


Effektiver Einsatz von Code-Reviews

18. Java Forum Stuttgart 2015

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de

Version: 1.4



Ihr Sprecher

Thorsten Maier


Trainer, Berater, Entwickler



Schwerpunkte
Menschen
Prozesse
Architektur
Code-Qualität

© 2015 Orientation in Objects GmbH | Effektiver Einsatz von Code-Reviews | 2


Agenda



Einsteiger	<i>Code-Reviews? Noch nie gehört!</i>
Skeptiker	<i>Wir haben es ausprobiert. Es funktioniert nicht!</i>
Profis	<i>Wir setzen Code-Reviews bereits erfolgreich ein und wollen noch besser werden.</i>

© 2015 Orientation in Objects GmbH | Effektiver Einsatz von Code-Reviews | 3

Agenda



Einsteiger	Warum Code-Reviews ?	Do it!	Die richtigen Tools
Skeptiker			
Profis			

© 2015 Orientation in Objects GmbH | Effektiver Einsatz von Code-Reviews | 4

Kennen Sie das?

„Das ist historisch gewachsen.“

Oder das?

„Das wollten wir schon lange
refactoren.“

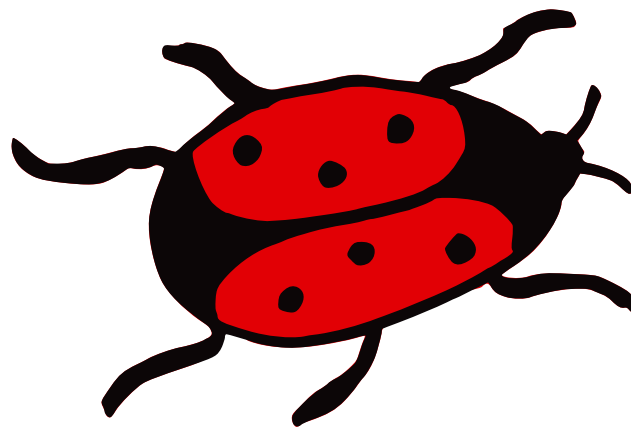
...noch ein Klassiker

„Thomas ist leider gerade
im Urlaub“

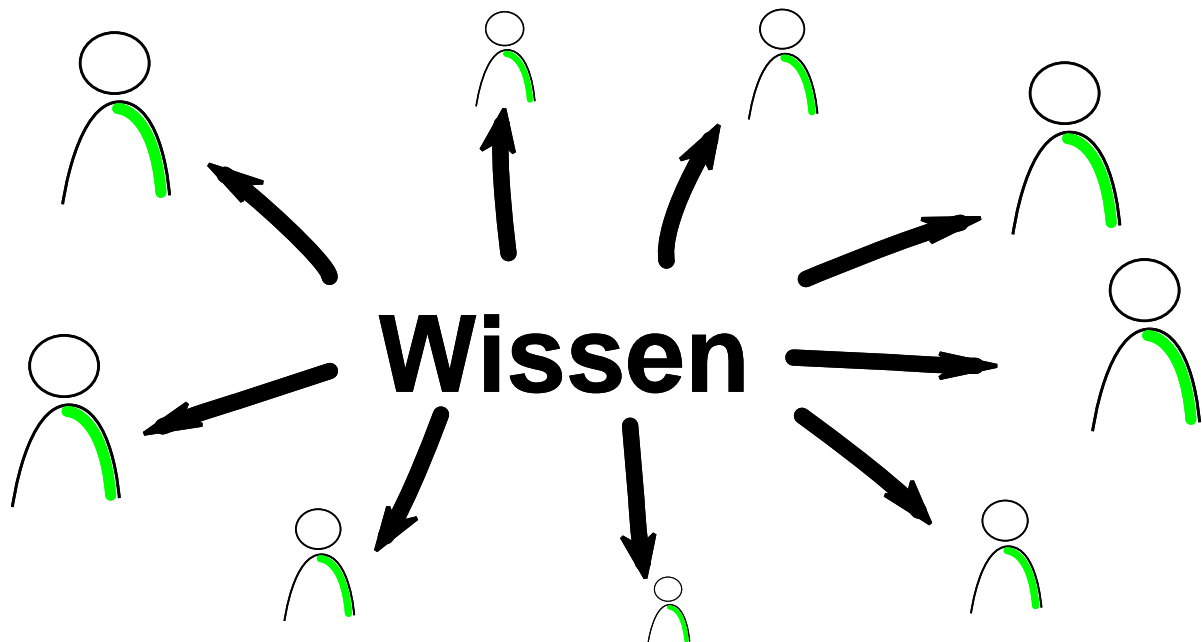
Es bleibt nie Zeit für das große
Refactoring

