



Code Quality in Practice

Entwicklertag Karlsruhe
Mai 2017

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Thorsten Maier

Trainer, Berater, Entwickler



 [@ThorstenMaier](#)

```
protected void extractRow(List<MdfSaving> validList, List<MdfSaving> invalidList, ValueMap valueMap, int rowNumber, User loggedInUser) {
    Integer currentYear = valueMap.getInteger("currentyear");
    if (currentYear == null) {
        currentYear = java.time.YearMonth.now().getYear();
    }

    String inputComment = valueMap.getString("notes", "comment");
    String inputPrincipalname = valueMap.getString("user", "username", "csuser", "csusername", "mail", "email");
    String inputOrganizationCode = valueMap.getString("lbu", "lpgu", "lpgureportingunit", "reportingunit", "organization", "organizationunit", "abbunit");
    String inputMdfCode = valueMap.getString("mdf", "mdfcode", "material");
    String inputErpSupplierCode = valueMap.getString("supplier", "erpsupplier", "localsupplier");
    String actualOrForecast = "0".equals(valueMap.getString("flagactualforecast")) ? "Forecast" : "Actual";
    String inputSupplierType = valueMap.getString("analexternalspend", "internalexternalspend", "internalexternalspend");
    String inputSavingType = valueMap.getString("savingtype", "savingtype");
    String inputChessboardStrategy = valueMap.getString("chessboardstrategy", "chessboardstrategy");
    Integer inputYearMonth = valueMap.getInteger("yearmonth", "yearmonth");
    BigDecimal inputForecastProbability = valueMap.getBigDecimal("forecastprobability", "forecastprobability");
    BigDecimal inputSavingValue = valueMap.getBigDecimal("savingvalue", "savingvalue");
    Boolean isCapex = getCapexFlag(valueMap);
    Map<String, String> uploadedStringValues = new HashMap<>();

    MdfCode mdfCode = mdfCodeRepository.findById(inputMdfCode);

    Supplier supplier = null;
    if (organization != null) {
        supplier = supplierRepository.findSupplierByCodeAndSourceSystem(inputErpSupplierCode, organization.getSourceSystem());
    }
    SupplierType supplierType = supplierTypeRepository.findByCode(inputSupplierType);

    SavingRationale savingRationale = getSavingRationale(valueMap);
    Currency usd = currencyRepository.findByCode("USD");
    Currency localReportingCurrency = (organization != null) ? organization.getCurrency() : null;
    ChessBoardStrategy chessBoardStrategy = chessBoardStrategyRepository.findByCode(inputChessboardStrategy);
    SavingType savingTypeCA = savingTypeRepository.findByCode("CA " + actualOrForecast);
    Map<Integer, BigDecimal> yearMonthToCAMap = getYearMonthValueMap(valueMap, currentYear, "currentyear", "yearmonth");
    yearMonthToCAMap.putAll(yearMonthValueMapFromCurrenters(valueMap, currentYear, "currentyear", "yearmonth"));
    List<MdfSaving> importedMdfSavings = createMdfSavings(yearMonthToCAMap, inputErpSupplierCode, supplier, savingRationale, savingRationale, localReportingCurrency, savingType, distribution, uploadedStringValues, isCapex, chessBoardStrategy);

    SavingType savingType = savingTypeRepository.findByCode(inputSavingType + actualOrForecast);
    Map<Integer, BigDecimal> yearMonthToCRMap = getYearMonthValueMap(valueMap, currentYear, "currentyear", "yearmonth");
    yearMonthToCRMap.putAll(yearMonthValueMapFromCurrenters(valueMap, currentYear, "currentyear", "yearmonth"));
    importedMdfSavings.add(createMdfSavings(yearMonthToCRMap, internalexternalspend, savingRationale, savingRationale, savingRationale, localReportingCurrency, savingTypeCR, distribution, uploadedStringValues, isCapex, chessBoardStrategy));
    if (!StringUtils.isEmpty(inputSavingType) && (inputYearMonth != null) && (inputSavingValue != null)) {
        SavingType savingType = savingTypeRepository.findByCode(inputSavingType);
        importedMdfSavings.add(new MdfSaving(inputSavingValue, interaction, inputForecastProbability, initiative, mdfCode, supplier, supplierType,
            savingRationale, usd, localReportingCurrency, savingType, distribution, uploadedStringValues, inputYearMonth, isCapex, chessBoardStrategy));
    }
    for (MdfSaving importedMdfSaving : importedMdfSavings) {
        if (importedMdfSaving.getSavingValue().compareTo(BigDecimal.ZERO) != 0) {
            (importedMdfSaving.validate() ? validList : invalidList).add(importedMdfSaving);
        }
    }
}
```

Code

Quality

TEAM

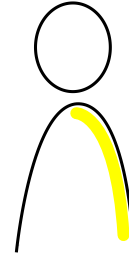
JOBTITEL-BANDIT



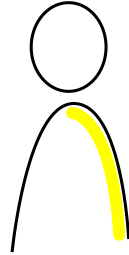
<http://www.spiegel.de/karriere/games/jobtitel-generator-neue-berufsbezeichnungen-fuer-angeber-und-aufschneider-a-771682.html>

*„Ein gut funktionierendes Team aus **adäquaten Mitarbeitern** wird ein Projekt beinahe unabhängig vom Verfahren und der Technologie [...] fertig stellen.“*

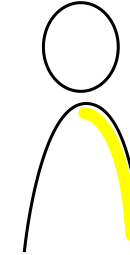
Alistair Cockburn (Agiles Manifest)



