



Quality Assurance with AspectJ

Gesellschaft für Informatik, Regionalgruppe München
Arbeitskreis Software-Engineering Live

Per Fragemann, Gerhard Müller
12th of December, 2006



**“My MasterCard
is virtual.”**

Get a virtual MasterCard with your
Wirecard account in only 30 seconds.

Endorsed by
MasterCard™

[SIGNUP](#)

[DEPOSIT MONEY](#)

[SEND MONEY](#)

[SECURITY](#)

[LANGUAGE SELECTION](#)

ENGLISH

What is Wirecard?

As easy as cash. As convenient as a card. Send money, receive money – worldwide and in realtime. Pay with your virtual MasterCard. Experience the advantages of a full bank account, which is available 24 hours a day, worldwide.

- Open a Wirecard account in only 30 seconds - free of charge.
- Pay with your virtual MasterCard worldwide at millions of merchants on the Internet.
- Experience international money transfers in real-time.



LOGIN

Username

Password

[Login](#)

[» Forgot password](#)

FAQs

For further information, see the Frequently Asked Questions.

Deposit Money

Amount / Currency 40,00 EUR

Fees 1,20 EUR

Deposit Method Credit Card, MASTERCARD 523252*****2713

Card Verification Code

Debited Amount 41,20 EUR

[Cancel](#)

[Deposit Money](#)

Fees

When you deposit money to your Wirecard-account by credit card, a small fee is deducted. To learn more about fees please refer to the table of fees.

[German Version](#)

[English Version](#)

Card Verification Code

The Card Verification Code is printed on your credit card and does not belong to the card number. The Card Verification Code are the last three digits on the signature field on the back of your card.



Endorsed by
MasterCard™

Send Money

Amount	35,00 EUR
Subject	Taxi money
Message	Here you get your money back, thanks again
Fees	0,50 EUR
Total	35,50 EUR

Recipient

Email Address

Wirecard Account No. 4610486097

Enter your activation code

Activation Code

[Cancel](#)[Send Money](#)

Activation Code

An activation code was sent to your mobile phone via SMS. Please enter this code into the respective field to confirm the transfer. In case you did not receive an activation code via SMS, please contact our service team.



[Overview](#)[My Profile](#)[Logout](#)[Help / FAQ](#)[DEPOSIT MONEY](#)[SEND MONEY](#)[MY ACCOUNT](#)[MY MASTERCARD](#)

Your postings of the last two weeks

Date	Posting information	Status	Credited	Charged
▶ 23.11.06	P2P Transfer 4610337608	✓	10,00 EUR	
▶ 26.11.06	P2P Transfer 4610421666	✓	0,06 EUR	
▶ 28.11.06	Card Payment PLAN NINE PUBLISHING	✓		16,32 EUR
▶ 28.11.06	Credit Card VISA, 403649*****3501	✓	20,00 EUR	
▶ 28.11.06	Credit Card MASTERCARD, 523252*****2713	✓	30,00 EUR	
▶ 29.11.06	1020769202 / 20050550	...	33,00 EUR	
▶ 30.11.06	P2P Transfer 4610486097	✓		-10,50 EUR
▶ 30.11.06	P2P Transfer 4610358043	✓		-3,19 EUR
▶ 30.11.06	Wire Transfer PER FRAGEMANN	✓	11,00 EUR	
▶ 04.12.06	Credit Card MASTERCARD, 523252*****2713	✓	30,00 EUR	

Wirecard | [Deposit Money](#) | [Send Money](#) | [Merchants](#) | [Fees](#) | [Security](#)
[Help](#) | [Contact](#) | [Help / FAQ](#)
Website | [Home](#) | [Imprint](#)
Company | [About Us](#) | [Terms & Conditions](#) | [Privacy Policy](#)

The Speakers

Per Fragemann

- Technical Project Lead at Wirecard AG for the new “Wirecard” product
- Studied Computing Science at Universität Hamburg
- Hands-on developer & architect with strong emphasis on test-driven development and agile processes

Gerhard Müller

- Partner at TNG Technology Consulting GmbH, <http://www.tngtech.com>
- Studied Computing Science at Technische Universität München and University of Newcastle upon Tyne
- Technical project management with agile, test driven development processes and software architecture