Code-Reviews

als kontinuierliche Lösung

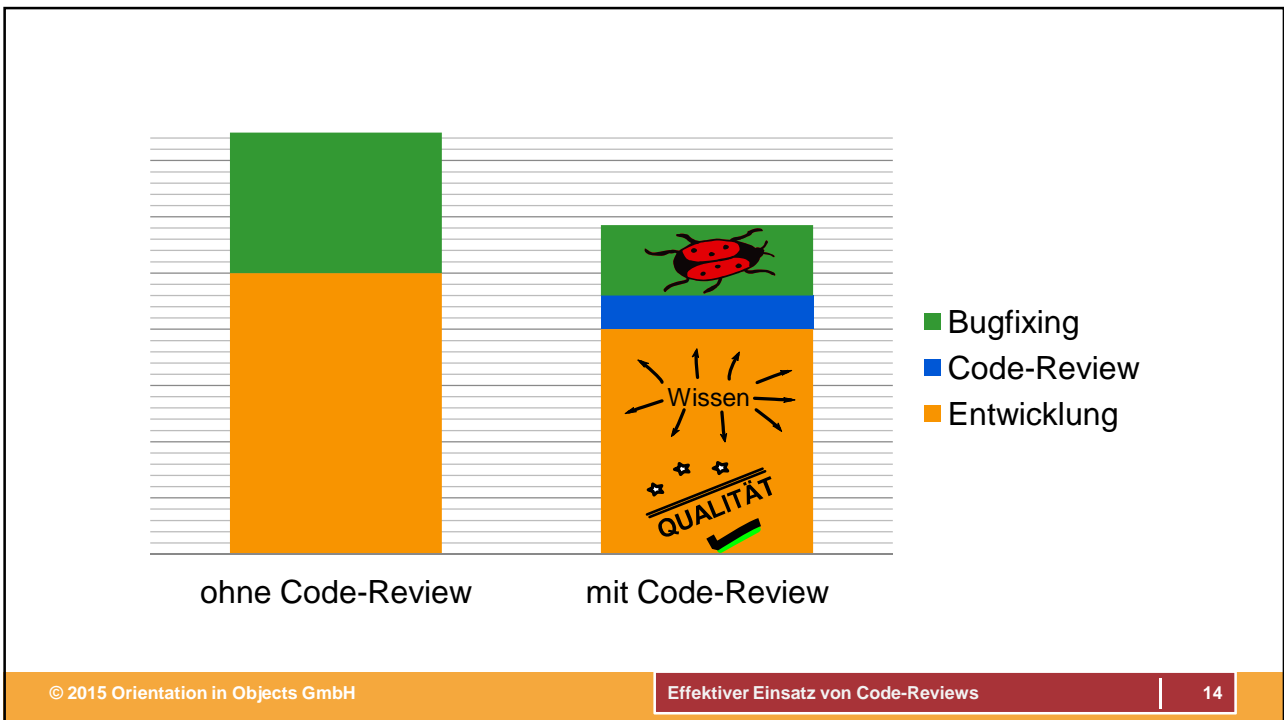
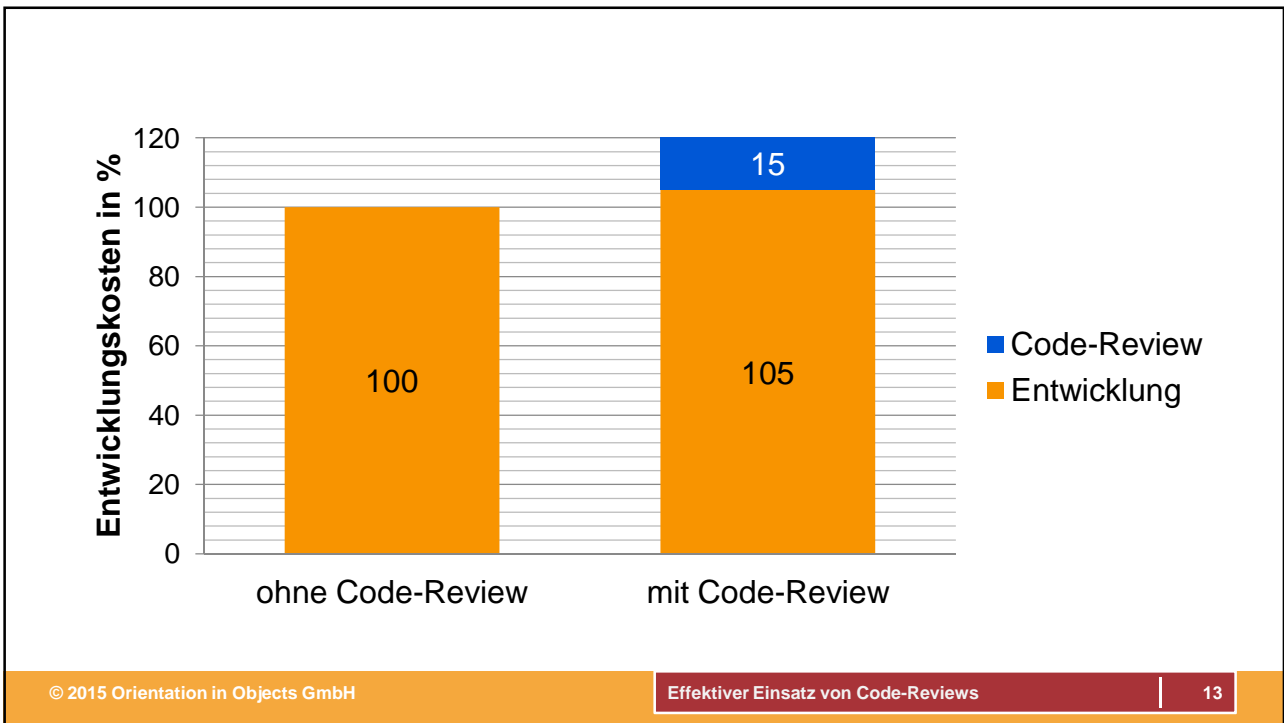


