

12. Workshop Software-Reengineering

Smell Detection in Context

Kontextualisierung von automatisierten Designmängelwarnungen

Daniel Speicher
dsp@iai.uni-bonn.de

Institut für Informatik III
Universität Bonn

Sebastian Jancke
sebastian.jancke@soptim.de

SOPTIM AG

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

Smell Detection in Context

MOTIVATION

Ziel: Langfristige Evolvierbarkeit

- Refaktorisieren
- Langfristig: Systeme effizient weiterentwickeln
- Kurzfristig: höherer Aufwand

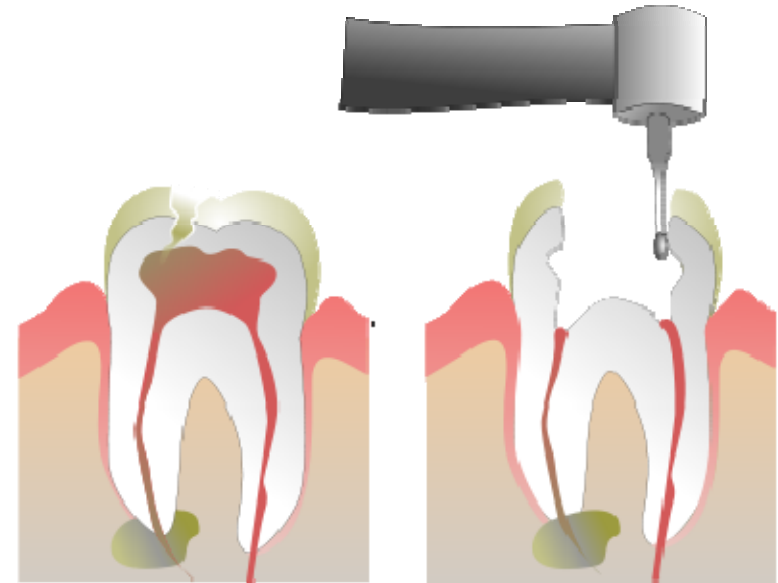
 Werkzeuge zur Refaktorisierung und **Analyse**

Refaktorisieren



Creative Commons Attributon:
www.flickr.com/photos/chrisschuette

**Zahnseide
(Floss)**



Jeremy Kemp, 2005, in die Gemeinfreiheit übergeben

**Wurzelbehandlung
(Root Canal)**

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

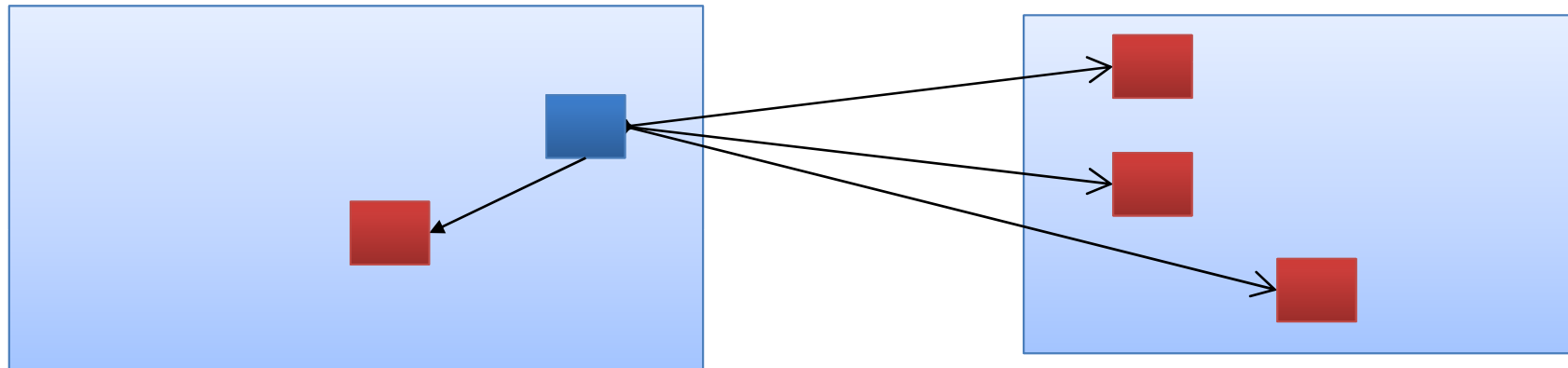
Smell Detection in Context

STATE OF THE ART

Design Probleme Erkennen

- Beck & Fowler: Smells
 - Ziel für Anwendung einer Refaktorisierung
 - Basiert auf „Intuition“
- Marinescu & Lanza: Design Flaw
 - Basiert auf Metriken
 - Abweichen von einem Qualitäts-Kriterium auf höherem Abstraktions-Niveau (als nur eine Metrik)

Beispiel: Neid (Feature Envy)



