  - **"The result is a smarter, leaner and faster application that provides a better user experience. "**

OSGi as an appserver

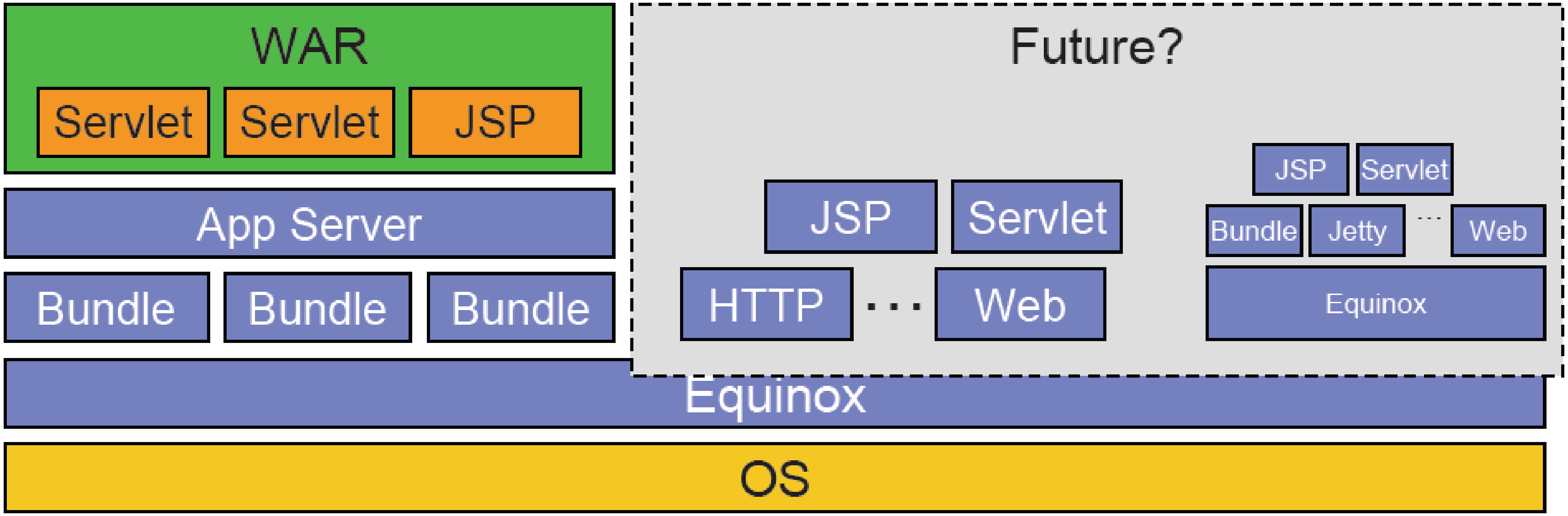
- **Traditional Appserver:**
  - **Server functionality packaged as WARS or part of the spec**
  - **Isolation provided by WARs**
  - **Installation/management at the granularity of WARS**



- Jeff McAffer & Tom Watson

OSGi – the new appserver platform?

- **Add specific server functions (ie: HTTP) as bundles, or add a whole appserver as a bundle (ie: Jetty, Tomcat, WS)**
- **Install only what you need**



- **http://www.infoq.com/osgi** **- Jeff McAffer & Tom Watson**

Trend: OSGi for application modularity

- **OSGi can form the perfect base for componentization in Java EE apps**
- **BPS (Toronto) is choosing it for their next re-architecture:**
  - **An interesting problem we have been struggling with for some time now is how to run multiple versions of a service simultaneously, in the same VM. The scenario is that two applications, A and B, have been integrated with our application, C. After the initial deployment of C, features are added to support the next release of A. Now the fun starts, as we need to update our deployed application with the new code, but without a server restart, and without breaking anything that B depends on. OSGi helps us solve this type of problem through the ability to dynamically provision and version software components (bundles).**
  - **The design center for OSGi matches our requirements: a lightweight, in-process, service container framework with full lifecycle support**