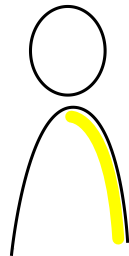
Der Gute-Laune-Macher



Der Kommunikator



Der Schaffer



**Das Organisationstalent
und der Realist**



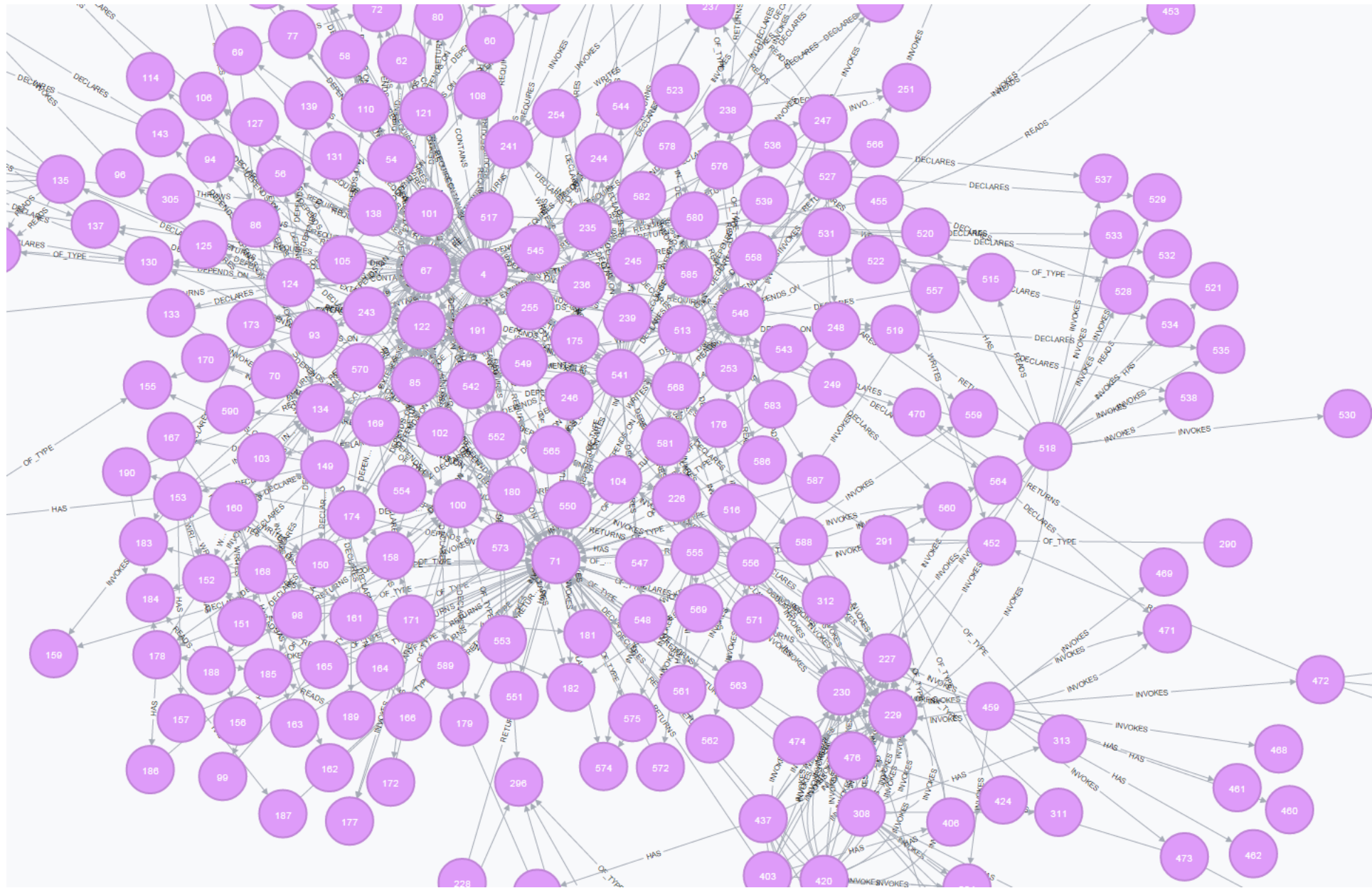
**Der Technikfreak
und Visionär**

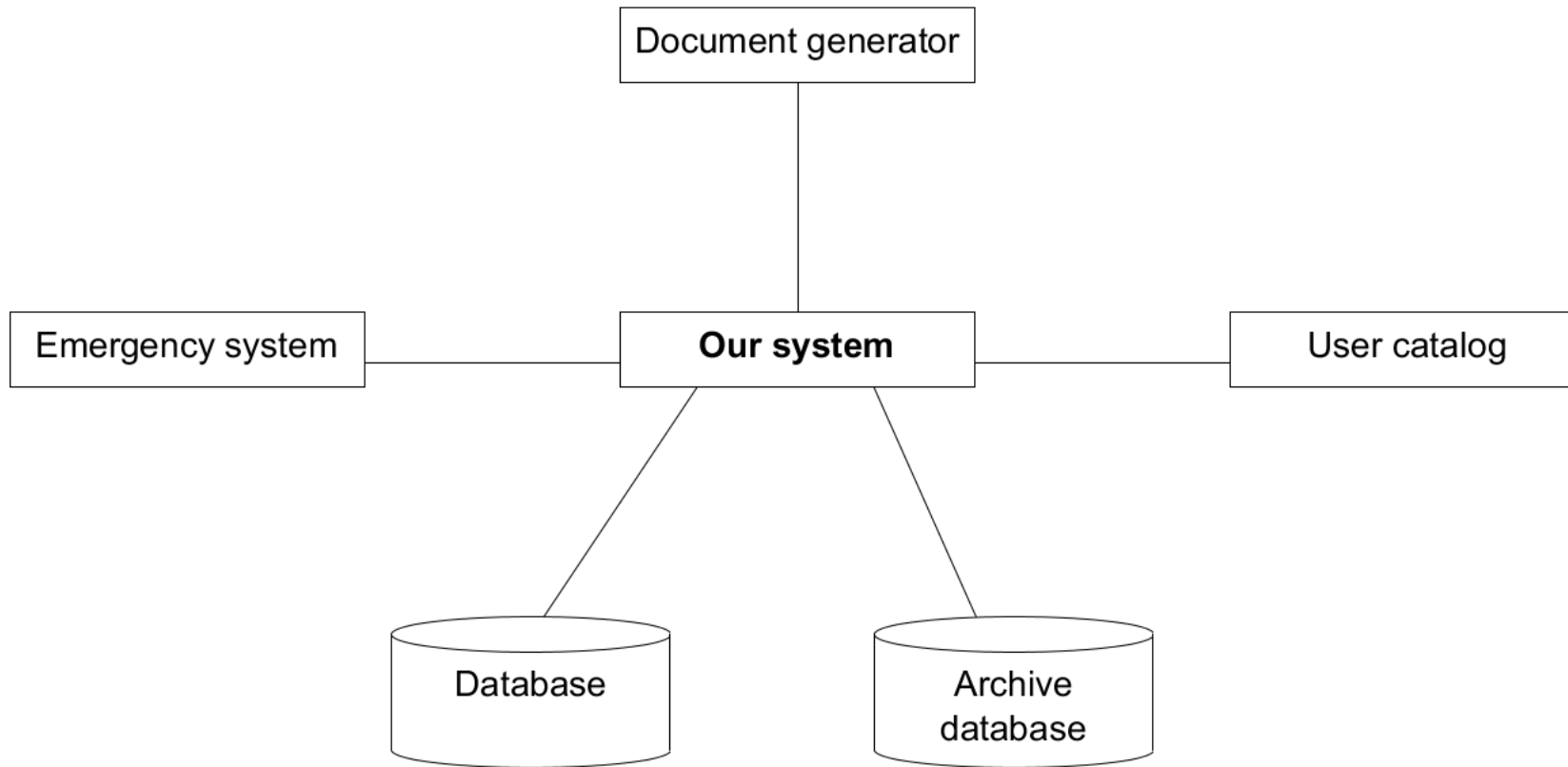
ARCHITEKTUR

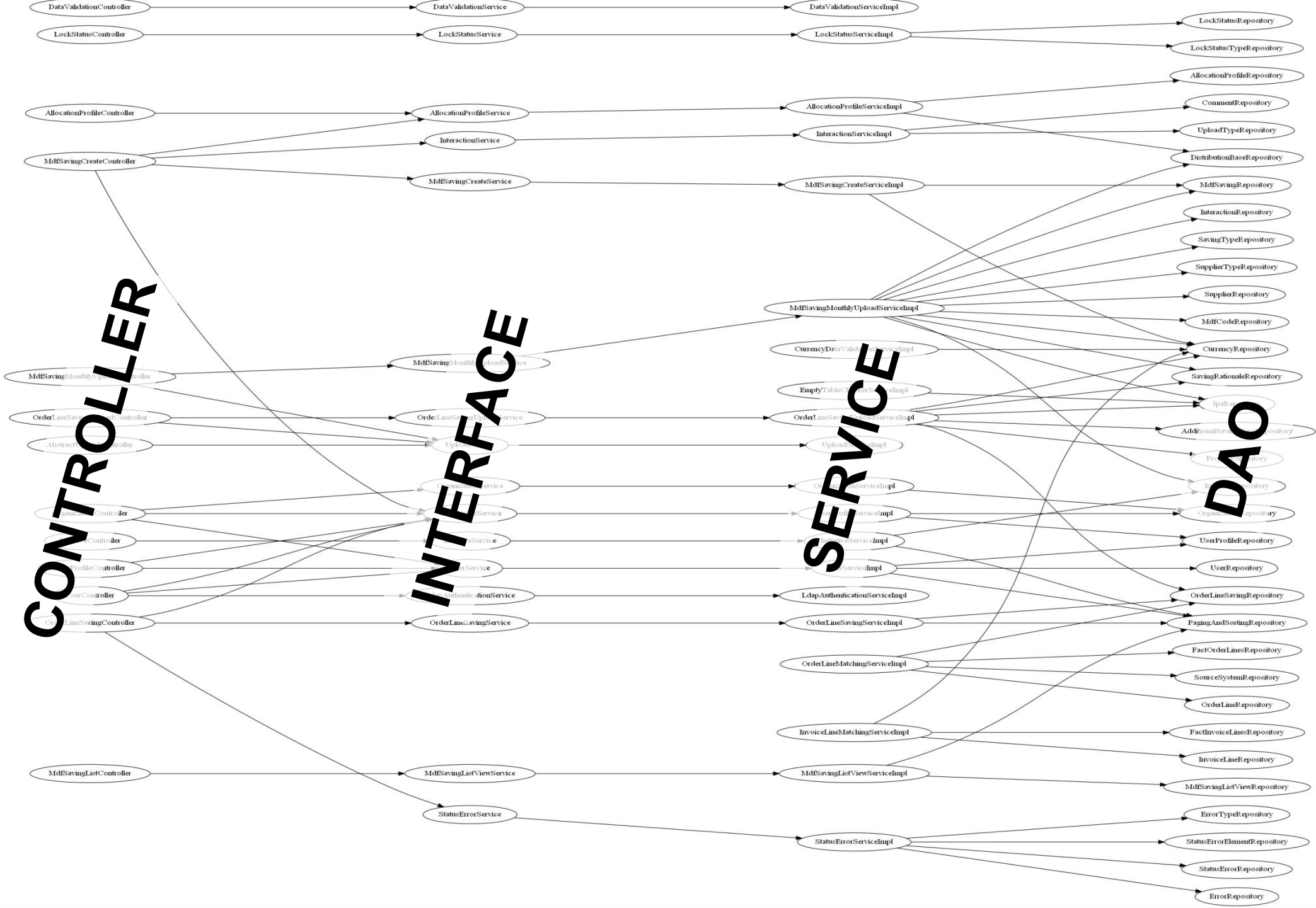
Architektur

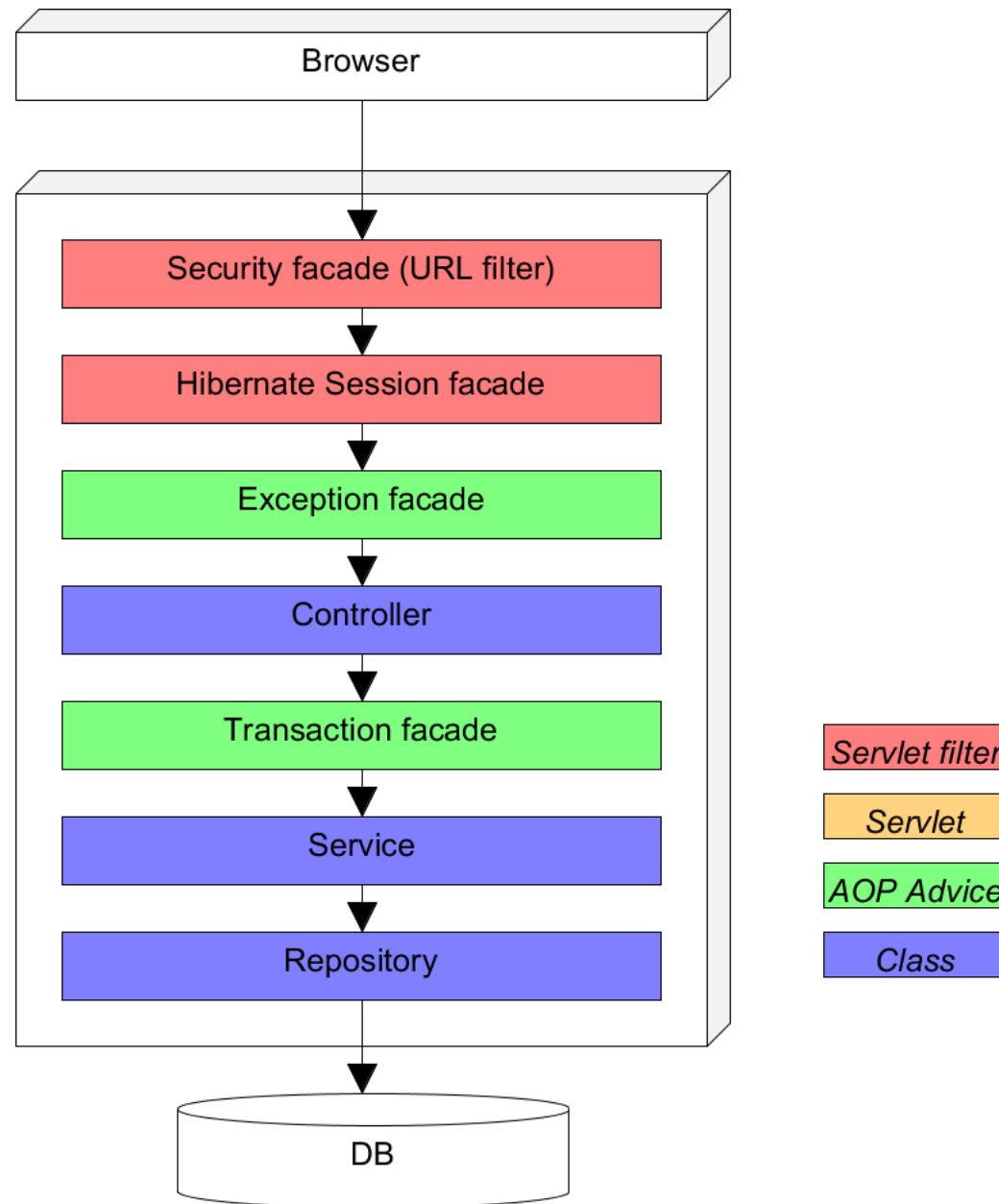
Fundamentale Strukturen, Konzepte, Entscheidungen und Lösungsansätze