■ Methode ■ Daten

Einige (3) fremde Attribute

Verhältnis: $0.25 < \mathbf{Drittel}$

Daten gehören zu **Einigen** (1) Klassen

Beispiel: Neid (Feature Envy)

```
smell(feature_envy, MethodId) :-  
  
    internal_method(MethodId),  
  
    metric(method_access_to_foreign_data, MethodId, AFTD),  
    AFTD > 2, %2-5 = few  
  
    metric(method_locality_of_attribute_access, MethodId, LAA),  
    LAA < 0.3, % 0.3 = third  
  
    metric(method_foreign_data_providers, MethodId, FDP),  
    FDP =< 5 %2-5 = few  
  
    .
```


Überblick: Detection Strategies

Kohäsionsmangel

- God class
- Data class
- Feature Envy
- Brain class
- Brain method
- Iceberg class
- Middle man

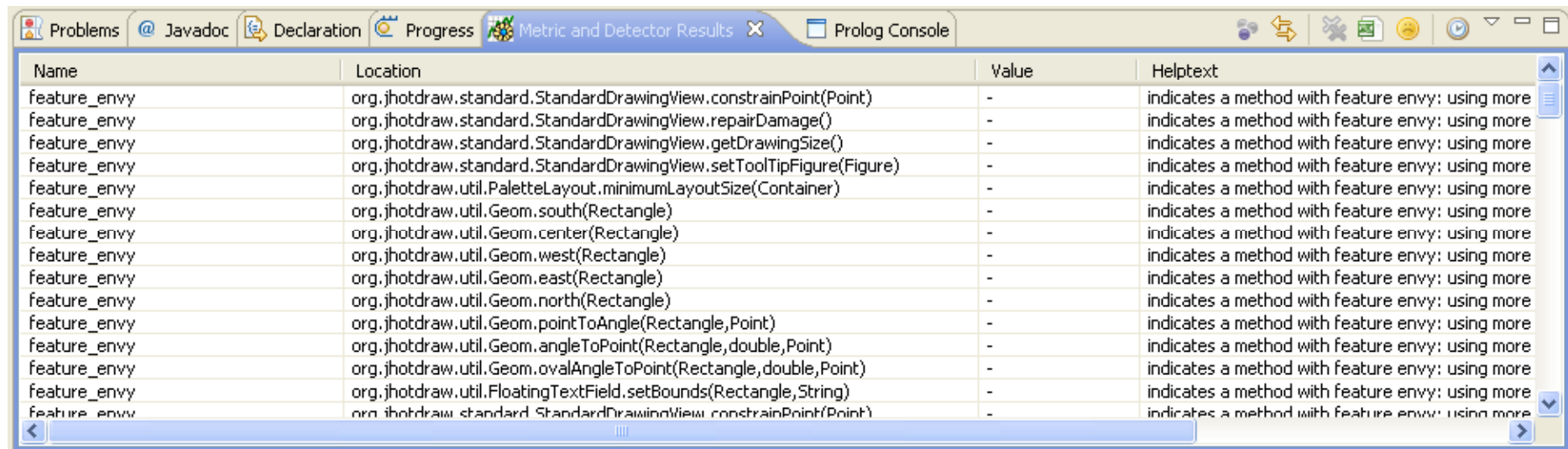
Zu hohe Kopplung

- Shotgun surgery
- Intensive Coupling
- Dispersed Coupling

Mängel in der Hierarchie

- Refused Parent
Bequest
- Tradition breaker

Aktuelle Werkzeugunterstützung



The screenshot shows an IDE window with several tabs: Problems, Javadoc, Declaration, Progress, Metric and Detector Results, and Prolog Console. The 'Metric and Detector Results' tab is active and displays a table with the following columns: Name, Location, Value, and Helptext. The table contains 15 rows of data, all with 'feature_envy' in the Name column and a hyphen in the Value column. The Helptext for all rows is 'indicates a method with feature envy: using more'. The Location column lists various methods from the 'org.jhotdraw' package.

Name	Location	Value	Helptext
feature_envy	org.jhotdraw.standard.StandardDrawingView.constrainPoint(Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.repairDamage()	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.getDrawingSize()	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.setToolTipFigure(Figure)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.PaletteLayout.minimumLayoutSize(Container)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.south(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.center(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.west(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.east(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.north(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.pointToAngle(Rectangle,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.angleToPoint(Rectangle,double,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.ovalAngleToPoint(Rectangle,double,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.FloatingTextField.setBounds(Rectangle,String)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.constrainPoint(Point)	-	indicates a method with feature envy: using more

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

Smell Detection in Context

PROBLEME

Offene Probleme

- Umgang mit Falsch-Positiven
 - ➔ *Adressiert durch Struktur-Kontext*
- Überflutung mit irrelevanten Informationen
 - Irrelevant im Kontext der aktuellen Aufgabe
 - Irrelevant durch Stabilität über die Zeit
 - ➔ *Adressiert durch Relevanz-Kontext*
- Design Probleme werden isoliert vorgestellt
 - ➔ *Adressiert durch Ko-Lokation*