Java the Platform, not the Language

- **Java has been successful as a platform**
  - **Write once, run anywhere**
- **A Java-language only strategy is not sustainable**
- **The Java platform is no longer just Java**
- **Java = JVM + JDK = bytecode = Any language**
- **Java is becoming a host to other languages**
  - **Groovy, Beanshell, VBScript (project Semplice), Javascript (project Phobos), Ruby (JRuby)**
- **Lack of standards / stablility have prevented mass Java-enabled scripting language adoption in the Java community**
- **Having multiple languages in Java will attract a wider audience to Java, use best tool for the job**
- **Good strategy to increase mass Java adoption**

Java as a platform for scripting languages

- Dynamically typed Scripting languages such as Groovy becoming very popular
- Groovy combines features from languages like Python, Ruby, Smalltalk, and Perl in a very Java-like syntax
- Designed to be quick and easy to use:
  - Optional static typing
  - Regular expressions, lists and maps are first class citizens
  - Terse, light weight syntax
  - Easily hook into system resources, especially the JDK!
- Groovy code compiles to Java .class files, and have access to the JDK
  - This is why Groovy got so popular

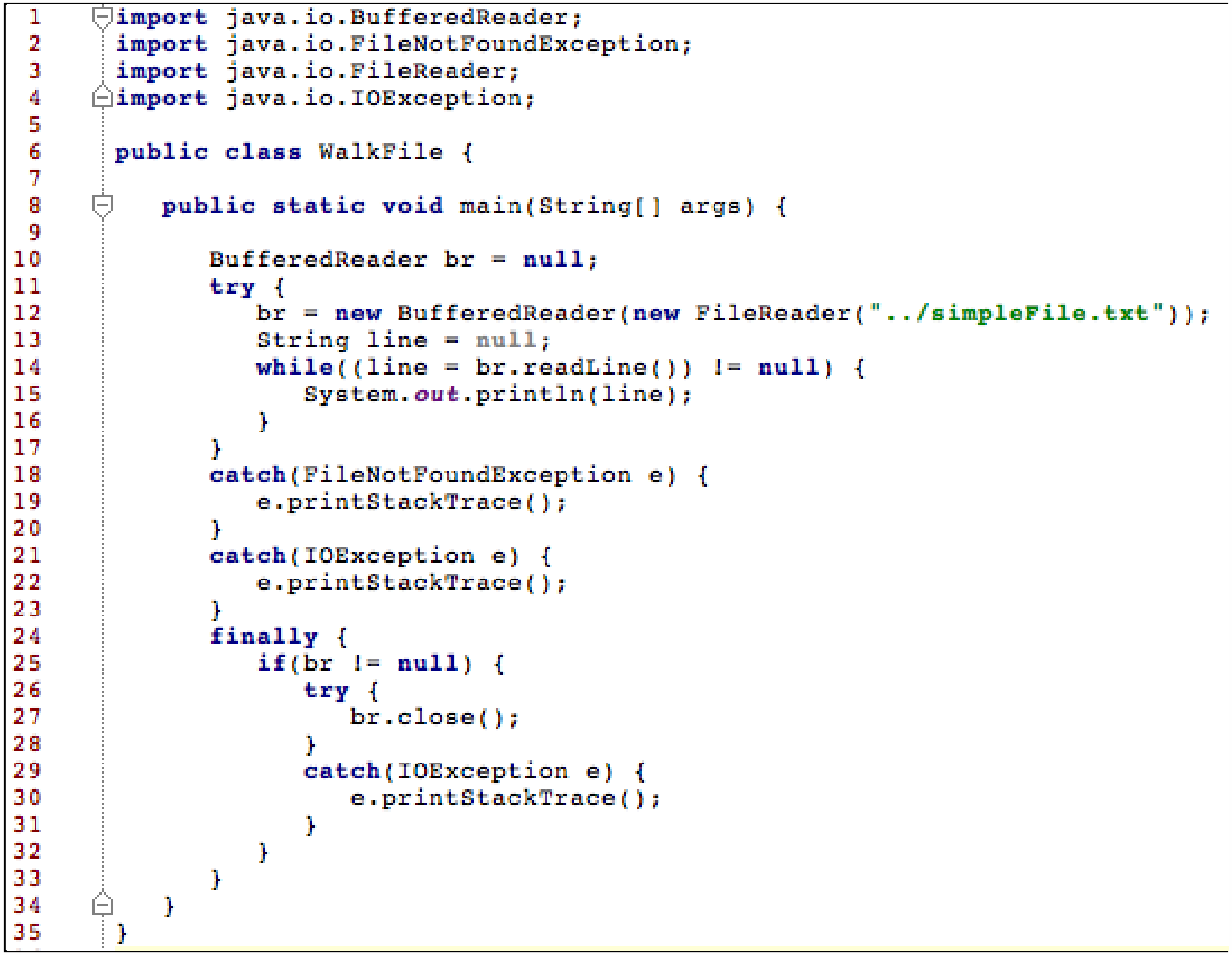
… Java as a platform for scripting languages