Bugs finden





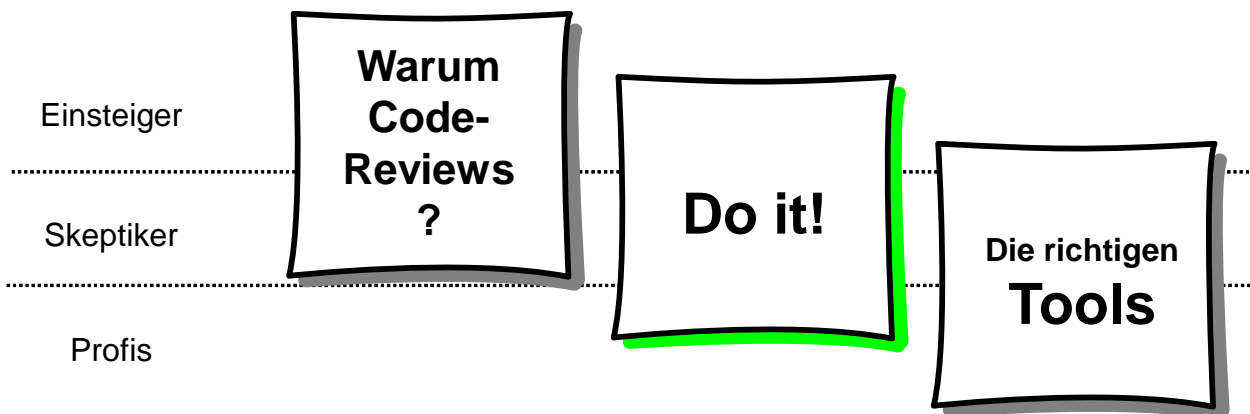
Anerkennung



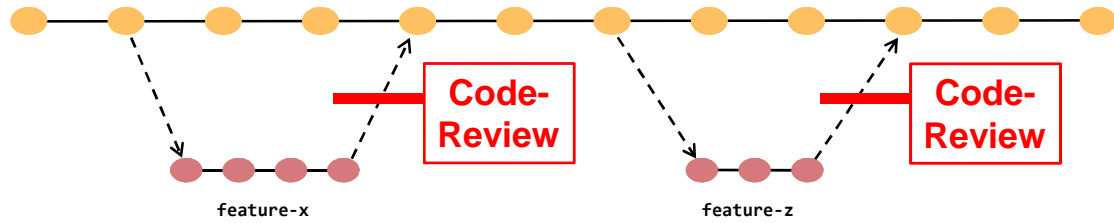
Code-Reviews für alle?

Projekt oder Produkt

Agenda

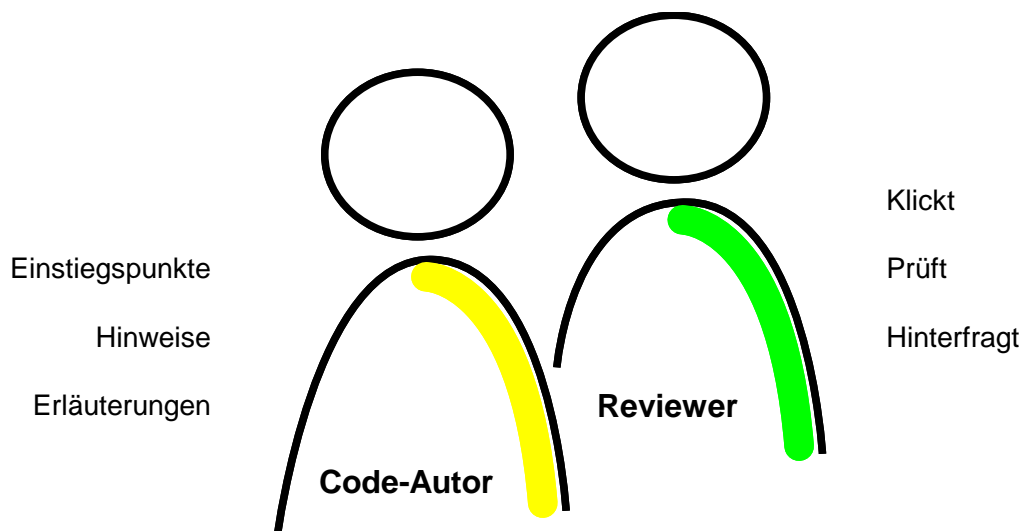


Wann?