... die man nicht mehr leicht los bekommt!









Theorie:

Architekturen früh vergleichbar

Praxis:

***Analyse wird während Schadensbegrenzung
spät im Projekt durchgeführt***

Szenariobasierte Architekturbewertung

SAAM – Software architecture analysis method

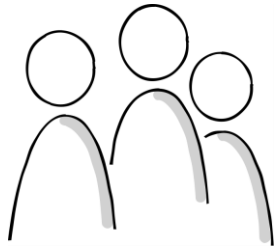
ATAM – Architecture tradeoff analysis method

ACDM – Architecture-centric design method

(1) Szenarien erheben und priorisieren

(2) Erstellen der Architektur(en)

(3) Bewertung aus dem Blickwinkel der Szenarien

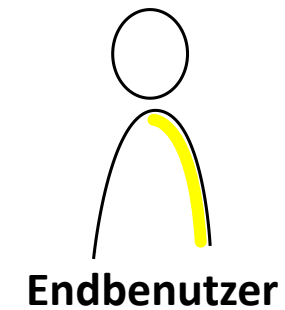
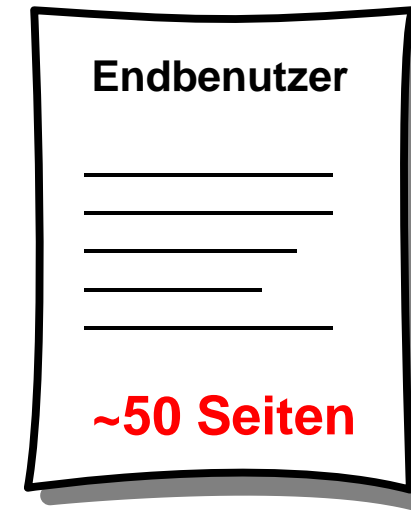
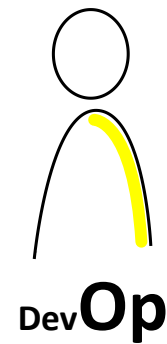
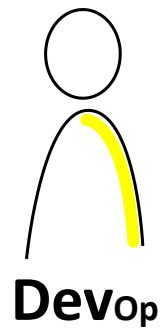


Stakeholder

	Architektur 1		Architektur 2	
	Komponenten	Aufwand	Komponenten	Aufwand
Neues Krankenversicherungsprodukt	3	28d	2	20d
DHL statt Hermes	-	-	4	12d
...				
Ergebnis	3	28d	6	32d

DOKU

Seite: 189 von 551





```

677 661 | jstl | 1.2 | Unknown license
678 662 | jTDS | 1.2.7 | LGPL
679 663 | JUnit | 4.11 | Common Public License Version 1.0
680 - | Liquibase Core | 3.3.2 | Apache License, Version 2.0
681 664 | mchange-commons-java | 0.2.9 | Eclipse Public License, Version 1.0, GNU Lesser
682 665 | Mockito | 1.10.8 | The MIT License
683 666 | MySQL java connector | 5.1.34 | The GNU General Public License, Version 2

```

```

803 786 | db.password | The database password
804 787 | db.hibernate.dialect | The Hibernate database dialect
805 788 | db.hibernate.show_sql | Show SQL statements in log file
806 - | db.liquibase.shouldRun | Liquibase is used for automatically updating the database schema. Liqui
807 789 | security.developerlogin.enabled | Enables the login **without** password.
808 790 | ldap.url | The LDAP url
809 791 | ldap.root | The LDAP root path of the user management

```

Überschrift 1

=====

Dies ist ein Text. Es folgt eine Aufzählung:

- Punkt 1
- *Kursiver* Punkt 2
- **Fetter** Punkt 3

Überschrift 1.1

Spalte 1	Spalte 2
Zeile 1	...
Zeile 2	...
...	...

Überschrift 1

Dies ist ein Text. Es folgt eine Aufzählung:

- Punkt 1
- *Kursiver* Punkt 2
- **Fetter** Punkt 3

Überschrift 1.1

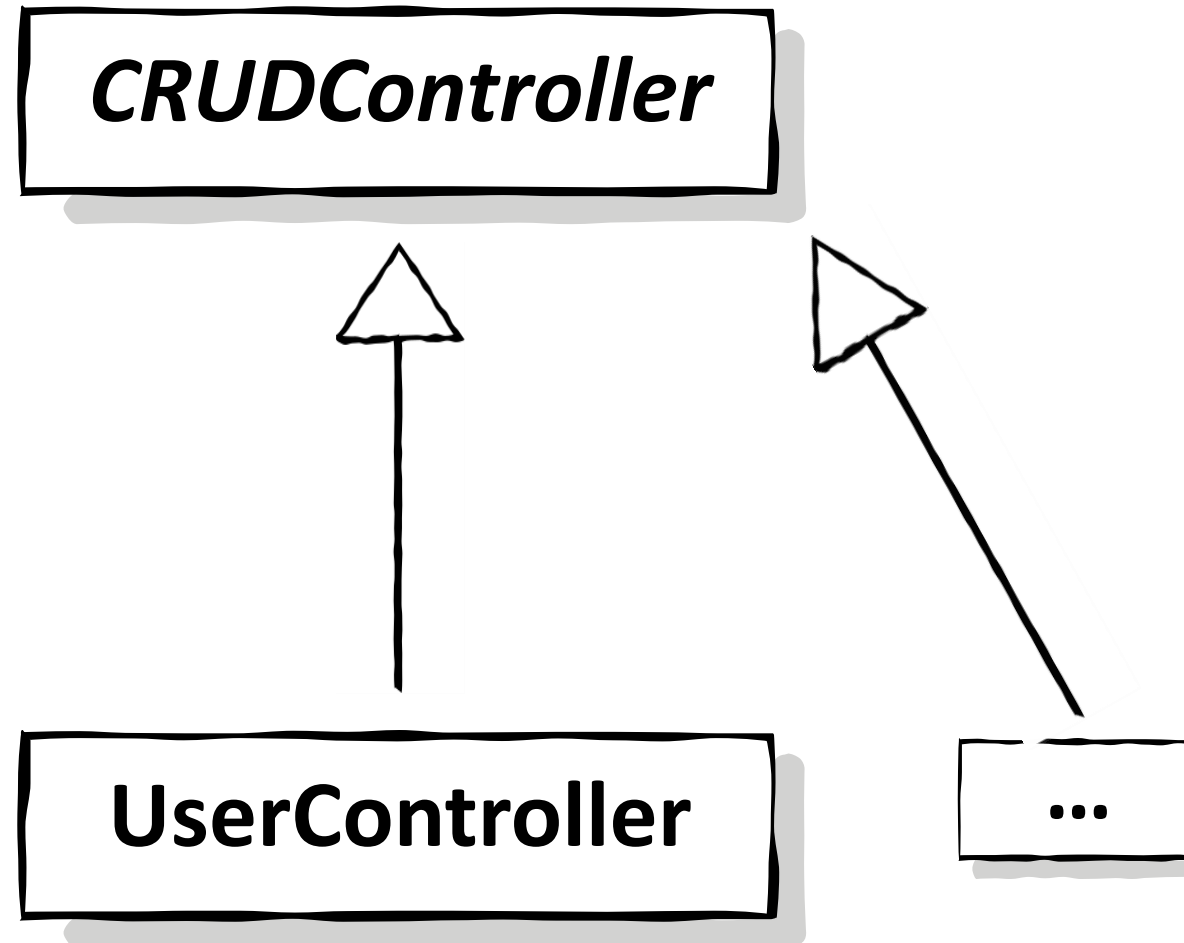
Spalte 1	Spalte 2
Zeile 1	...
Zeile 2	...
...	...