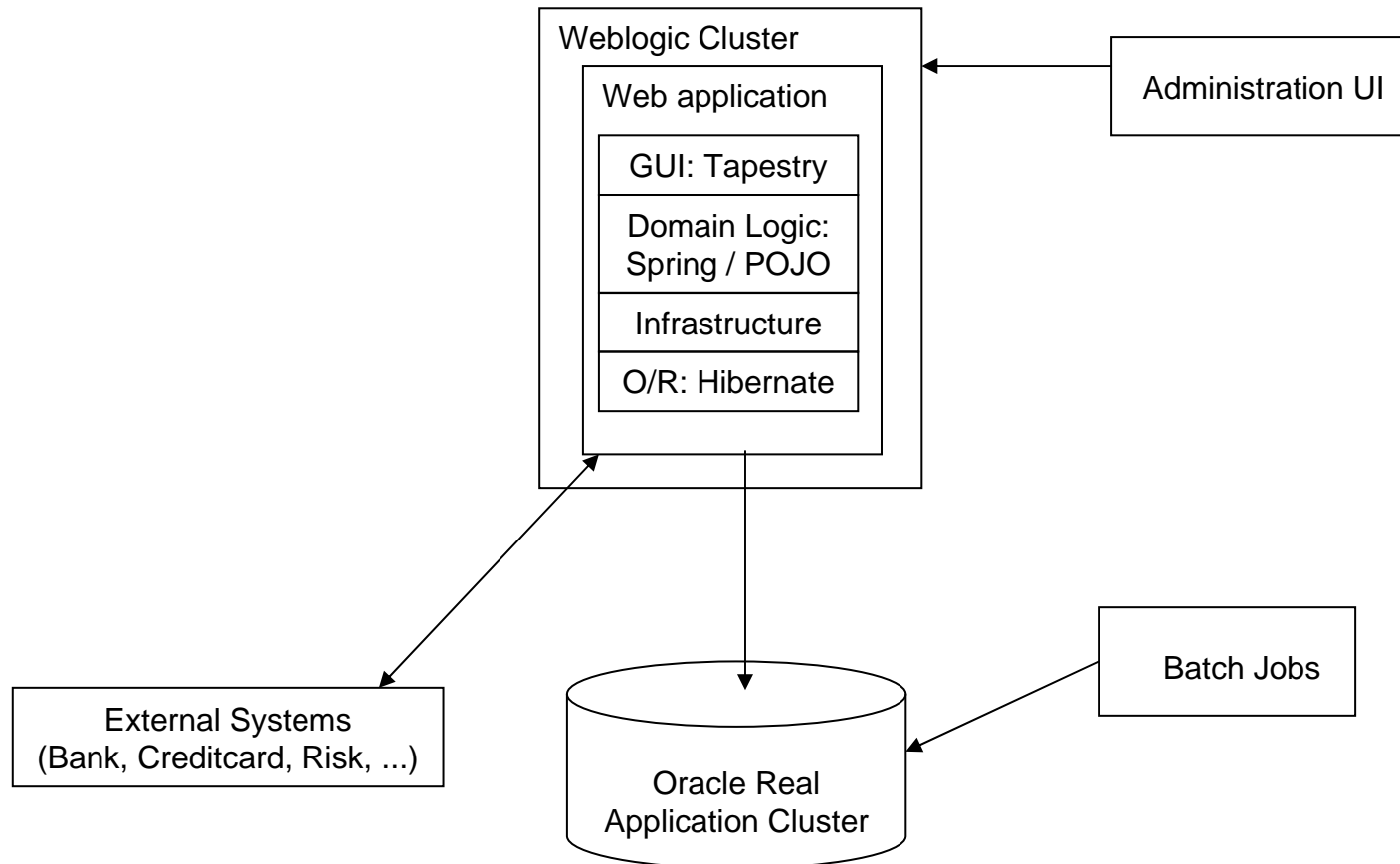
Agenda

1. Motivation
2. Project Challenges & Environment
3. Feedback Processes
4. Brief AspectJ Introduction
5. Quality Assurance with AspectJ
6. Summary and Outlook

Project Challenges

- Time-to-Market – tight schedule
- Dependencies to external systems
- Rapidly growing & mixed project team
- Different levels of experience
- Complete rewrite from scratch, using latest technologies
 - Lots of learning necessary
 - No established usage guidelines for new technologies available
 - Propagating new knowledge causes communication overhead

Technical Project Overview (vastly simplified)



Technical Project Overview – Tools and Technologies

IDEs: IntelliJ IDEA, Eclipse
<http://www.jetbrains.com/idea/>
<http://www.eclipse.org/>