Hypothesen

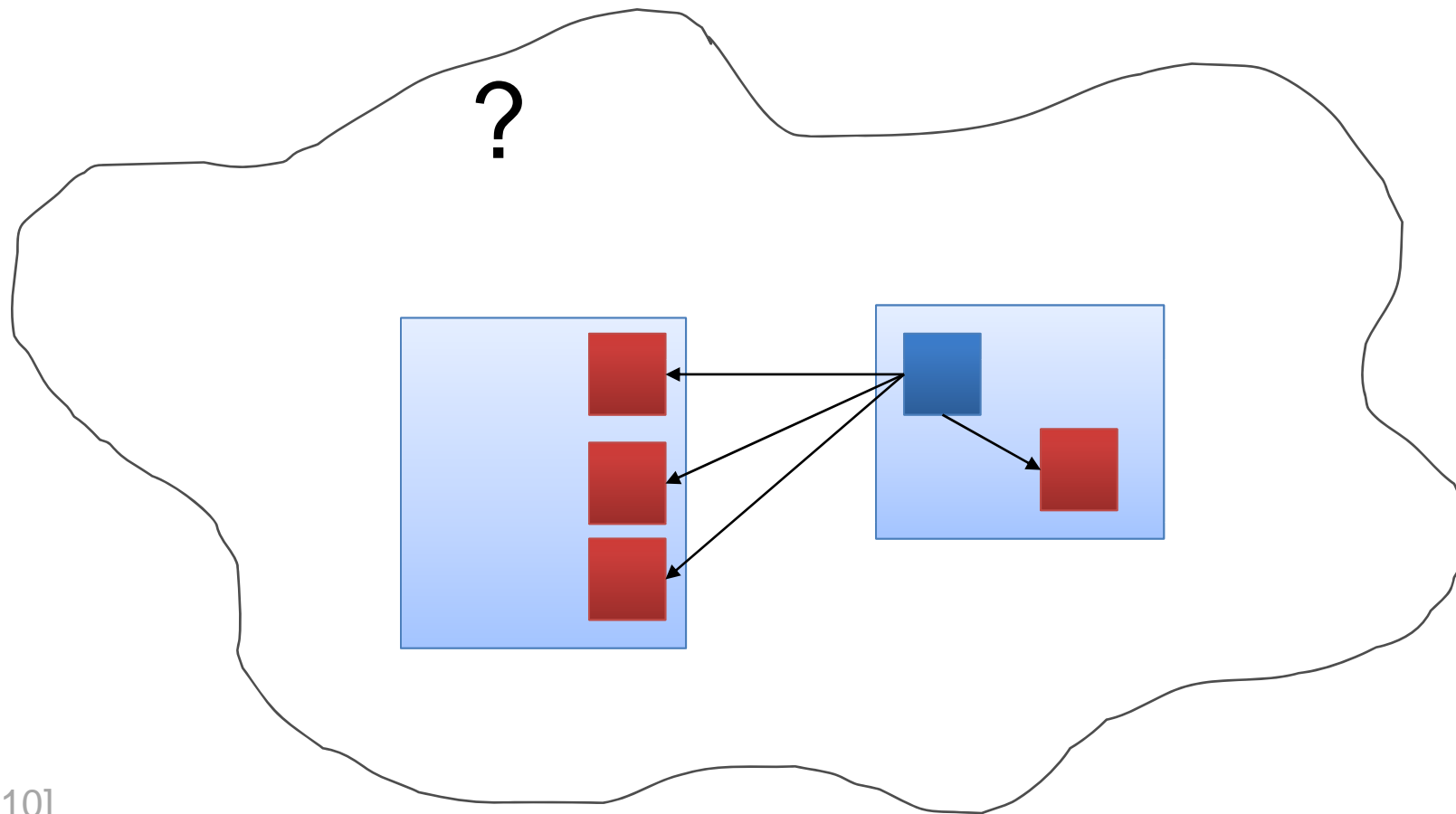
- Kontext-Bezug erleichtert Problem-Identifikation während des Floss Refactorings.
 - Strukturelle Hinweise reduzieren Falsch-Positive.
 - Gewichtung der Smells durch u.a. Historie und Task liefert nur relevante Smells im Projekt
 - Beziehungen zwischen Smells verbessern die Verständlichkeit eines Design Problems.