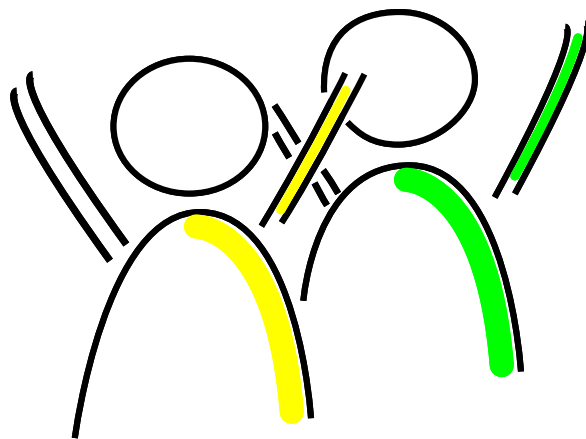


„Nach jedem Feature / Bugfix“

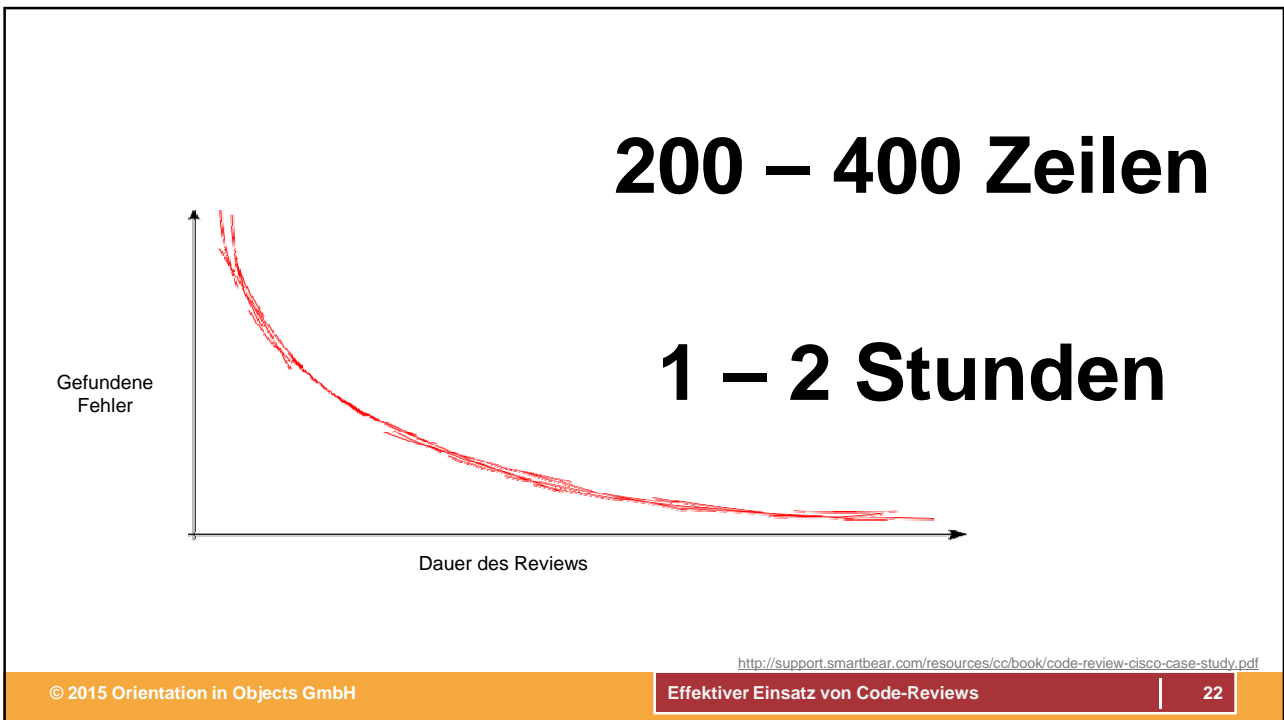
Wer?



Jeder mit jedem!



*„Wir wollen unsere Anwendung **gemeinsam** besser machen“*



Aufgaben aus Code-Review "Neue Benutzerverwaltung" #34

- Potentielle NullPointerException im UserService abfangen
- Typos in der Benutzer-Dokumentation entfernen
- Unit-Test "UserServiceTest" um den neuen Anwendungsfall erweitern
- Neues Widget für die Eingabe des Benutzernamens in Bibliothek auslagern
- Type note...

17:18

Story-Link	Autor	Reviewer	Zeit Impl	Zeit Review	Nachbearbeitung	~LOC	Anzahl Verbesserungen	CR war sinnvoll	Anzahl Bugs
EJFYLOLAINTERN-5010	SOS	TM	8h	1h	0h	100	0	ja	0