Issue Tracker: JIRA
<http://www.atlassian.com/software/jira/>

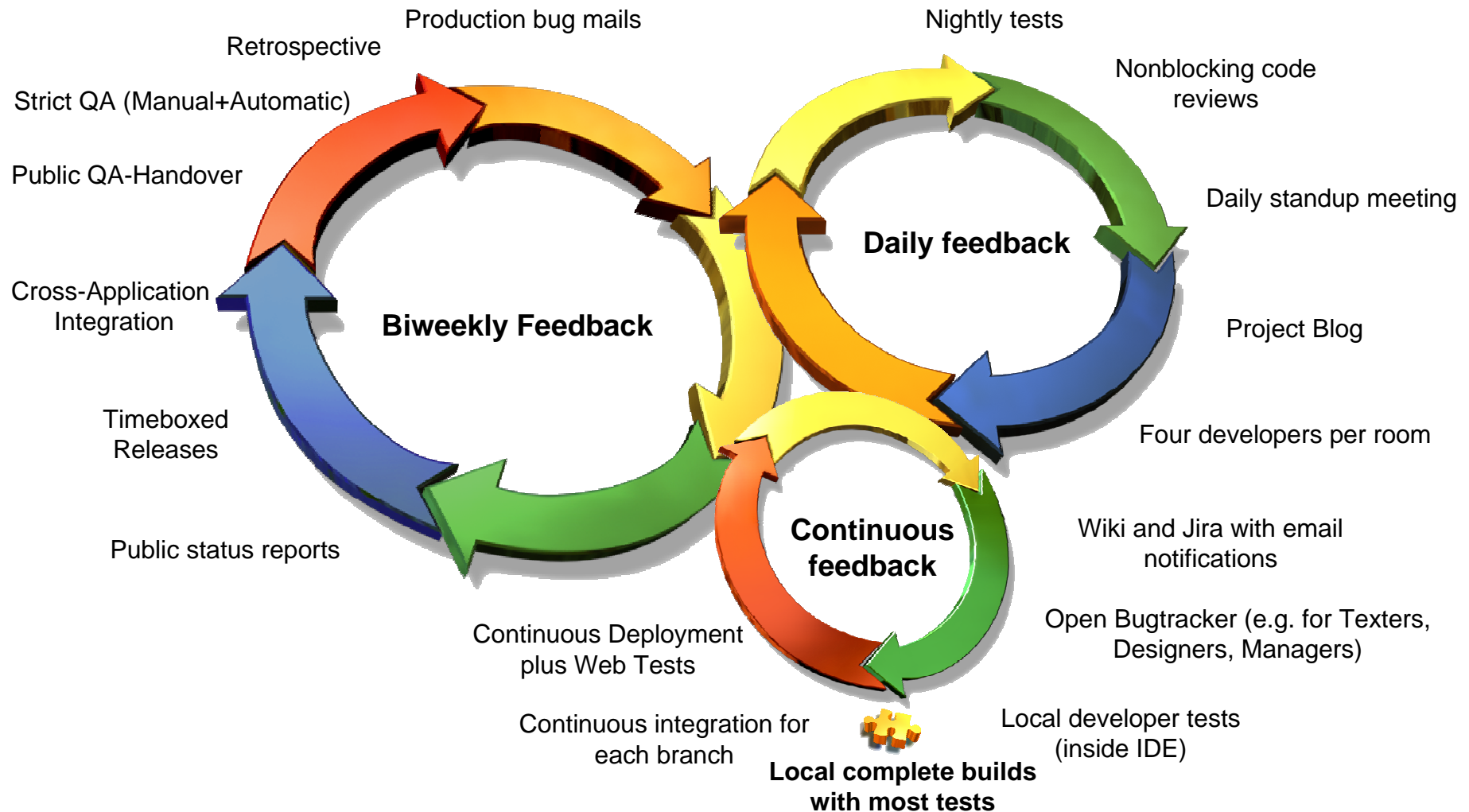
Build System: Maven2
<http://maven.apache.org/>

Wiki: Twiki
<http://twiki.org/>

VCS: Subversion
<http://subversion.tigris.org/>

Continuous Integration: Cruise Control
<http://cruisecontrol.sourceforge.net/>

Feedback Processes in the Development Team



AspectJ on one Slide (Informal Description)