- **Developers can be more productive than in Java when:**
  - **IO/Data manipulation**
  - **SQL scripting**
  - **Gui prototyping**
  - **Unit testing**
  - **Batch/script processing – used on TSS as cron scripts**
  - **Implementing actions in MVC web apps**
  - **Workflow / Rules / Integration**
  - **Build system, embedded in ant scripts**
  - **Any non-OO, linear procedure**

Groovy

- **a scripting language that runs in Java**
  - **You can run it as an interpreted (uncompiled) scripting language (allows changing source code at runtime)**
  - **You can compile it into bytecode and mix it in with your existing Java codebase**
  - **You can have a Groovy script configured as a Spring Bean**
- **Groovy Greatly Simplifies**
  - **File I/O**
  - **GPath – interact with XML as objects**
  - **XML Builders – use objects to create XML**
  - **SQL**
  - **Groovlets (aka Servlets)**
  - **Grails**

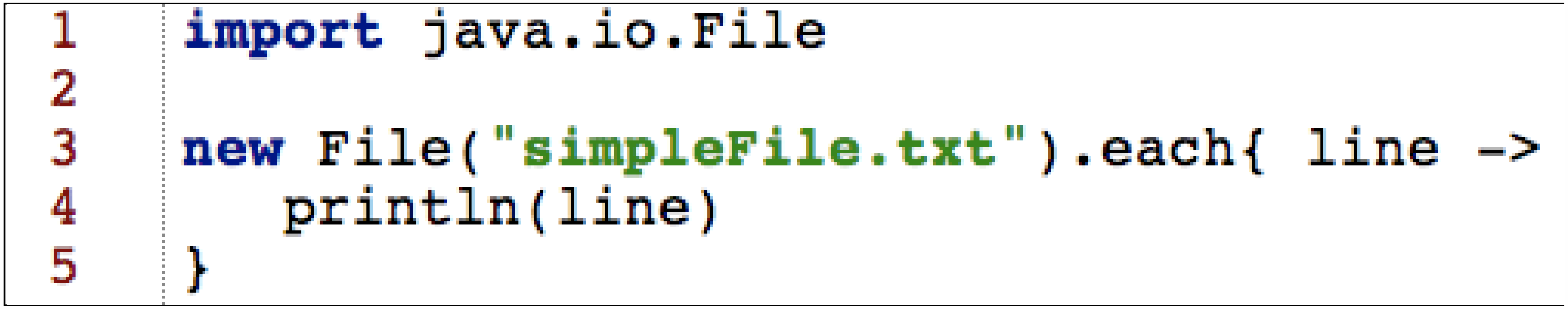
Groovy – Turns this:



```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class WalkFile {

    public static void main(String[] args) {

        BufferedReader br = null;
        try {
            br = new BufferedReader(new FileReader("../simpleFile.txt"));
            String line = null;
            while((line = br.readLine()) != null) {
                System.out.println(line);
            }
        }
        catch(FileNotFoundException e) {
            e.printStackTrace();
        }
        catch(IOException e) {
            e.printStackTrace();
        }
        finally {
            if(br != null) {
                try {
                    br.close();
                }
                catch(IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

- Scott Davis

…Groovy: into this