EJFYLOLAINTERN-5025	RK	TM	5h	1h	0h	50	1	ja	1
EJFYLOLAINTERN-5122	TM	FS	2h	30m	0h	10	0	ja	0
EJFYLOLAINTERN-4764	RK	TM	3h	1h	1h	20	1	ja	0
EJFYLOLAINTERN-5094	TM	SOS	4h 45m	30m	0h	50	0	ja	0
EJFYLOLAINTERN-5178	SOS	TM	3d5h	1h 30m	30m	300	1	ja	0
EJFYLOLAINTERN-5212	RK	FS	2h 45m	1h 15m	1h 15m	15	1	ja	1
EJFYLOLAINTERN-5174	SOS	TM	3h 45m	1h	15m	100	1	ja	0
EJFYLOLAINTERN-5227	TM	SOS	2h	30m	45m	30	1	ja	0
EJFYLOLAINTERN-5253	TM	SOS	1h	30m	0	20	0	ja	0
EJFYLOLAINTERN-5245	TM	SOS	1h 15m	30m	0	20	0	nein	0

- ✓ Sinnvolle Namen?
- ✓ Klare Struktur?
- ✓ Korrekte Verwendung von Datentypen?
- ✓ Kommentare?
- ✓ Code smells?
- ✓ Tests vorhanden?
- ✓ Wohldefinierte Schnittstellen?
- ✓ Exception-Handling?
- ✓ I18N?
- ✓ Typische Programmier-Fehler?