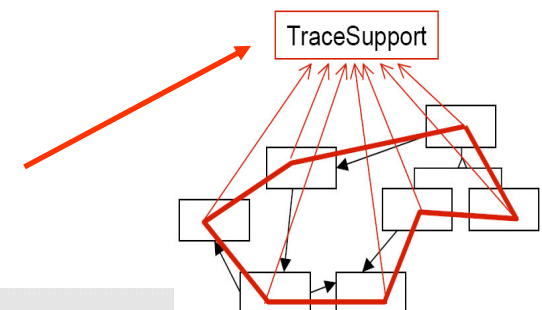
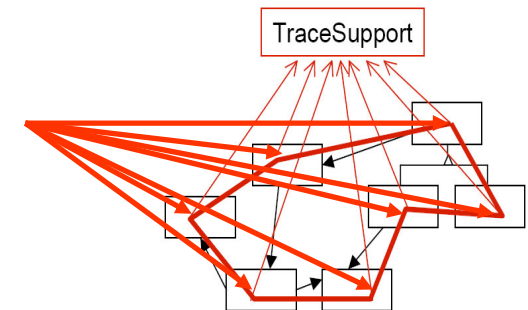
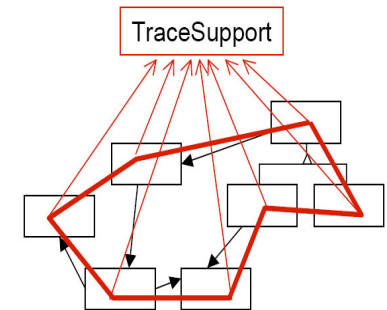
- AspectJ extends the Java language with aspect-oriented programming techniques (AOP)
- The developer writes small Java-like rules and extensions into *.aj -textfiles
- AspectJ ships with an extended Java-compiler that weaves these enhancements into the bytecode
- The resulting classfiles are regularly usable by normal JVMs -> no black magic, „just a compiler-extension“
- Additional Tools (e.g. AJDT for Eclipse) provide syntax highlighting and show how the aspects will influence the sourcecode

See also:

- <http://eclipse.org/aspectj>
- <http://en.wikipedia.org/wiki/AspectJ>

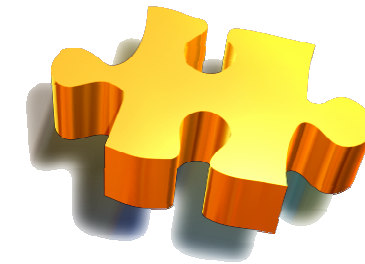
AspectJ Concepts

- Join points
 - Events that occur during program execution – join points have always existed in Java programs
 - A join point model defines the kinds of join points available and how they are accessed and used.
- Pointcuts
 - Select join points of interest (“Where & When”)
 - A pointcut is a predicate that matches join points. A pointcut is a relationship ‘join point -> boolean’, where the domain of the relationship is all possible join points.
- Advice
 - Java code that executes where pointcuts match (“What”)



Quality Assurance with AspectJ

- Problem of enforcing general „good practice“
 - Architect knows how the application code should behave
 - Junior developers do not know this yet
 - Architect can't watch the juniors all the time
- Enforce project-specific decisions / rules
 - Architect (transparently) switches the compile process to AspectJ's ajc
 - Architect adds AspectJ-rules that will cause compile errors as soon as developers break the rules
 - Example rule: „Never modify Date objects“
- Architecture Enforcement
 - Not just Library usage can be monitored: Define rules for major package and layer-dependencies



Advantages of Quality Assurance with AspectJ

- Immediate feedback „build is broken“ (locally before committing) increases attention for architectural patterns
- Junior & new developers will learn faster
- Newly discovered rules can be embedded easily once the infrastructure is in place
- Subproject-Pattern can be delayed until the project is in a stable state – new projects can save some time at the time by having simpler build processes
- Rules don't have to be monitored manually, code reviews can focus on other problems

Our first aspect: Enforcing correct Date-usage

```
public aspect DatesAreImmutable {  
  
    pointcut dateModificationCalls():  
        call (* java.util.Date.set*(..));  
  
    declare error: dateModificationCalls():  
        "dates have to be treated as immutables, no changes to existing  
        date objects are allowed";  
  
}
```