```
import java.io.File

new File("simpleFile.txt").each{ line ->
    println(line)
}
```

- Easily allows DSLish syntax:

myAccount + 500.euros
myAccount - 200.dollars
myAccount << 500.euros
myACcount >> 500.euros

Ruby on Rails & the integrated stack

- **Bruce Tate: “For web-based apps on a relational database where you control your own schema, you'd be crazy not to consider Rails”**
- **Reasons**
  - **Integrated stack**
  - **Removes the compile/deploy/debug cycle – just save and reload**
  - **Starting point – working crud app per table**
  - **Reduced configuration – almost no XML**
  - **Multi-purpose language**
  - **Lots of people reporting 10x productivity enhancements**

Ruby on Rails & the integrated stack

- **RoR early adoption indicators**
  - **Defection of key java evangelists**
  - **20,000 purchases of Dave Thomas' book**
  - **400+ messages in TSS thread makes it top 5 of all time**
  - **Lots of noise in blogspace**
  - **Outspoken, loyal community**
  - **Consultants actually charging 30% less for Rails projects**
- **Late adoption indicators such as job postings, show little, but growing traction for Rails**
- **Prediction – Rails was an opportunity for growth in the Java community, not really a threat**

JRuby brings Rails (and Ruby) to the JVM

- **Sept 2006 – Sun hires JRuby team**
- **JRuby aims to make Ruby run well on the JVM**
- **Rails runs ‘well’ on JRuby right now**
- **Deployment & performance a problem in the Rails community, being addressed by Java**
- **ThoughtWorks uses JRuby to deploy Rails apps for their customers**
  - **Easier to get their Rails app deployed a customer sites!**
- **JRuby compiler coming soon (probably by June 2008)**

Grails – Java fights back

- **Grails provide a Rails-like environment that integrates with existing Java infrastructure (via Groovy), and based on Java idioms, best practices, and technologies**
- **Groovy-based integrated full-stack framework combining:**
  - **Spring MVC, Hibernate, Quartz, SiteMesh, Compass, etc**
  - **"just a Spring MVC application under the covers" – G.Rocher**
- **A number of commerical sites deployed Grails in 2007**
- **Impressive code generation**
  - **Write some domain objects and generate a whole CRUD web app and O/R mapping**
  - **Or generate a web app from existing java entity beans**
  - **Convention over configuration**
- **http://www.infoq.com/grails**

Grails more “Enterprise” than Rails?

- **Easier to start using Grails in Java projects**
  - **Integration with Spring, Hibernate**
  - **Can build a Grails app around EJB3 / Hibernate POJOs**
- **IT/Operations people only see Java. No additional work for them**
- **Grails controllers use standard Servlet API objects like request, response, session etc. and can sit alongside other servlets (Graeme Rocher)**
- **More OO – no “domain models” in Rails**

Browser-based applications (web 2.0)

- **Enterprise User Interface evolution**
  - **Dumb terminal**
  - **Thick client / Client-Server / Fat Client**
  - **Today's standard: Thin client / HTML+HTTP / three tier**
- **Benefits of thin client / HTML based UI's:**
  - **Zero install**
  - **Standard platform for hosting/viewing UI's (the browser)**
  - **Real time updating of application (pages served from the server)**
- **Problems with thin client / HTML UI's:**
  - **Lack of rich features available in the desktop worlds**
  - **Stateless, request-response content-retrieval system**
  - **Server maintains state, all operations need to go to the server**
  - **Poor support for non-linear workflows**

Web 2.0

- **Definition: Web 2.0 is the internet as an application platform, as opposed to its previous purpose as a publishing platform**
- **Browser/Standards based:**
  - **AJAX – Asynchronous Java Script and XML**
- **Brower-installed rich client runtime based:**
  - **Adobe Flash/Flex, Lazslo**
  - **JavaFX**
  - **Microsoft Silverlight**

Java to make a comeback on the client?

- **The initial Java Download Problem**
- **The Consumer JRE – focus on installation & deployment**
  - **Quick run only small JRE kernel + what you need to run your applet or webstart app**
  - **Rest of JRE downloads in background after app start**
  - **QuickStart – prefetch portions of JRE in memory to decrease the average JRE start-up time after reboot**
  - **Limewire starts up 3x faster – Chet Haase**
- **JavaFX Script, Designer and Developer Tools**
  - **DSL for accessing rich UI features of the SE stack**
- **JavaFX Mobile**
  - **entire OS + apps + apis written entirely in Java**

Rich Client Revolution– Adobe Flex/Flash

- **Flash VM is ubiquitous – in every major browser, on every major OS. *As standard as AJAX***