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

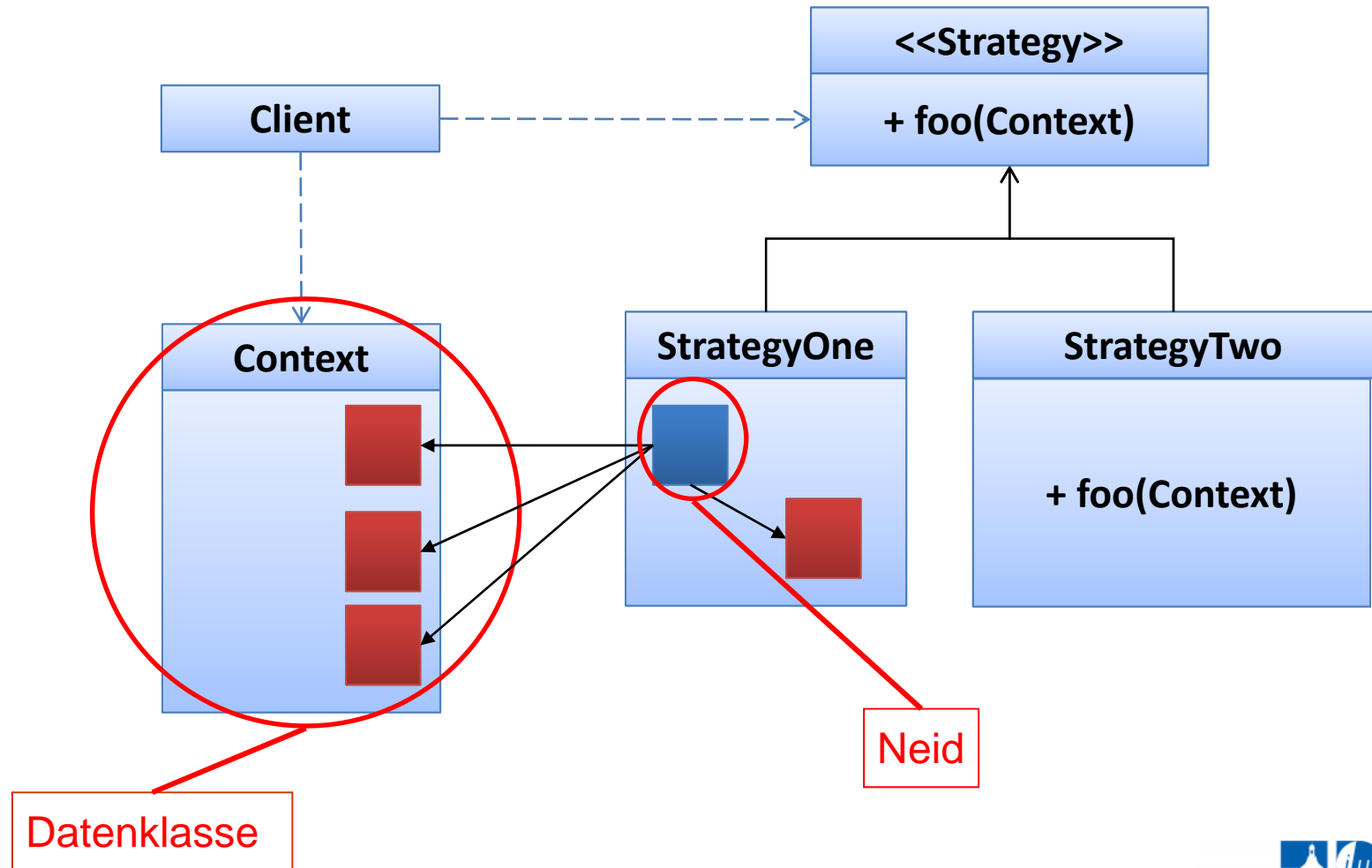
Smell Detection in Context

STRUKTUREN

Einflüsse auf Neid



Strategy Pattern impliziert Neid



Embedded DSLs

- Auch: Internal DSL, Fluent API (Fowler, Evans)
- Stile:
 - Geschachtelte Funktionen
 - Methodenkettens
- Ziel: Lesbarkeit