Agenda



Einsteiger

Skeptiker

Profis

**Warum
Code-
Reviews
?**

Do it!

**Die richtigen
Tools**

© 2015 Orientation in Objects GmbH

Effektiver Einsatz von Code-Reviews

27

Graph	Description	Date	Author	Commit
tsk624	Create LocalMDFCode for every SourceSystem (multi-create) + Update already existing er	12 Mrz 2015 13:10	Thorsten Maier <Thorsten.Maier@oio.de>	fcdb799
tsk602	Fixed search filter in ERP and GIS Supplier list view	11 Mrz 2015 18:15	Benjamin Barth <benjamin.barth@oio.de>	0983e84
tsk630	Add link from MDFCode to LocalMDFCode	11 Mrz 2015 16:55	Thorsten Maier <Thorsten.Maier@oio.de>	b336eab
tsk602	Fixed Unit Tests for GIS Suppliers	11 Mrz 2015 16:35	Benjamin Barth <benjamin.barth@oio.de>	938b0d4
tsk629	Edit LocalMDFCode	11 Mrz 2015 16:15	Thorsten Maier <Thorsten.Maier@oio.de>	f07207e
	Merge branch 'feature/sto31-suppliers'	11 Mrz 2015 16:01	Benjamin Barth <benjamin.barth@oio.de>	b2387a9
	origin/feature/sto31-suppliers	11 Mrz 2015 15:46	Benjamin Barth <benjamin.barth@oio.de>	00cef2c
	Added explicit logging configuration for "org.springframework.transaction"	11 Mrz 2015 11:46	Benjamin Barth <benjamin.barth@oio.de>	070cb88
	tsk602 - Added missing information to ERP and GIS Supplier history list view	10 Mrz 2015 18:29	Benjamin Barth <benjamin.barth@oio.de>	f48cb86

Filename	Path
MdfCode.java	src/main/java/com...domain
list.jsp	src/main/webapp...mdfCode

```

import javax.validation.constraints.Pattern;
import javax.validation.constraints.Size;
import org.hibernate.annotations.Formula;
import org.hibernate.validator.constraints.NotBlank;

@Entity
@hibernate.joins({
    @JoinColumn(name = "MaterialType_id")
})
private MaterialType materialType;

@Formula(value = "(select count(1.id) from MDF.LocalMdfCode l where l.materialType_id = id)")
private Integer numberOfLocalMdfCodes;

@Embedded
private ValidityInfos validityInfos;
    
```

Commit: b336eab274f86d3608893b43558d67688024be8b [b336eab]
 Parents: 938b0d49f6
 Author: Thorsten Maier <Thorsten.Maier@oio.de>
 Date: Mittwoch, 11. März 2015 16:55:02

File Status | Log / History | Search

Clean master Atlassian

© 2015 Orientation in Objects GmbH

Effektiver Einsatz von Code-Reviews

28

- ✓ Compile erfolgreich
- ✓ 356 Tests erfolgreich
- ✓ Checkstyle erfolgreich

```
3 public static void main(String[] args) {  
4     String name = (args.length > 0) ? args[0] : null;  
5  
6     if (name.equals("admin")) {  
7         System.out.println("Hello Administrator");  
8     }  
9 }  
10 }
```

```

3 public static void main(String[] args) {
4     String name = (args.length > 0) ? args[0] : null;
5
6     if (name.equals("admin")) {
7         System.out.println("Hello Administrator");
8     }
9 }
10 }

```

Pro... @ Java... Decl... Con... Bug ... Bug ... Bug ...