- **Flex – Adobe's platform for rich browser based app development that runs in the Flash VM**
- **June 2006 – Adobe makes Flex SDK free**
- **Flex server side declarative model (JSP like)**
  - **vector graphics, drawing APIs, rich media (video/audio)**
  - **UI widgets and gui components**
  - **dev tools such as SWF compilation, debugger**
  - **Actionscript (javascript equivalent) for coding**
  - **Native remoting to Java, or web services**

Why Flex/Flash

- **Performance**
  - **Flex/Flash apps run as bytecode in the Flash VM using JIT**
- **Simplicity / Reliability**
  - **No worries about browser compatibilities, browser DOM/JavaScript irregularities, etc**
- **Expressiveness (vector graphics)**
  - **Microsoft betting on this future with WPF/Avalon**
- **Real time (binary sockets and pub/sub messaging)**
- **Rich media (video and audio streaming)**
- **Adobe AIR – run web apps on the desktop**
  - **local storage, desktop launchable client, etc**
- **Google, YouTube, InfoQ – making Flash even more widely known**
- **http://infoq.com/flash**

Future trend

- **Desktop & Browser apps merged**
  - **Apps built with new browser runtimes also run-able form the desktop**
  - **Web apps with offline storage**
    - **(e.g.: Google Reader, via Google Gears)**

Most popular AJAX frameworks

- **Java Community**
  - **Prototype (35.2%), DWR (31.2%), and Dojo (28.6%). GWT was at 7.2%. 22% do direct XMLHTTPRequest.**
- **.NET**
  - **Prototype (32.4%), MS Atlas (27.2%), Dojo (17.6%), AJAX.NET professional (14%). 37% do direct XMLHTTPRequest.**
- **Rails**
  - **Prototype (74.8%), Dojo (20.2%). Interestingly, 67.2% use Scriptaculous for effects. Only 16% do direct XMLHTTPRequest.**

Source: analysis and filtering of ajaxian survey

DOJO getting major industry traction

- **Dojo Toolkit**
  - **Abstracts XMLHTTPRequest**
  - **Javascript libraries and convenience utilities**
  - **Rich DOM manipulation support**
  - **Out of the box AJAX widgets**
- **Sun backs DOJO – June 2006**
  - **Contributing ajax widgets, helping with documentation**
  - **Adding tooling for netbeans**
- **IBM backs DOJO – June 2006**
  - **Donates i18n code**
  - **Committed to making DOJO support accessibility**

Multi-core & Parallelism

- **Single threaded performance to stagnate or decline in future**
- **8 core boxes to be come common and commoditized (2 quad core cpus) for server applications**
- **Amdahl’s law: app speed limited by non-parallel parts of your app. N tasks do not execute in 1/N time**

…Multi-core – what is hbeing done about it?

- **Language Designers have acknowedged that we need new language contructs:**
  - **Fortress – research language at Sun**
  - **IBM X10 – language artifacts for managing concurrent operations and the distribution of data associated with those operations.**
  - **Parallel Extensions for .NET**
- **IO Performance may make up for loss**
  - **New solid state drives show promising potential**
- **Virtualization**
  - **Split a large box into more manageable sub-instances and then cluster**

Domain Driven Design – Eric Evans

- **Principle of focusing on the domain**
- **Make the domain a reflection of real world concepts**
- **Collaborate with business owners to ensure a realistic domain model**
- **Ubiquitous language**
  - **A language structured around the domain model and used by all team members to connect all activities the of the team with the software.**
- **model**
  - A *system of abstractions* that
    - describes *selected* aspects of a domain
    - and can be *used* to solve problems related
    - to that domain.

Domain Driven Design – Why now?

- **5 year web learning curve over, web technologies no longer ‘getting in the way’ of domain logic**
- **Software projects getting more complex**
- **Agile practices emphasizing stakeholder collaboration**
- **InfoQ DDD book 20,000 downloads**
- **DDD is becoming an important buzzword in 2007 – it’s just starting now**

Domain Specific Languages

- **Languages ‘useful for a specific set of tasks’, as opposed to *general purpose* languages**
- **Adding domain concepts to the language instead of libraries/frameworks**
- **DSLs should:**
  - **Contain abstractions from the domain**
  - **Be readable by business stakeholders**
  - **Increase productivity and maintainability**

DSLs

- OO builds up a vocabulary
- DSLs add the grammar
- “this shift of moving from thinking about vocabulary, which is objects, to the notion of a language that combines vocabulary and grammar.” (Ford, Fowler)

DSL Recent Discussions

- **Design Patterns are a sign of language deficiency – Mark Dominus**