Checking System.out

```

public aspect SystemOutUsage {

    pointcut systemOutErrUsage():
        get(* System.out) || get(* System.err) || call(* Throwable+.printStackTrace());

    pointcut inAllowedProjectSources() :
        withincode(* service.runtimedetection.impl.RuntimeDetectionServiceImpl.*(..)) |
        withincode(* util.outputter.SystemOutErrOutputter.*(..)) ||
        withincode(public static void com.companyname..*.main(String[]));

    declare error: (systemOutErrUsage() && !inAllowedProjectSources() ) :
        "Please do not use System.out/System.err/Exception.printStackTrace(), but use
        log4j logging! Please refer to http://twiki/bin/view/<confidential> for further
        details.";

}

```

Quality Assurance with AspectJ: Architecture Enforcement (Simplified)

```

public aspect SystemArchitecture {
    public pointcut uiCall() :
        call (* ui..*(..));

    public pointcut domainCall() :
        call (* domain..*(..));

    public pointcut infrastructureCall() :
        call (* infrastructure..*(..));

    public pointcut inUI() :
        within(ui..*);

    public pointcut inDomain() :
        within(domain..*);

    public pointcut inInfrastructure():
        within(infrastructure..*);

    declare error: inDomain() && uiCall() :
        "No calls allowed from domain to ui";

    declare error: inInfrastructure() && domainCall() :
        "No calls allowed from infrastructure to domain";

    declare error: inInfrastructure() && uiCall() :
        "No calls allowed from infrastructure to ui";
}

```

Prohibiting certain calls

```
public aspect UsageOfWeblogicProblematicClasses {
```

```
    pointcut httpURLUsage():
```

```
        call(* java.net.HttpURLConnection.*(..))
```

```
        ||
```

```
        call(* javax.net.ssl.HttpURLConnection.*(..));
```

```
    declare error: httpURLUsage():
```

```
        "Please do not use HttpURLConnection / HTTPSURLConnection, as the  
        keystore used inside weblogic when using these classes is NOT the one we  
        configured / agreed with operations (some weblogic specific keystore would  
        be used). See http://twiki/<confidential> for detailed reasoning.";
```

```
}
```

Checking Usage of Exceptions

```
public aspect ExceptionUsage {  
  
    pointcut exceptionUsage():  
        call (java.lang.Exception.new(..)) ||  
        call (java.lang.Throwable.new(..)) ||  
        call(java.lang.RuntimeException.new(..));  
  
    pointcut allowedExceptionUsage():  
        within (util.errorhandling.StackTraceUtils);  
  
    declare error: exceptionUsage() && (!allowedExceptionUsage()) :  
        "Please do not throw Throwable, Exception or RuntimeExceptions, but  
        instances of SystemException or DomainException";  
}
```

Checking Usage of internal APIs for testing purposes only

```
/** check that high risk calls to services are never used within production code:*/
public aspect TestApiAccess {
    pointcut projectTestApiMethods():
        call (* UserActivationService.activateUserAndCustomerWithoutChecks(..));

    declare error: projectTestApiMethods():
        "This method is only allowed from within test code.";

    pointcut dummyMoneyProviderUsage():
        call (domain.accountingmodel.DummyMoneyProvider.new(..));
    declare error: dummyMoneyProviderUsage() :
        "Please do not use DummyMoneyProvider outside of test code";
}
```

Checking method calls from allowed sources only

```

/** check that the following calls are not done from within production code
void setId(ID id) {
void setInsertedBy(String insertedBy) {
void setUpdatedBy(String updatedBy) {
*/
public aspect BasePersistentObjectSetUsage {
    pointcut basePersistentbaseMethods():
        call (void persistence.base.BasePersistentObject.setId(..)) ||
        call (void persistence.base.BasePersistentObject.setInsertedBy(..)) ||
        call (void persistence.base.BasePersistentObject.setUpdatedBy(..));

    pointcut inAllowedProjectSources() :
        withincode(* ui.commandline.setup.SpecialAccountsMasterData.createOrUpdate(..)) ||
        withincode(* domain.accountingmodel.impl.BasicAccountImpl.updateBalance(..));

    declare error: basePersistentbaseMethods() && !inAllowedProjectSources() :
        "it is not allowed to call setXXX() methods of the basic attributes of BasePersistentObject, as they
        are only allowed to be set by the hibernate interceptor";
}

```

Integration of AspectJ into a project

- Notepad / other editors: for simple examples good enough
- IDE: Eclipse has nice AspectJ support (AJDT)
- Java 5 Annotation Syntax: would work also with other IDEs