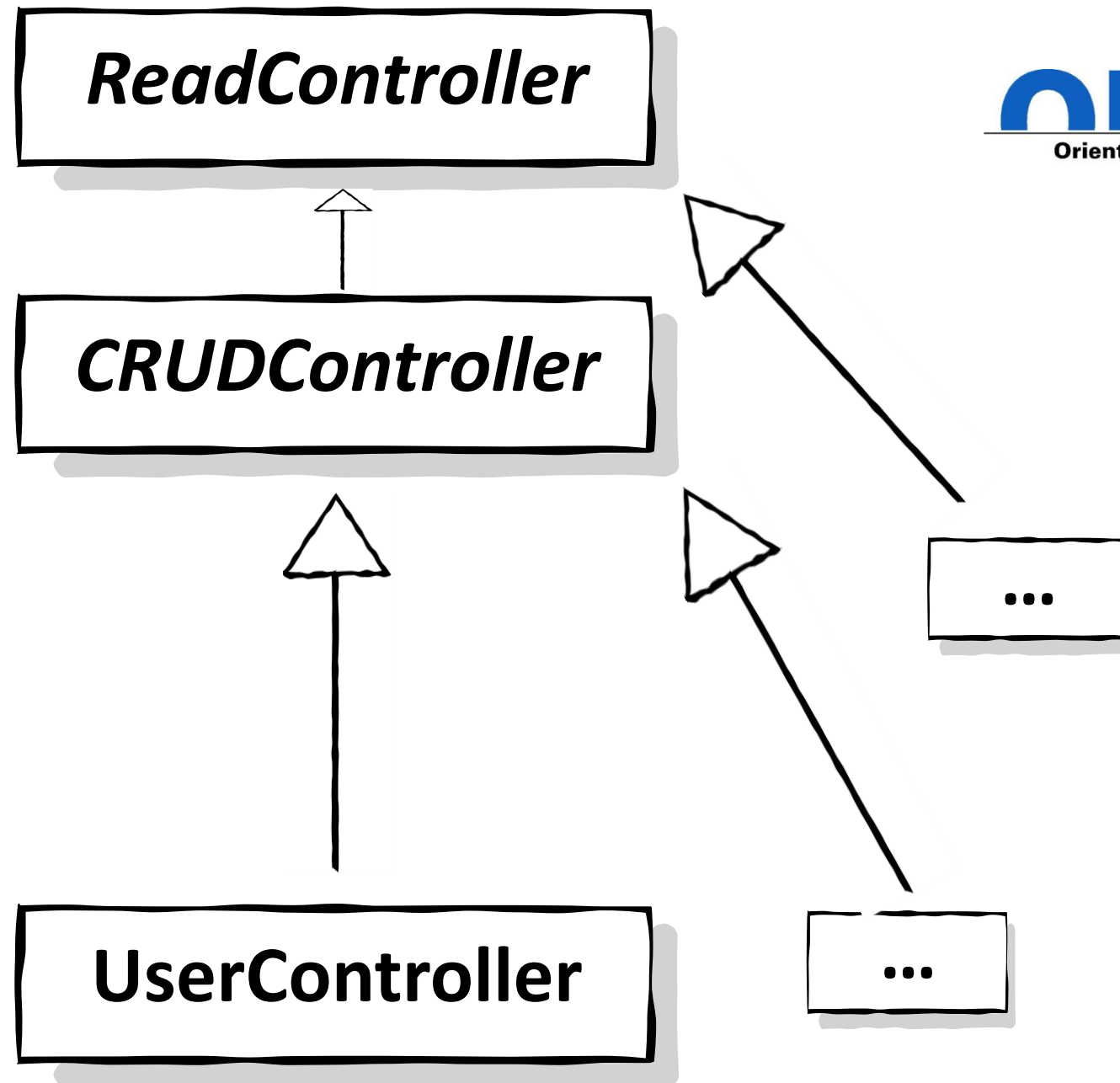


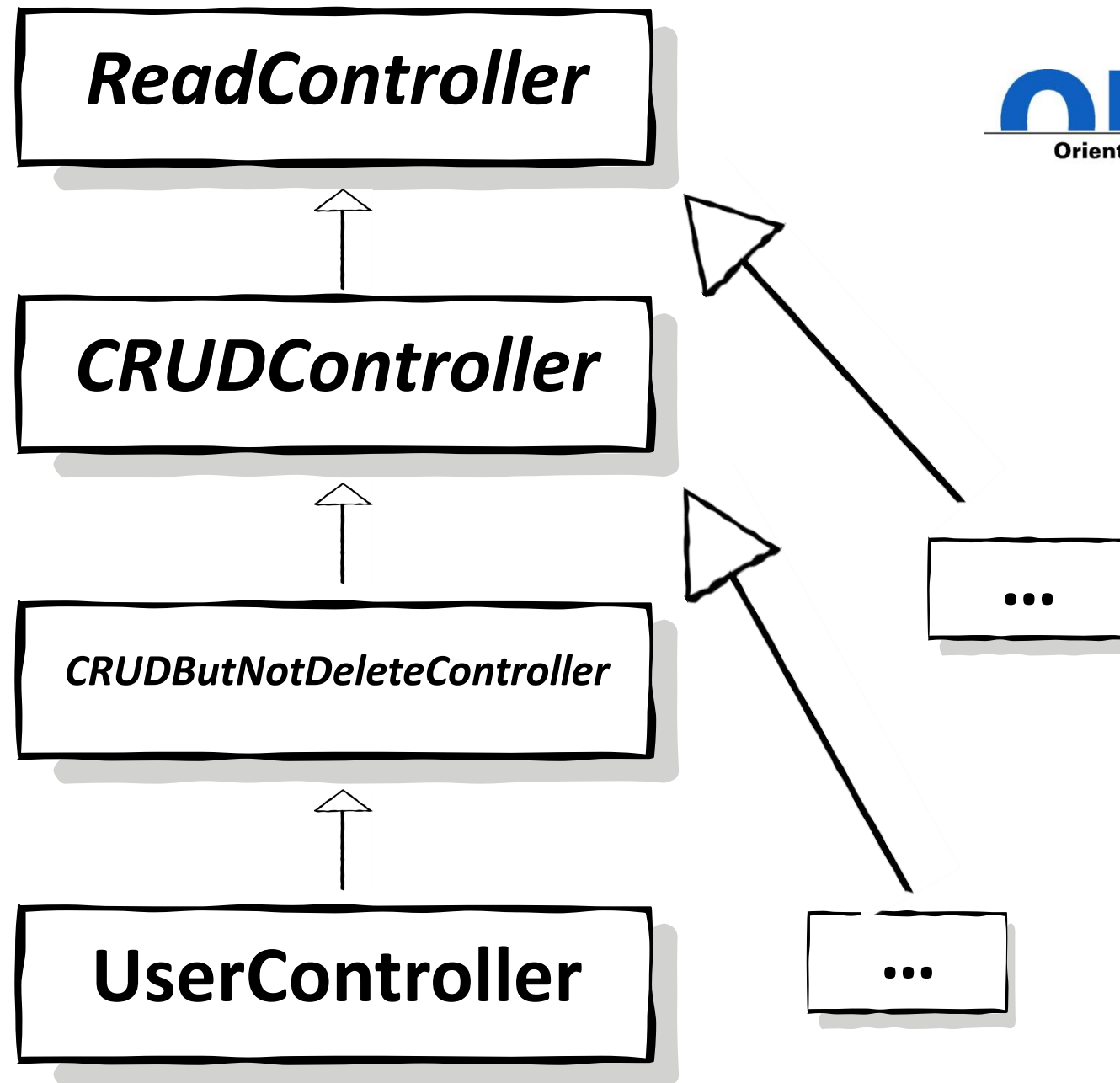
```
[Company] <>-1> [Location]
[Location] +-> [Point]
```

CODE

UserController







CreateController



UserCreateController

ReadController



UserReadController

UpdateController



UserUpdateController

Testarten



**Unit
Test**

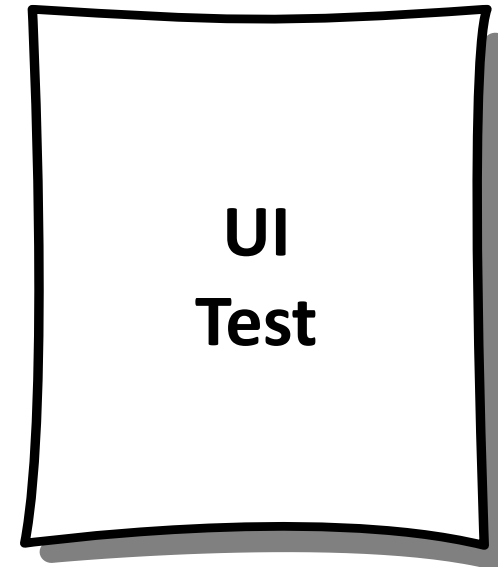
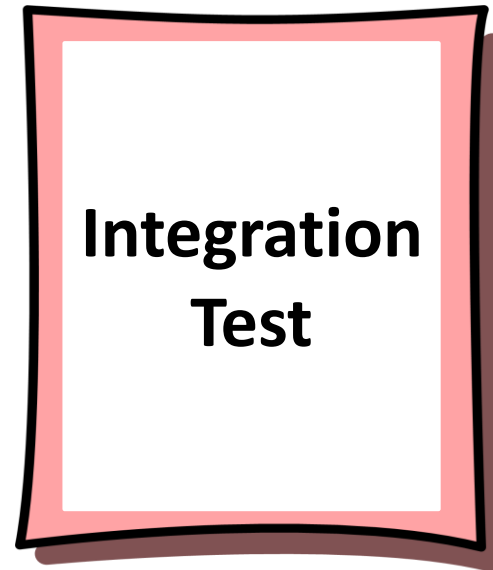
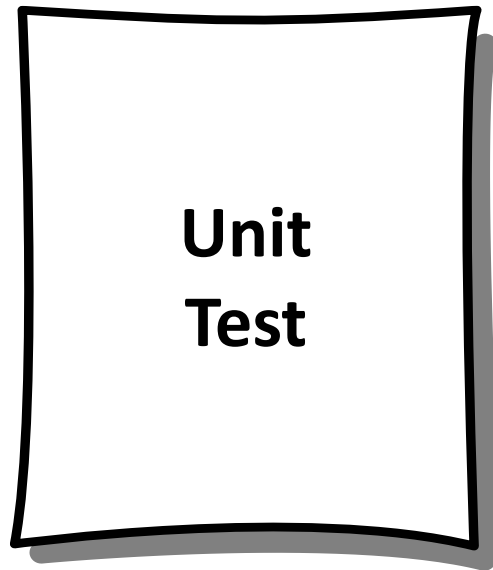