  - **Historically – abstractions defined in patterns have migrated to languages (e.g.: Structs, Objects, Iterators)**
  - **"Instead of seeing the use of design patterns as valuable in itself, it should be widely recognized that each design pattern is an expression of the failure of the source language. "**
- **DSLs are a bad idea because they do not do a good job controlling change over time. - Buko Obele**
  - **"the *expressive power* of a language is not the measure of a language's ability to model a problem domain, it's rather the ability of the language to control changes in the problem domain. "**
  - **DSL's only good for static, non-changing domains**

The end of the single language projects?

Neal Ford & Martin Fowler: Polyglot programming

- **"...will see multiple languages used in projects with people choosing a language for what it can do in the same way that people choose frameworks now" Martin**

Embedding DSLs instead of using libraries

- **Fowler asserts that "larger frameworks like Hibernate, Struts, ADO present as much of a challenge to learn as a language even if you program them in a single host language."**

What's different this time vs. 1980:

- **Interoperability between languages on the JVM/CLR minimizes 'silo risk'.**

Web services and Service Oriented Architecture

- **Web services being used pragmatically – far below the initial hype**
- **SOA – everyone talking about it – few realizing it's full potential**
- **What is SOA?**
  - **SOA is an architectural style that encourages the creation of loosely coupled business services**
  - **Loosely coupled services that are interoperable and technology-agnostic enable business flexibility**
  - **A SOA solution consists of a composite set of business services that realize an end-to-end business process**
  - **Each service provides an interface/contract-based service description to support flexible and dynamically re-configurable processes**

- John Reynolds

…The truth about SOA

- **Few are actually doing SOA, most simply using web services:**

  1. We connect fat .Net clients to Java application servers (and use soap in between)
  2. We connect to some ASP (like Amazon, SF.com, etc.)
  3. We front ended some legacy system with services.

  – Jeff Schneider

- **Most applying the 3+N pattern**
  - **Three tier application using web services to talk to N external services – this is not service composition**

- **When services are “composed” into wider business processes, then you have an example of SOA**

- **We are seeing successes with SOA – notably amazon**

SOA

- **IT/Business Alignment – the only thing 'NEW' about SOA**
- **SOA cannot succeed as an IT initiative, it must start from the business down**
- **First define the services that bring business value, then figure out the technology stack**
- **Agile SOA approaches**
- **Governance the biggest challenge with SOA – not technology**
- **ESBs – messaging, routing, transformation**

Future Importance of EJB Uncertain

- **3 high-level categories of EJB benefits**
  - **Framework Benefits** – Security, Transactions,Pooling, packaging/deployment
  - **Distribution Benefits** – It's the way to do distributed communication
  - **Component Benefits**

Quick Poll: How are people using EJB?

- **Where the community has been going:**
  - **Framework** - POJO's via the three foundations: AOP + Dependency Injection + Annotations
  - **Distribution** – POJO's with remoting as an aspect, lightweight remoting frameworks, web Services
  - **Components** – Who cares? The Enterprise Component Market is dead
- **So what about EJB 3?**

Emerging Economies and Open Source

- **Emerging economies will further open source**
- **Case study: Brazil**
  - **Government mandate to use open source, and multi-platform software. Linux and Java seen as core to national interest.**
  - **Building their own open source JVM called JAVALI**
  - **Freedom from vendor lock-in important on a national scale**
  - **Guarantees they will have the rights to use it in the future**

Apache Harmony

- **Open source JVM and JDK**
- **Goals:**
  - **create a compatible, independent implementation of J2SE 5 under the Apache License v2**
  - **create a community-developed modular runtime (VM and class library) architecture to allow independent implementations to share runtime components, and allow independent innovation in runtime components**
- **Brought together some of the best minds on JVM's to contribute**

Apache Harmony – Why?

- **An open source implementation can survive if Sun / vendors stop supporting Java**
- **Distribution rights – Linux and FreeBSD & more availability for Java**