- Build integration: Maven / Ant / etc

Limitations of using AspectJ for static code checking

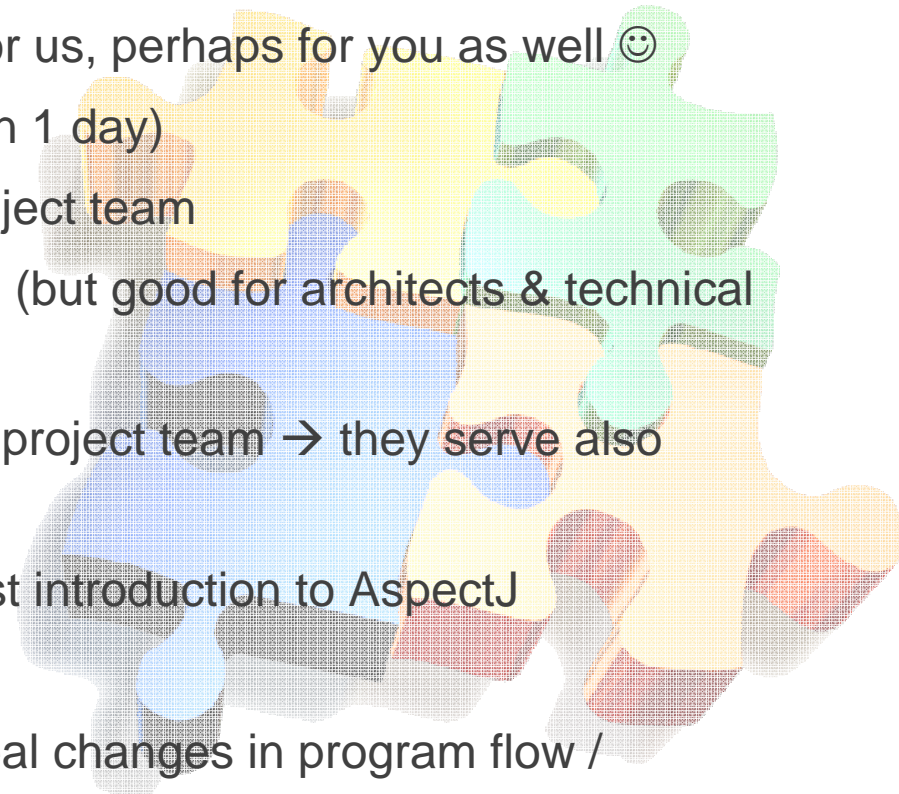
Use the right tool for the job. AspectJ not ideal for checking

- Interface quality
- Correct declarative transaction definition
- Duplicated code
- Tabs in code
- Localization consistency / duplicate keys in property files
- XML files start with correct XML preamble
- Consistency of database definition with domain objects with gui usage

→ AspectJ is a mosaic tile, not a silver bullet

Summary & Outlook

- Quality-related aspects worked fine for us, perhaps for you as well 😊
- Low set-up & learning curve (less than 1 day)
- Fast, project-specific feedback for project team
- Writing aspects: not for all developers (but good for architects & technical project leader!)
- Aspects are formalized “rules” for the project team → they serve also documentation purposes
- Quality-related aspects are the easiest introduction to AspectJ
- Use AspectJ in production code for real changes in program flow / crosscutting concerns (we use Spring 2, so no real urgent requirement for us)



Thank you for your time.