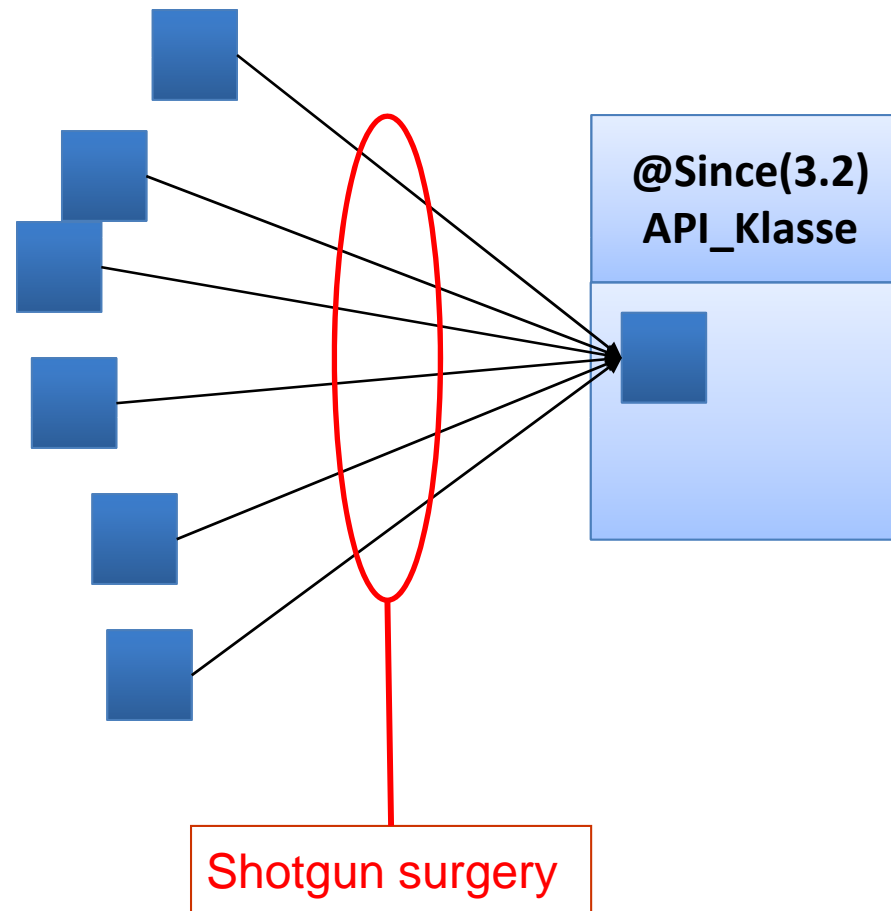
```
bind(CreditCardProcessor.class)  
    .annotatedWith(Names.named("Checkout"))  
    .to(PayPalCreditCardProcessor.class);
```

Embedded DSLs haben Smells

```
bind(CreditCardProcessor.class)
    .annotatedWith(Names.named("Checkout"))
    .to(PayPalCreditCardProcessor.class);
```

- Methoden Ketten
- „Law of Demeter“ Verletzung
- Möglicherweise auch
 - Gestreute Kopplung
 - Intensive Kopplung
- Abwägung: Smell - Lesbarkeit

APIs haben Smells



Unittests haben Smells

- Durch Erzeugung des Szenarios für den Test
 - Gestreute Kopplung
 - Neid
- Methode mit vielen Tests -> Viele Aufrufer
 - Shotgun Surgery
- Falsch-Negative: Komplexität eines Tests ?!

Umgang mit Falsch-Positiven, 1-3/6

1. Warnung global abschalten

 Werkzeug erfüllt seinen Zweck nicht

 Reduzierter Standard wird einhaltbar

2. Warnung ignorieren

 Kognitiver Aufwand!

3. Warnungen nach Schwere priorisieren

 Fokussierung auf die nützlichsten Korrekturen

 Benötigt gute Schwellwerte, Nicht-lokal

Umgang mit Falsch-Positiven, 4-6/6

4. Warnung lokal abschalten: `@SuppressWarnings`

 Prüfung andernorts weiter aktiv

 Abwägungsprozess intersubjektiv intransparent

5. Automatische Heuristiken für Ausnahmen

 Falsch Negative, Reduzierte Verständlichkeit

6. Rolle in Strukturen im Code zuordnen

 Aufwand für Strukturdefinition und Zuordnung

 Zusatznutzen: Strukturspezifische Bewertungen

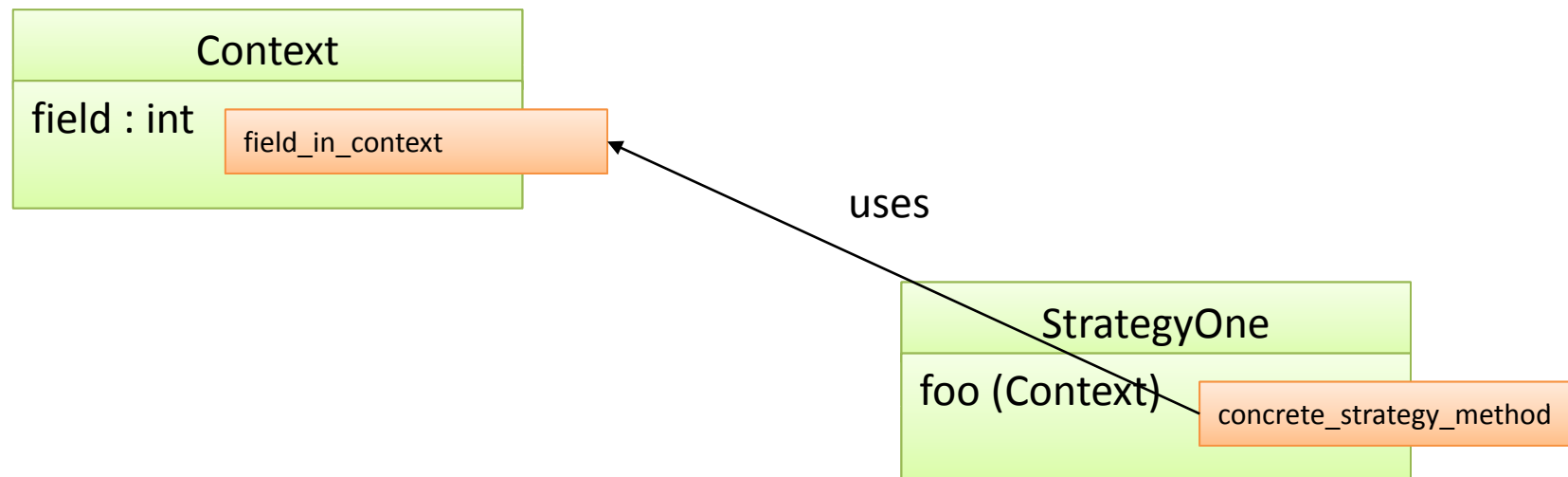
 Abwägungsprozess intersubjektiv nachvollziehbar