Main.java: 6

Navigation

Bug: Possible null pointer dereference of null in Main.main(String[])

There is a branch of statement that, *if executed*, guarantees that a null value will be dereferenced, which would generate a `NullPointerException` when the code is executed. Of course, the problem might be that the branch or statement is infeasible and that the null pointer exception can't ever be executed; deciding that is beyond the ability of FindBugs.



```

public class DemoServiceImpl implements DemoService {
    @Autowired
    private DemoController demoController;

    public void superServiceMethod() {
        demoController.doSomething();
    }
}

```



```

public class DemoServiceImpl implements DemoService {
    @Autowired
    private DemoController demoController;

    public void superServiceMethod() {
        demoController.doSomething();
    }
}

```



© 2015 Orientation in Objects GmbH

Effektiver Einsatz von Code-Reviews

33

```

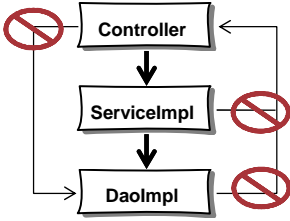
@RunWith(SpringJUnit4ClassRunner.class)
@SpringApplicationConfiguration(classes = DependencyTesterApplication.class)
public class DependencyTesterApplicationTests {

    @Autowired
    private DefaultListableBeanFactory beanFactory;

    @Test
    public void checkDependencies() {
        for (String bean : beanFactory.getBeanDefinitionNames()) {
            for (String dep : beanFactory.getDependenciesForBean(bean)) {
                checkNotAllowedConnection(bean, dep, ".*ServiceImpl", ".*Controller");
                checkNotAllowedConnection(bean, dep, ".*DaoImpl", ".*Controller");
                checkNotAllowedConnection(bean, dep, ".*Controller", ".*DaoImpl");
            }
        }
    }

    private void checkNotAllowedConnection(String beanName, String depName, String beanPattern, String depPattern) {
        assertFalse(beanName.matches(beanPattern) && depName.matches(depPattern));
    }
}

```

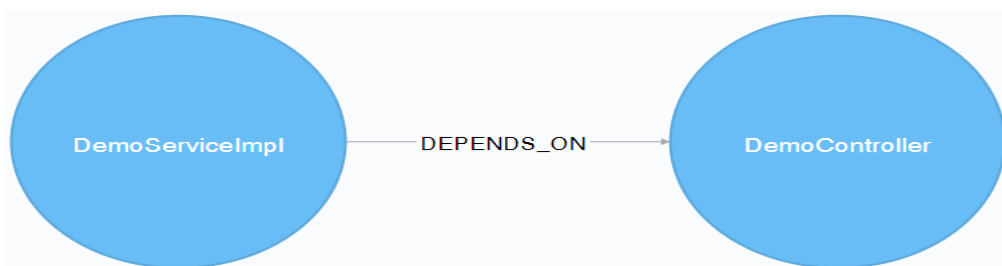


<https://github.com/thorstenmaier/architecture-layer-check/>

© 2015 Orientation in Objects GmbH

Effektiver Einsatz von Code-Reviews

34

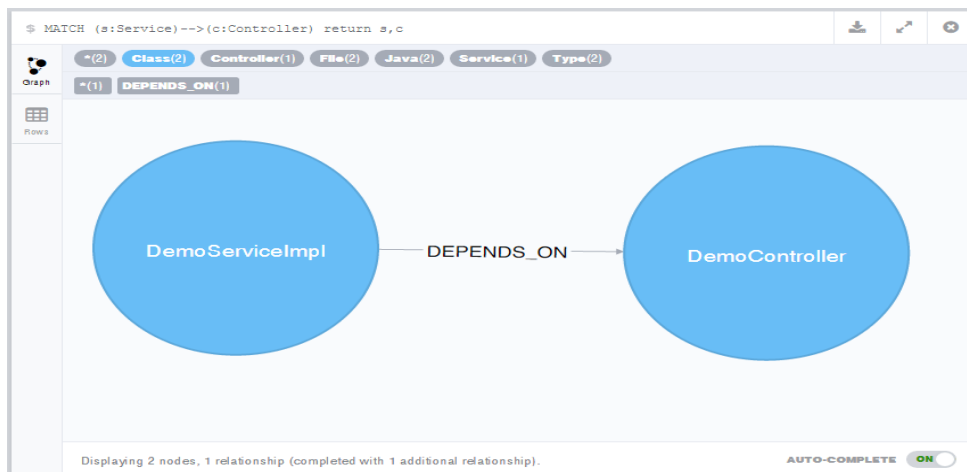


(Service) - [DEPENDS_ON] -> (Controller)



```
MATCH (s:Service)-[:DEPENDS_ON]->(c:Controller) return s,c
```