- **Ability to ship custom JVM's with an application**
  - **Eg: JVM's with custom features, or ship apps with parts of JVM's (2meg exe file, or 100KB JAR that needs the JRE?)**
- **"enables collaboration in the parts of technology that are common to all"… letting organizations focus on things that add value on top of the common – Geir Magnusson**
- **Risks**
  - **A project of this scale is no easy matter**
  - **Other similar projects (Geronimo) proceeding slowly, so might this**
  - **Forking Java**

Apache Harmony – Current Status

- **Enough of the class library to run Ant and the Eclipse compiler**
- **A standard interface for plugging any JVM into a class library**
- **Major contributions from IBM and Intel**
- **Future in question due to Sun open sourcing Java**

Open Source Java

- **Sun finally open sources Java**
- **OpenJDK**
  - **Currently has: HotSpot JVM, javac**
  - **Buildable JDK to be added in Q1 2007**
- **Java ME, and Glassfish also GPLv2**
- **Governance and Licensing are two key aspects**

Open source Java licensing & compatibility

- **GPLv2**
- **Makes Java open source while forcing any modifications to also be open source under GPL (minimizing chances of forks)**
- **Java Compatibility – can't call it Java unless:**
  - **passes the TCK**
  - **Get copyright clearance from Sun**
- **Java applications are NOT *derivative works* of GPL'd Java**

Open Source Java - Governance

- **Governance issues**
  - **Who has ultimate control – Sun or the community?**
  - **Who decides on commits?**
  - **Shared copyrights (which is the case now but wasn’t before)**

Open source Java Impact

- **On developer day to day: not much**
- **Linux greater Linux Adoption**
  - **Tim Bray: Hopes Java becomes UI technology of choice on Linux**
- **Government adoption / emerging economies**
- **New technical use cases for Java possible, but GPL may limit commercial innovation**

New open source business models

- **JBoss – pioneer, validated by RedHat purchase**
- **Simula labs – VC focused on open source**
- **Interface21, numerous others**
- **Gluecode software – IBM buys Gluecode**
- **Terrracotta**

Marc Fleury’s take

- **“the power of the model rests in the extremely low cost of distribution and sales. We reach millions of folks with free distribution and then monetize this base. It is a very efficient way to acquire customers. The result is that we spend 30 cents for every dollar of maintenance revenue, while the competition, on average, spends $3 for every dollar that ultimately comes in as maintenance. The downside, compared to proprietary software, is that on average we only monetize 3% of our user base for JBoss and roughly 10% for Linux. This low cost of sales we achieve through mass distribution is what makes the model tick. The customer gets to make up his own mind as to whether the software is any good as opposed to having to go through the vendor’s pricey and biased salesforce. “**

Casestudy: Terracotta

- **a JVM clustering solution that can turn single-node, multi-threaded apps into distributed, multi-node apps with no code changes**
- **Viable, cool product**
- **How the VC's were convinced:**
  - **80% of installs of Terracotta were in environments that were mixed with open source**
  - **Proof-of-concept-eval process for commercial license too complex – barrier to adoption**
- **Model:**
  - **Greater return per-customer after 5 years of support contracts compared to upfront license**
  - **Must have quality support services to keep a 5 year subscription**
  - **Must have order of magnitude more customers**
  - **Focus on delivering value**

What’s left for commercial models?

- **Open source software is increasingly standardizing more complex infrastructures**
- **Open source also enabling commercial models**
  - **Release open source version to build a qualified customer base you can then upsell to enterprise versions.**
  - **Everybody is doing it – IBM, Adobe, Sun, and numerous smaller players**
- **Opportunities for commercial firms**
  - **High end, large scale (Websphere XD, BEA Tuxedo)**
  - **Niche tools – vertical specific, highly specialized**
  - **Anything ahead of the open source curve**
- **Open source will ensure that there are many more software development jobs, but perhaps less opportunities for product companies**

Questions / final

- **InfoQ's mission is to *track change and innovation in the Enterprise Software Development Community***
- **We hope you use the site, help us spread the word!**
- **Want to write an article? Email editors@infoq.com**

InfoQueue

Enterprise Software Development Community