Falsch-Positive Reduzieren

- Strukturen können uns dabei helfen
- Strukturelle Hinweise:
 - Metadaten / Java-Annotationen („@Strategy“)
 - Namenskonventionen
 - Von Nutzer definierte Regeln
 - Pattern Detection

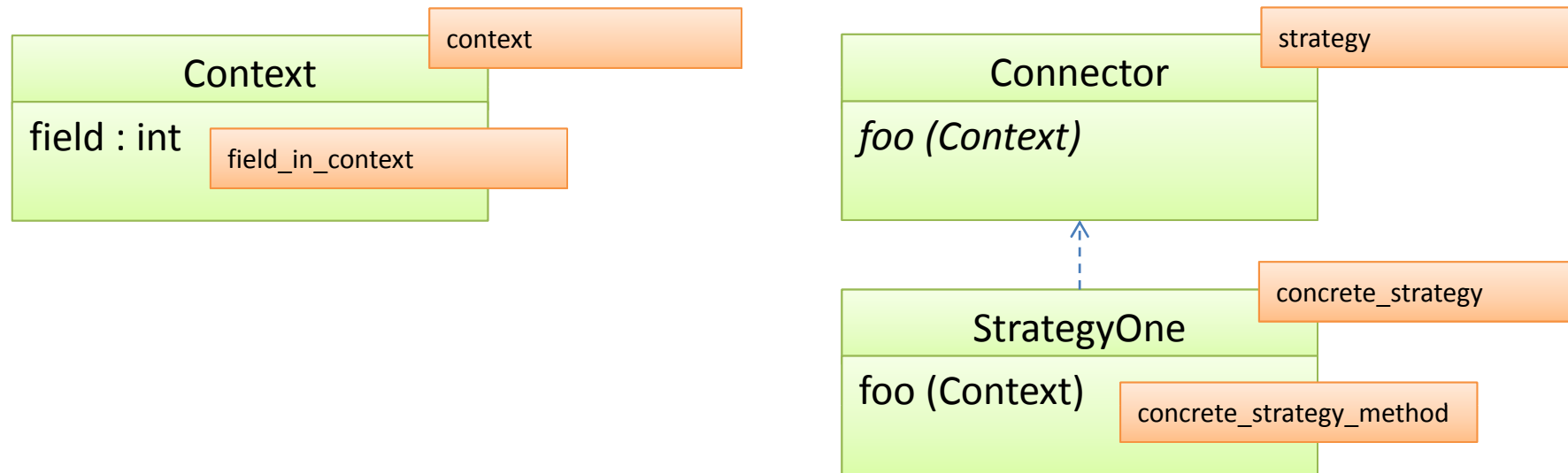
Strukturen definieren

```
structure (strategy_pattern) .  
role_uses (strategy_pattern,  
    concrete_strategy_method,  
    field_in_context) .
```