**Integration
Test**



**UI
Test**

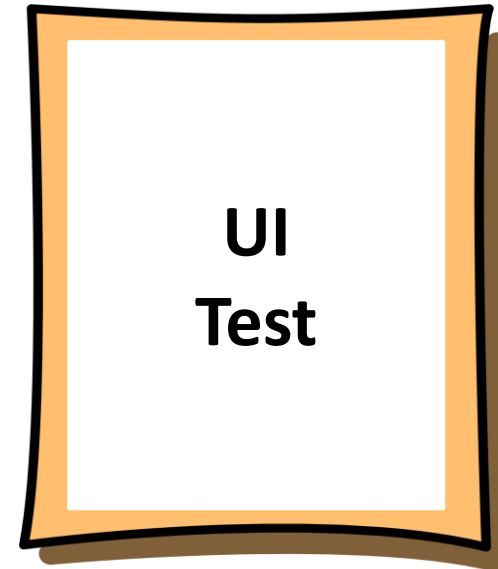
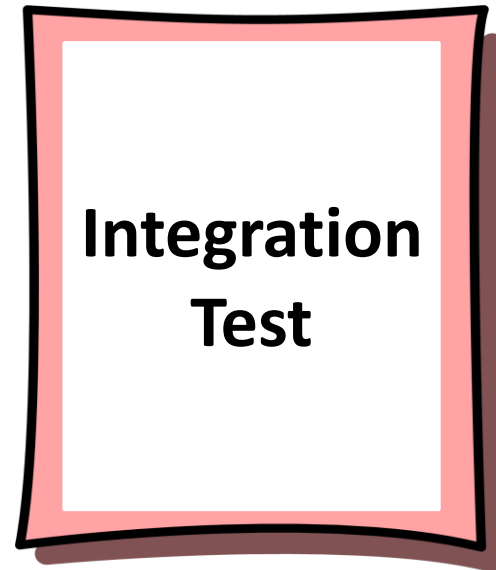
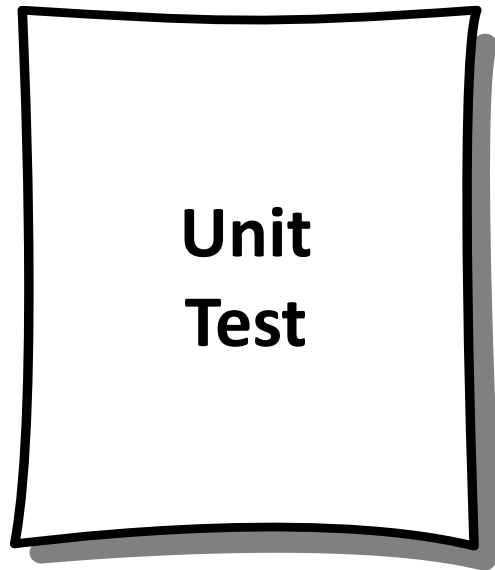
Testarten



🕒 30s+

„Funktioniert mein Mock?“

Testarten



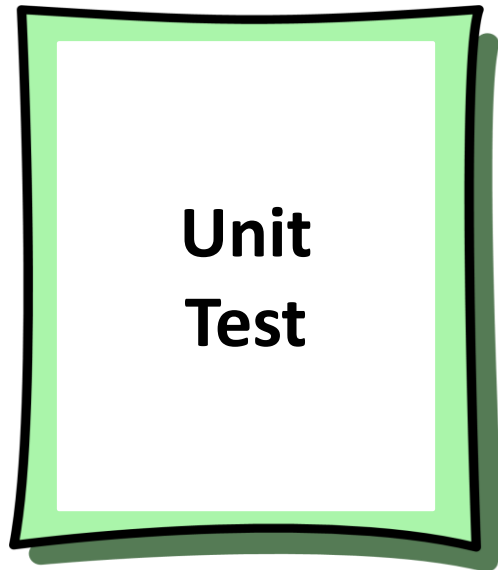
🕒 30s+

„Funktioniert mein Mock?“

🕒 30s+

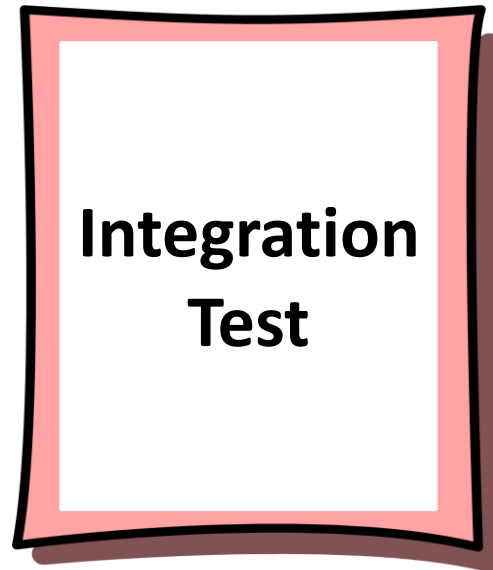
„Das sieht der User“

Testarten



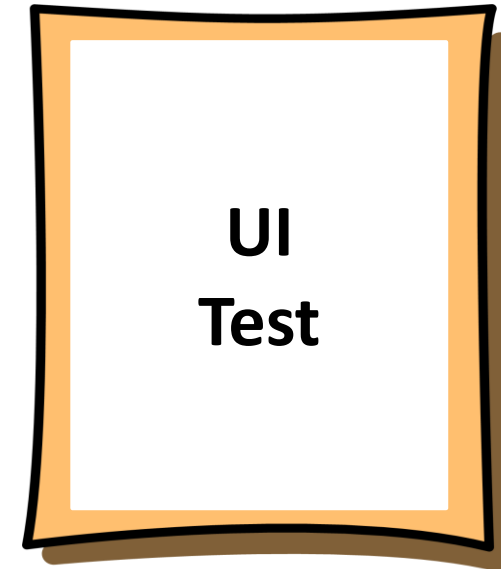
🕒 1s

Aussage über Codequalität!



🕒 30s+

„Funktioniert mein Mock?“



🕒 30s+

„Das sieht der User“

REVIEWS

Wer kennt das?