Wirecard AG
Bretonischer Ring 4
85640 Grasbrunn, Germany

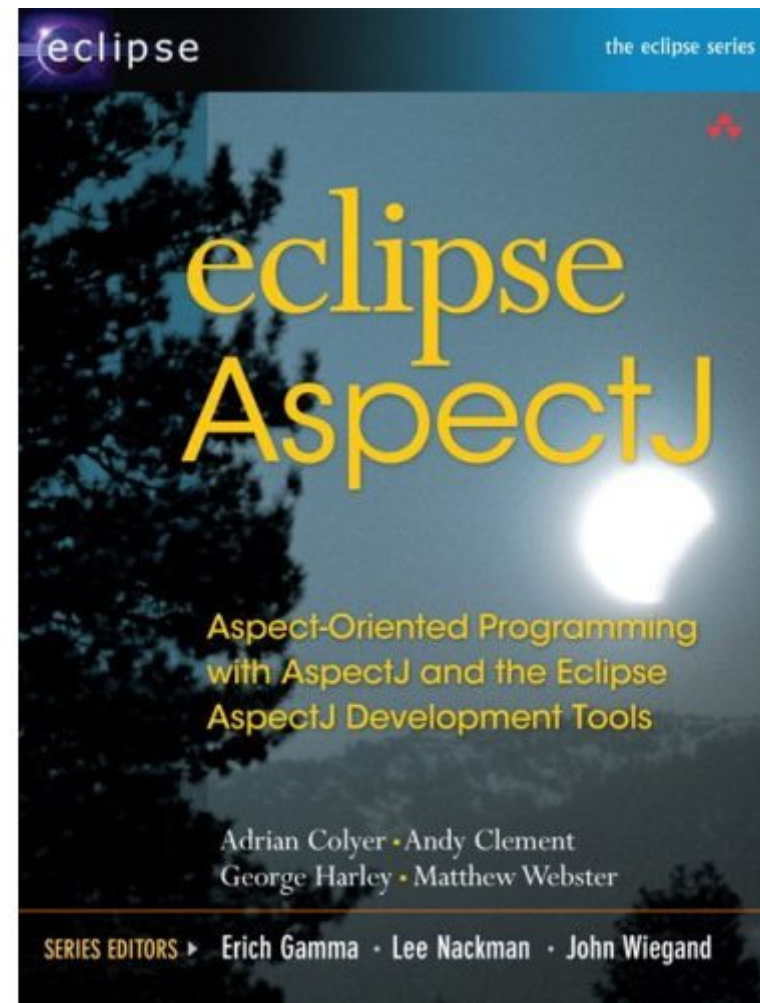
www.wirecard.com
info@wirecard.com

TNG Technology Consulting GmbH
Betastr. 13a
85774 Unterföhring, Germany

www.tngtech.com
info@tngtech.com

Links & Literature

- AspectJ:
<http://www.eclipse.org/aspectj/>
- AJDT: AspectJ Development Tools:
<http://www.eclipse.org/ajdt/>
- Adrian Colyer, Andy Clement, George Harley, Matthew Webster: Eclipse AspectJ: Aspect-Oriented Programming with AspectJ and the Eclipse AspectJ Development Tools; Addison-Wesley; 2004; ISBN: 0321245873
 - <http://www.amazon.com/exec/obidos/ASIN/0321245873>



The Speakers



Per Fragemann (Per.Fragemann@wirecard.com) works as technical project lead for Wirecard AG since July 2005. In the past 6 months he was responsible for the technical aspects of the implementation of the new Wirecard product. He studied computer science in Hamburg and worked as architect at Genteware AG for four years before joining Wirecard. Since nine years he focusses on object oriented development with Java, with a strong emphasis on testdriven development ranging from unit-testing to fullscale automated ui-coverage-testing.



Gerhard Müller (Gerhard.Mueller@tngtech.com) is partner at TNG Technology Consulting GmbH, <http://www.tngtech.com>, Unterföhring near Munich since 2001. He studied computing science at the Technische Universität München and University of Newcastle upon Tyne. He is involved in software development since 1997, especially with Java / JEE. His main focus are technical project management with agile, test driven development processes and software architecture, design and implementation of JEE applications. He is author of several articles about Java and is actively taking part in several working groups of the Gesellschaft für Informatik, Regionalgruppe München.

The Companies

wirecard

Wirecard AG is one of the leading providers of payment systems with integrated risk management services. The Company provides customers with a wide range of national and global payment options. Being a Principal Member of Visa International Service Associates (VISA), MasterCard Europe sprl. as well as JCB International, Wirecard Bank AG supplements the portfolio of services. The software platform is being used by more than 6,000 enterprises across the globe, including dba, Gulf Air, HSE24 and QVC. Wirecard AG is listed on the German stock exchange in Frankfurt (TecDAX, ISIN DE0007472060, WDI).
<http://www.wirecardbank.com> <http://www.wirecard.com>

TNG TECHNOLOGY CONSULTING

TNG Technology Consulting is a consulting partnership focused on high end information technology. We support our customers with up-to-date tools and innovative ideas. Our mission is to analyze and solve the strategic or routine IT problems our customers have. By relying on teamwork and fast communications, each customer can access the expertise pool of the whole company. We hire only the best experts. Our people have the freedom to grow and the responsibility to do so. We believe that a rational cooperation between our customers and us is possible and mutually beneficial.
<http://www.tngtech.com>