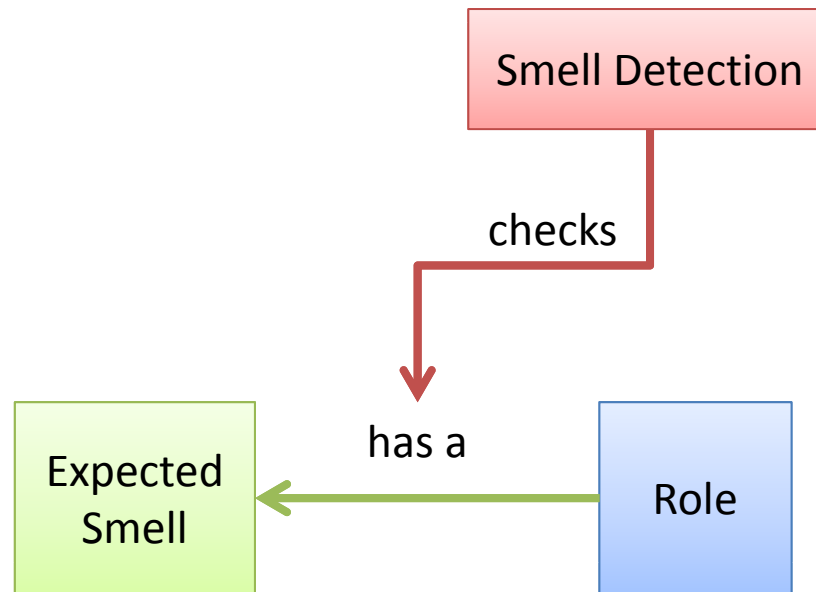


Annotierung mit Hinweisen

- `fqn_role(strategy_pattern, strategy, 'org.jhotdraw.framework.Connector')`



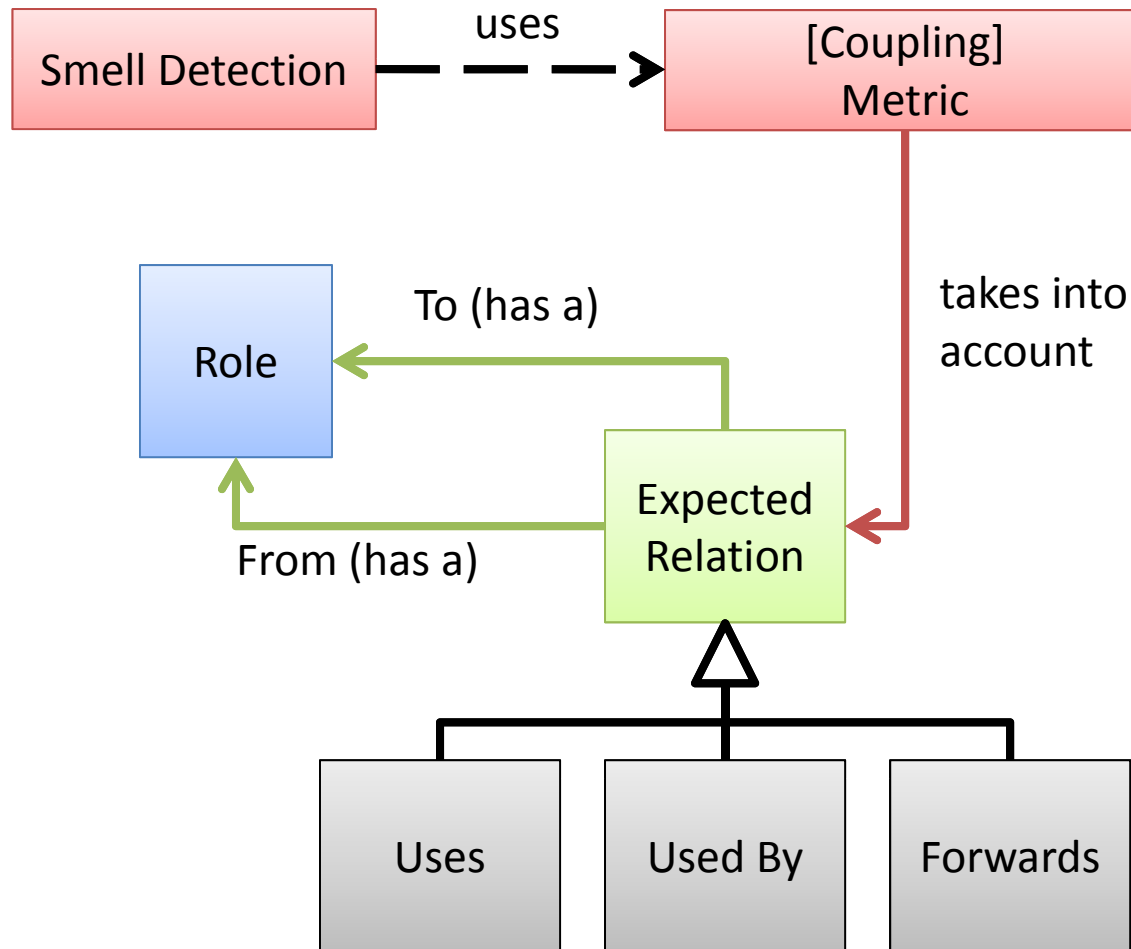
Natural Odor



Natural Odor

- Data class:
 - Element in Visitor Pattern
 - Data Structure in Procedural Components
- Refused Parent Bequest:
 - Concrete Decorator in Decorator Pattern
 - Component in Composite Pattern