QAAssistant



- ✓ Compile erfolgreich
- ✓ 356 Tests erfolgreich
- ✓ Checkstyle erfolgreich

↓

Neue Aufgabe erstellen

Änderung auf Abnahme-System bereitstellen


© 2015 Orientation in Objects GmbH
Effektiver Einsatz von Code-Reviews
39

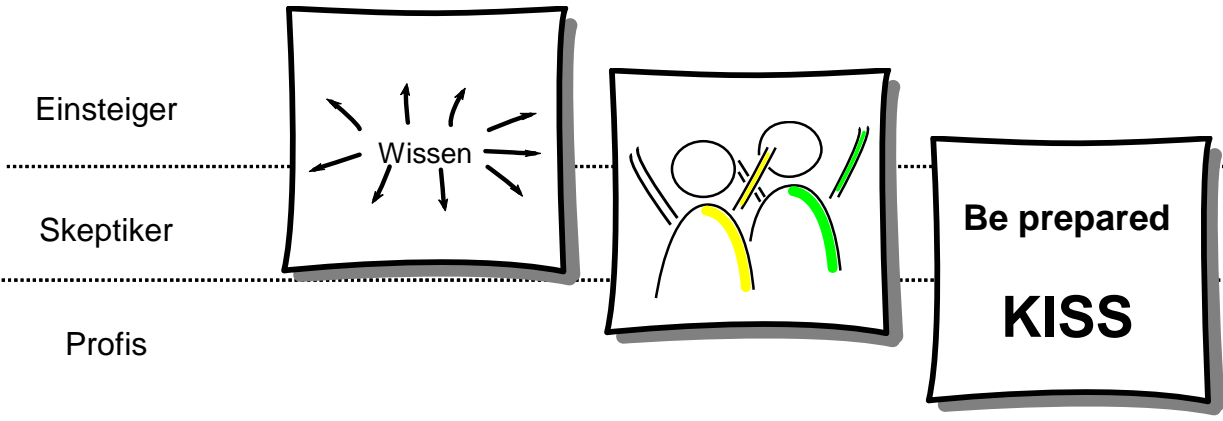
3b.

- ✓ Compile erfolgreich
- ✓ 356 Tests erfolgreich
- ✓ Checkstyle erfolgreich

© 2015 Orientation in Objects GmbH
Effektiver Einsatz von Code-Reviews
40

Fazit


Orientation in Objects




The diagram illustrates a progression through three stages of expertise, each represented by a box:

- Einsteiger** (Beginner): A box containing the word "Wissen" (Knowledge) with several arrows pointing outwards, symbolizing initial learning and exploration.
- Skeptiker** (Skeptic): A box containing two stick figures, one yellow and one green, with arrows pointing towards each other, symbolizing collaboration and questioning.
- Profis** (Professionals): A box containing the text "Be prepared" and "KISS" (Keep It Simple, Stupid), symbolizing the final stage of expertise where simplicity and preparedness are key.

© 2015 Orientation in Objects GmbH | Effektiver Einsatz von Code-Reviews | 41

Mehr von OIO zum Thema...


Orientation in Objects

- Schulung: Refactoring Workshop
 - <http://www.oio.de/seminar/methodik-prozess-management-soft-skills/training-java-kurs-refactoring-schulung.htm>
- Schulung: Git kompakt
 - <http://www.oio.de/git-kompakt-schulung-versionsverwaltung-seminar-dvcs-training.htm>
- Schulung: Versionsverwaltung mit Git
 - <http://www.oio.de/workflows-schulung-git-seminar-dvcs-training.htm>
- Schulung: Atlassian Stash – Enterprise Git
 - <http://www.oio.de/seminar/methodik-prozess-management-soft-skills/seminar-stash-schulung-atlassian-enterprise-training-git-kurs.htm>
- Schulung: JIRA – Fachliche Administration
 - <http://www.oio.de/seminar/methodik-prozess-management-soft-skills/seminar-training-atlassian-jira-schulung.htm>

© 2015 Orientation in Objects GmbH | Effektiver Einsatz von Code-Reviews | 42



Fragen ?

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de



**Vielen Dank für ihre
Aufmerksamkeit !**

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de