**„Das ist historisch
gewachsen.“**

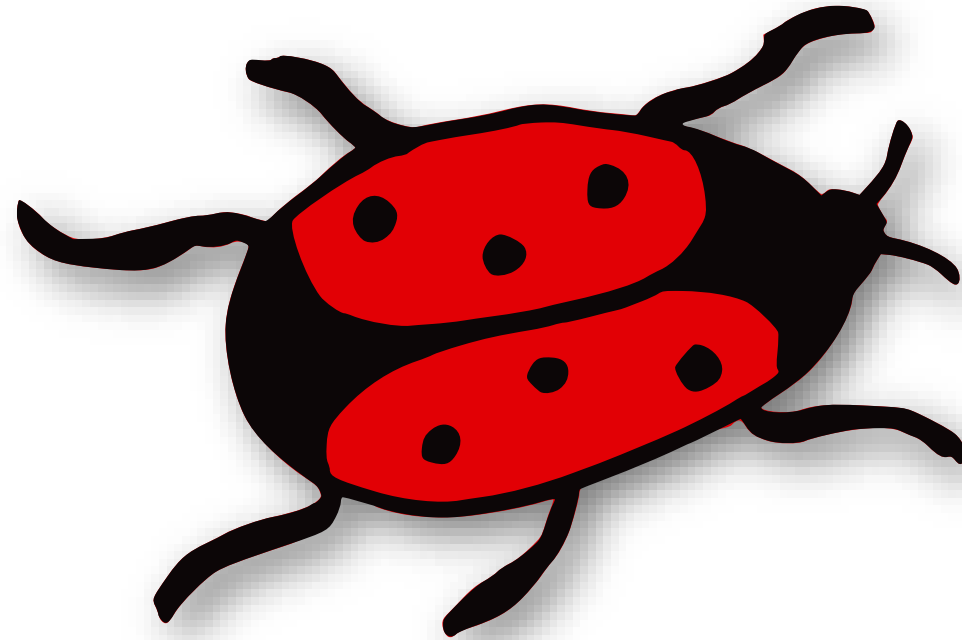
...noch ein Klassiker

**„Thomas ist leider gerade
im Urlaub“**

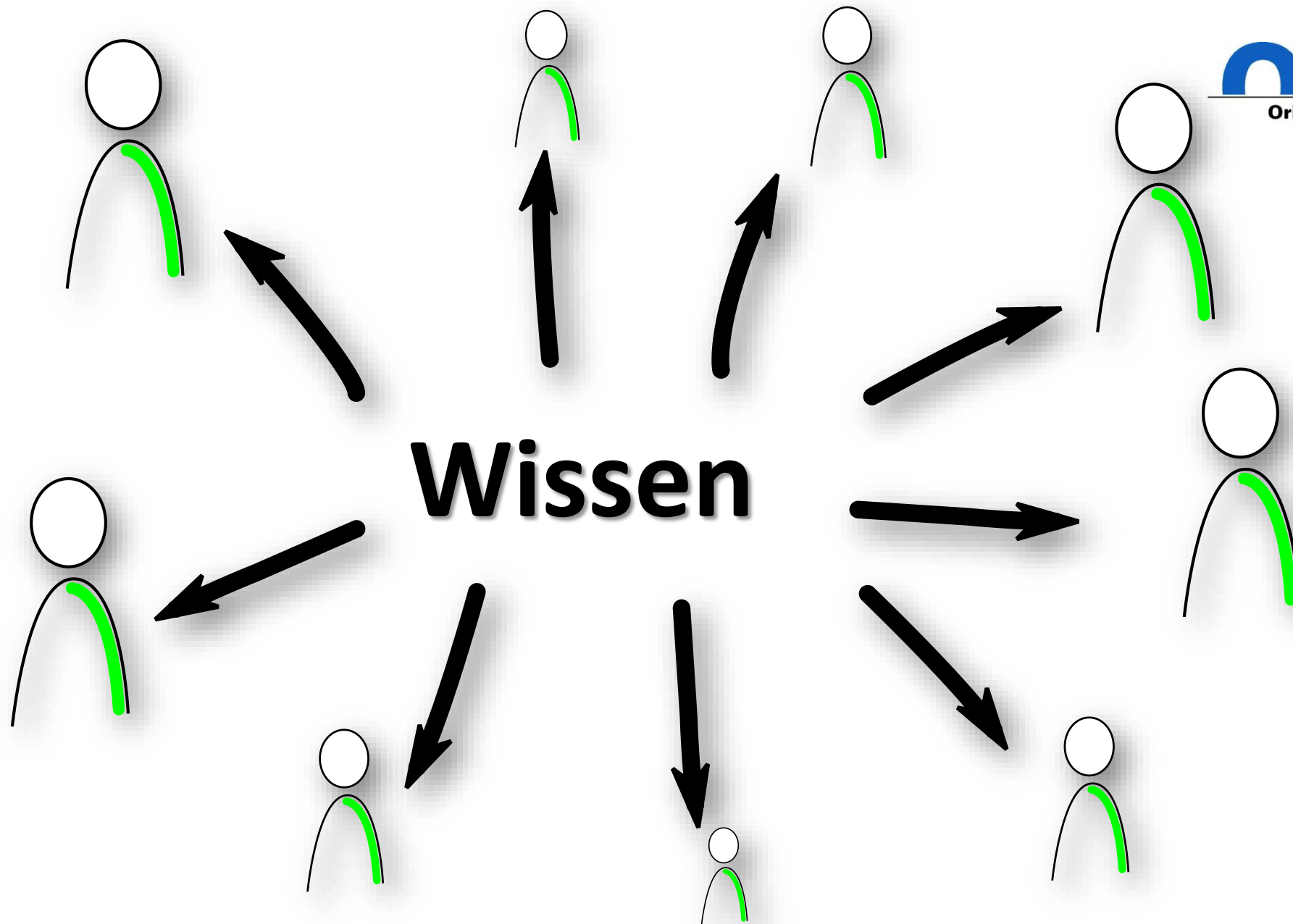
Es bleibt nie Zeit für das große
Refactoring

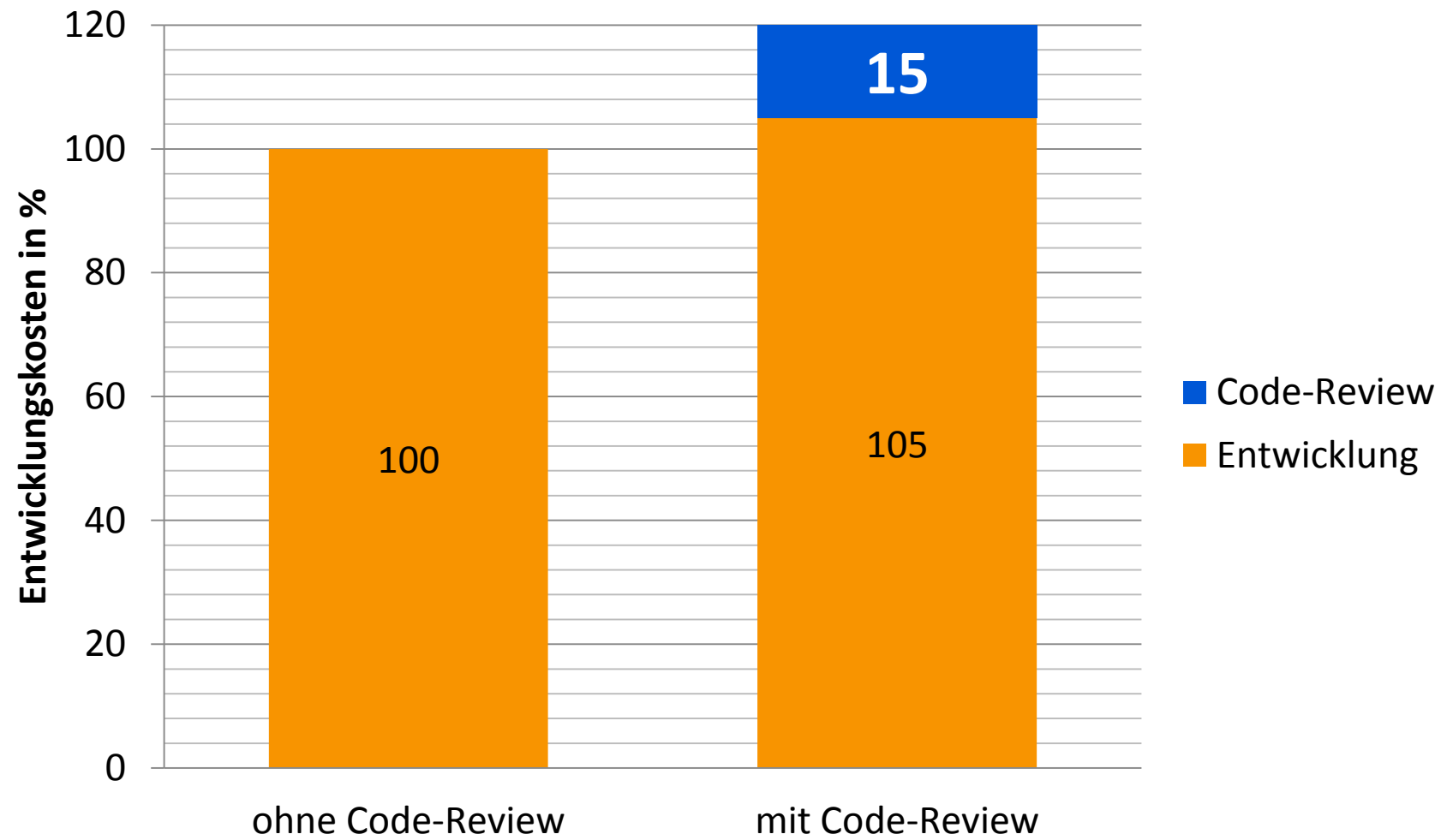
Code-Reviews

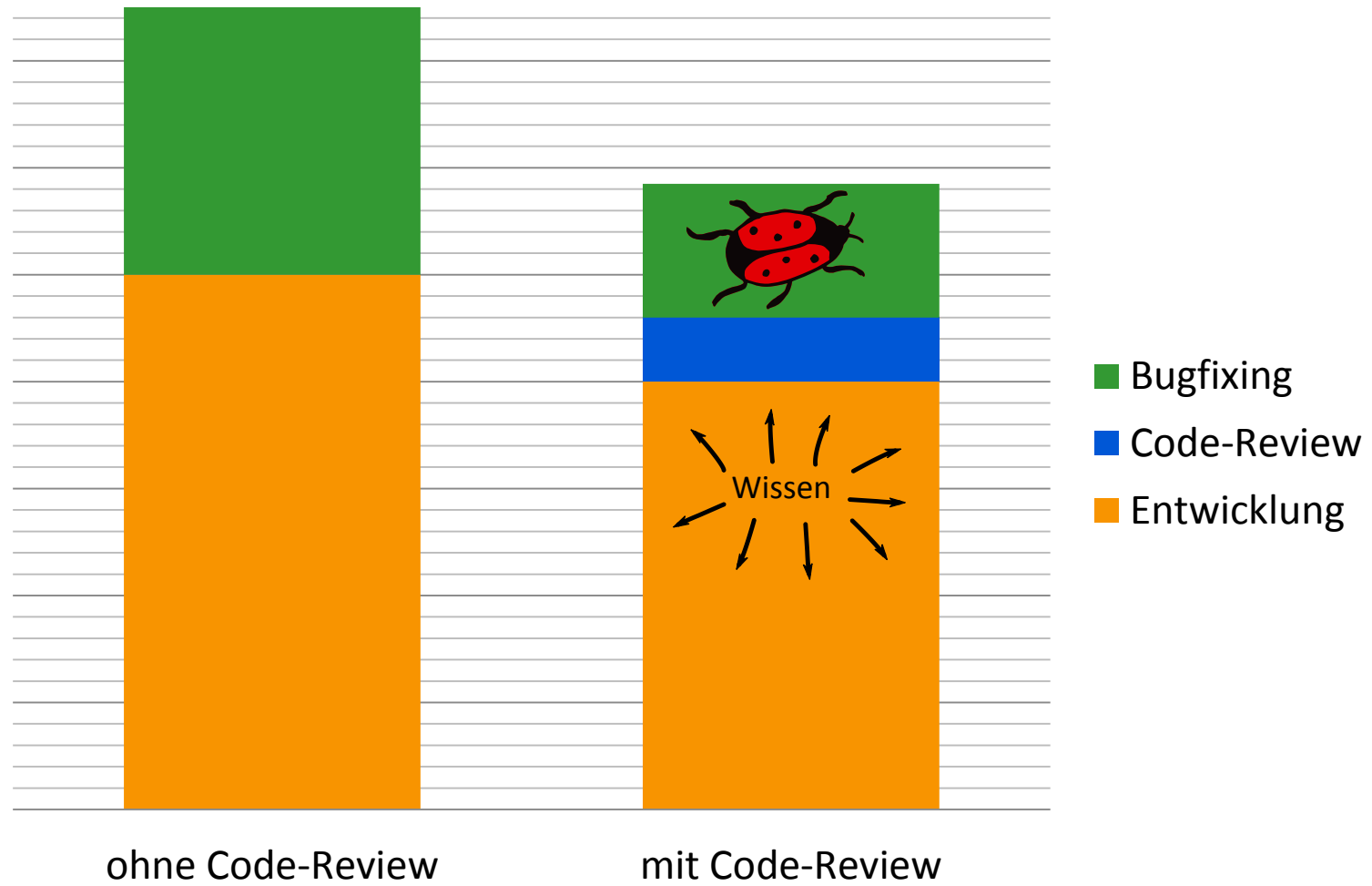
als kontinuierliche Lösung



Bugs finden







Jeder mit jedem!

200 – 400 Zeilen

1 – 2 Stunden



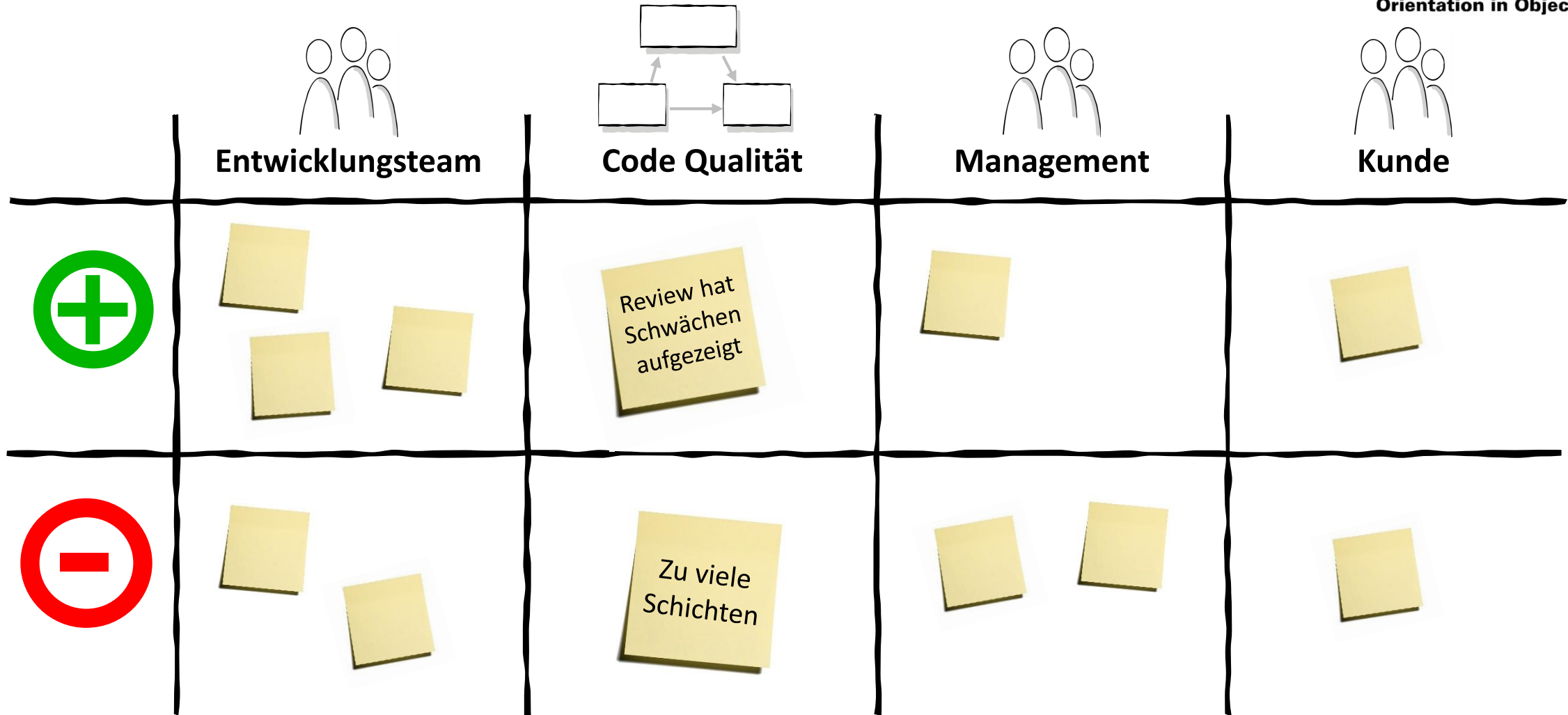
<http://support.smartbear.com/resources/cc/book/code-review-cisco-case-study.pdf>

Aufgaben aus Code-Review "Neue Benutzerverwaltung" #34

<input checked="" type="checkbox"/>	Potentielle NullPointerException im UserService abfangen
<input checked="" type="checkbox"/>	Typen in der Benutzer-Dokumentation entfernen
<input type="checkbox"/>	Unit-Test "UserServiceTest" um den neuen Anwendungsfall erweitern
<input type="checkbox"/>	Neues Widget für die Eingabe des Benutzernamens in Bibliothek auslagern
<input type="checkbox"/>	Type note...

     17:18

RETRO

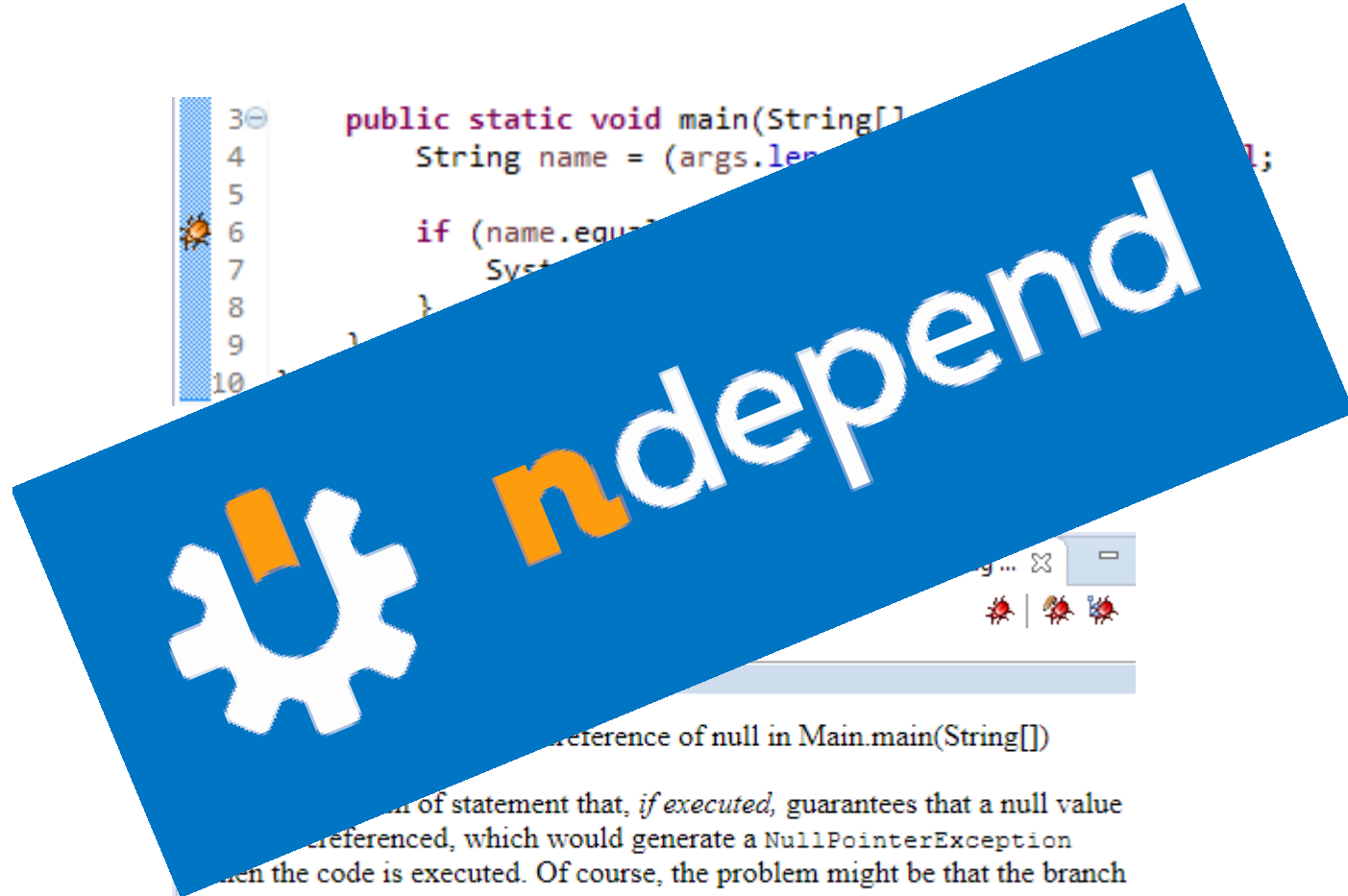


ANALYSE

```
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("null")) {  
7         System.out.println("Name is null");  
8     }  
9 }  
10
```

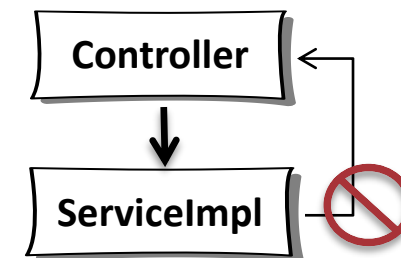


reference of null in Main.main(String[])

of statement that, *if executed*, guarantees that a null value is referenced, 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.



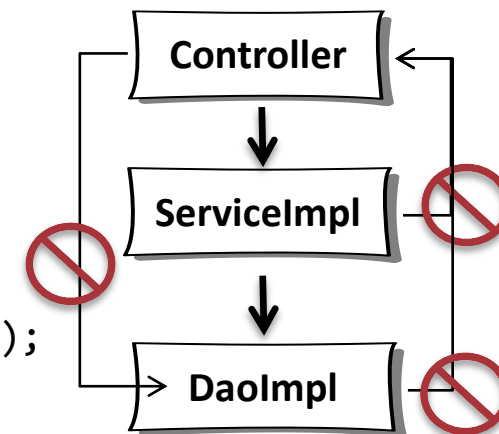
```
public class DemoServiceImpl implements DemoService {  
  
    @Autowired  
    private DemoController demoController;  
  
    public void superServiceMethod() {  
        demoController.doSomething();  
    }  
}
```



```
@RunWith(SpringRunner.class)
@SpringBootTest
public class DependencyTesterApplicationTests {
```

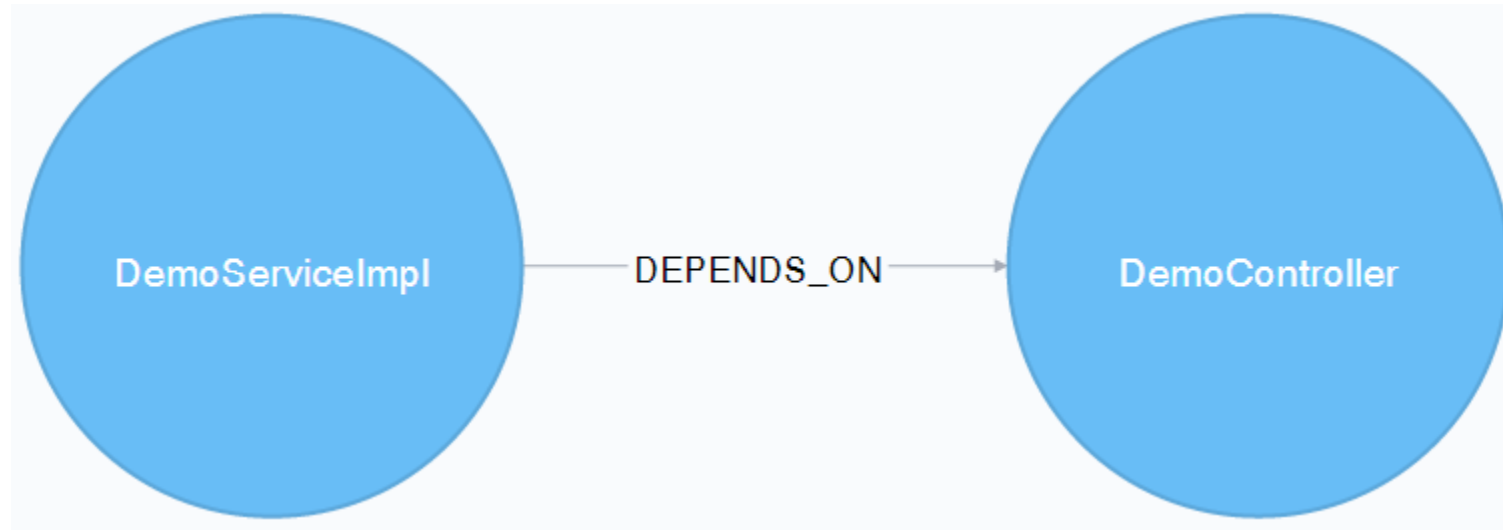
```
@Autowired
private DefaultListableBeanFactory beanFactory;
```

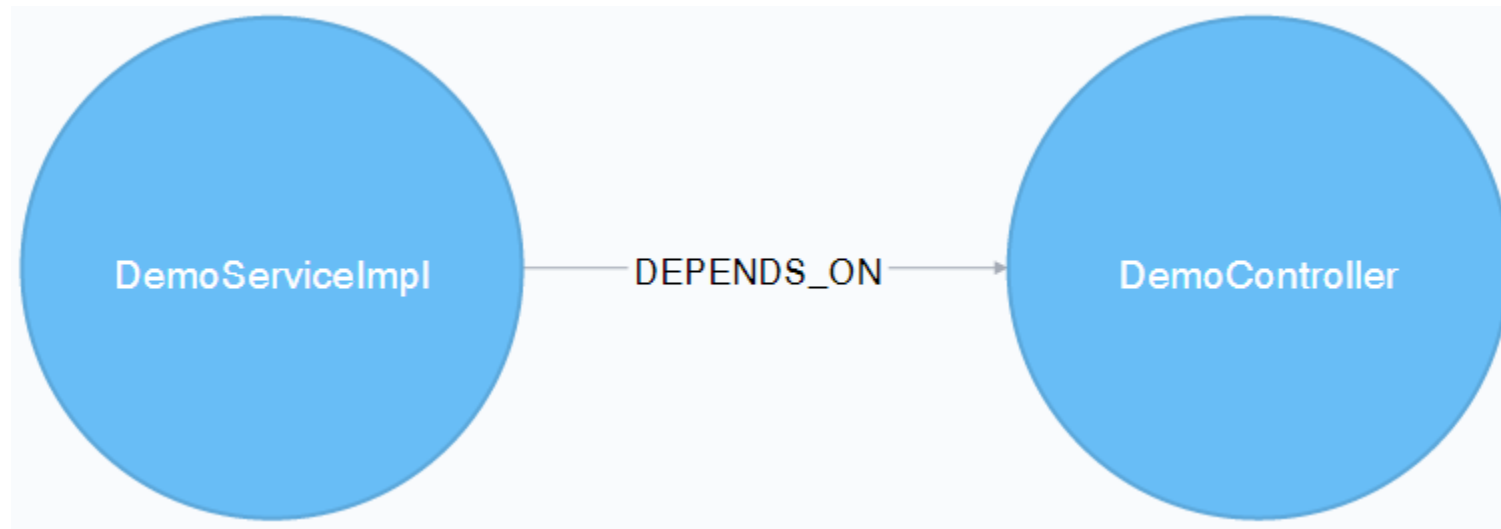
```
@Test
public void checkDependencies() {
    for (String beanName : beanFactory.getBeanDefinitionNames()) {
        for (String depName : beanFactory.getDependenciesForBean(beanName)) {
            checkNotAllowedConnection(beanName, depName, ".*ServiceImpl", ".*Controller");
            checkNotAllowedConnection(beanName,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



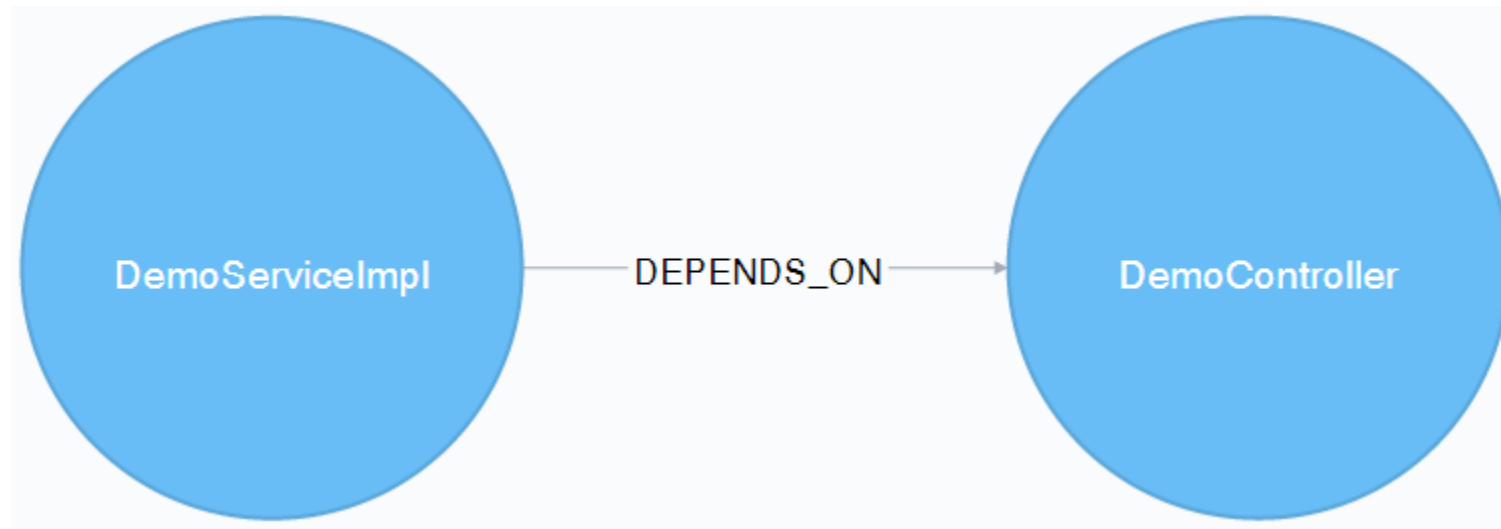
```
private void checkNotAllowedConnection(String beanName, String depName, String beanNamePattern,
    String dependencyNamePattern) {
    String msg = "Connection not allowed: " + beanName + " -> " + depName;
    assertFalse(msg, beanName.matches(beanNamePattern) && depName.matches(dependencyNamePattern));
}
}
```

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





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



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



```
<constraint id="my-rules:ServiceInServicePackages">  
  <description>Services must be in a service package</description>  
  <cypher><![CDATA[  
    MATCH (c:Class)-[:ANNOTATED_BY]->(a:Annotation) MATCH (a)-[:OF_TYPE]->(at {name: 'Service'}) MATCH (  
      p:Package)-[:CONTAINS]->(c) WHERE p.name <> 'impl' RETURN c,p LIMIT 10  
  ]]></cypher>  
</constraint>
```



```
<constraint id="my-rules:ServiceInServicePackages">  
  <description>Services must be in a service package</description>  
  <cypher><![CDATA[  
    MATCH (c:Class)-[:ANNOTATED_BY]->(a:Annotation) MATCH (a)-[:OF_TYPE]->(at {name: 'Service'}) MATCH (  
      p:Package)-[:CONTAINS]->(c) WHERE p.name <> 'impl' RETURN c,p LIMIT 10  
  ]]></cypher>  
</constraint>
```

17 Findings 😞



`git ratchet check`

<https://github.com/iangrunert/git-ratchet>

tl;dr;

Die richtigen Leute

Klare Struktur(en)

4-Augen-Prinzip

Miteinander reden

Automatische Codetests

Automatische Strukturtests



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