Intended Collaboration



Intended Collaboration

- Feature Envy:
 - Concrete Visitor in Visitor Pattern
 - Concrete Strategy in Strategy Pattern
 - Creation Methods
 - Service in Procedural Components
 - Presenter in Model View Presenter
 - Unit Tests
- Method Chain and "Law of Demeter" Violation:
 - Client of Embedded DSL
- Shotgun Surgery:
 - Application Programming Interface
- Middle Man:
 - Abstract Decorator in Decorator Pattern
 - Facade Pattern
 - Adapter Pattern
 - Mediator Pattern
- Intensive Coupling and Dispersed Coupling:
 - Façade Pattern
 - Mediator Pattern
 - Client of Embedded DSL
 - View in Model View Presenter
 - Unit Test
 - Creation Methods

Ergebnis des Struktur-Kontext

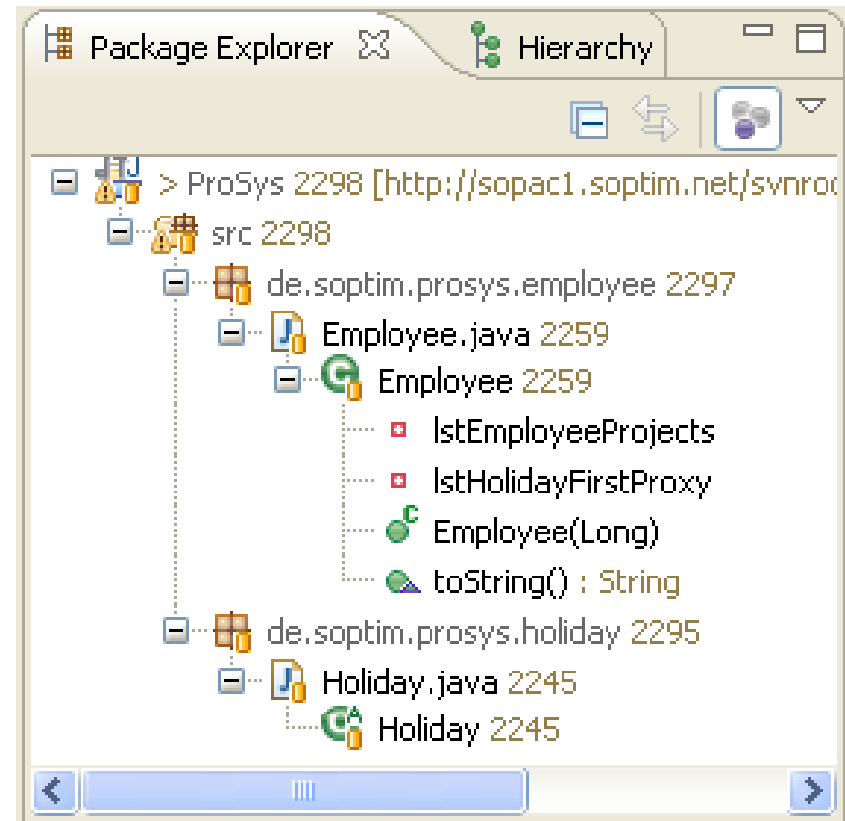
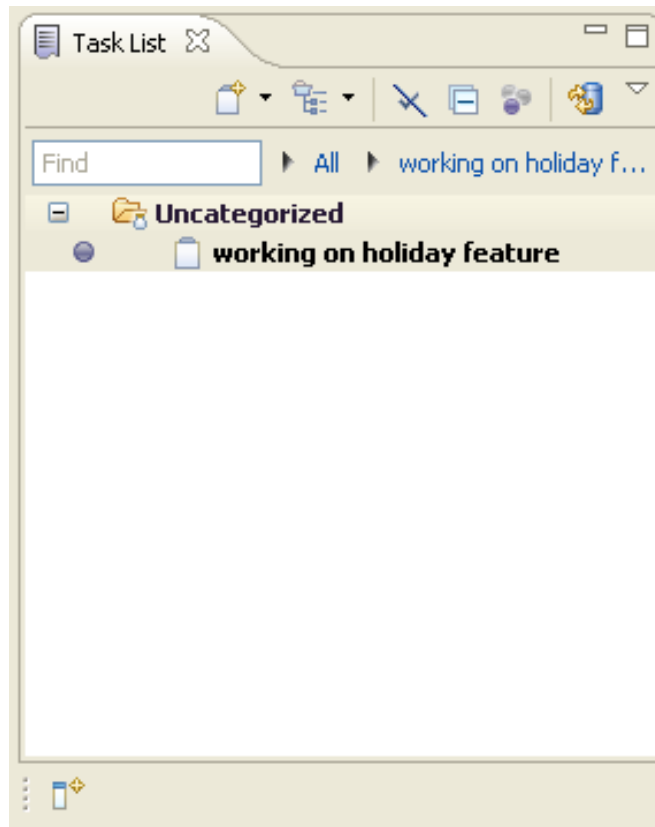
- Zusammenhang von Strukturen und Smells wird sichtbar
- Aussagekräftige Annotation
- Strukturen verursachen nun keine Falsch-Positiven

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

Smell Detection in Context

RELEVANZ

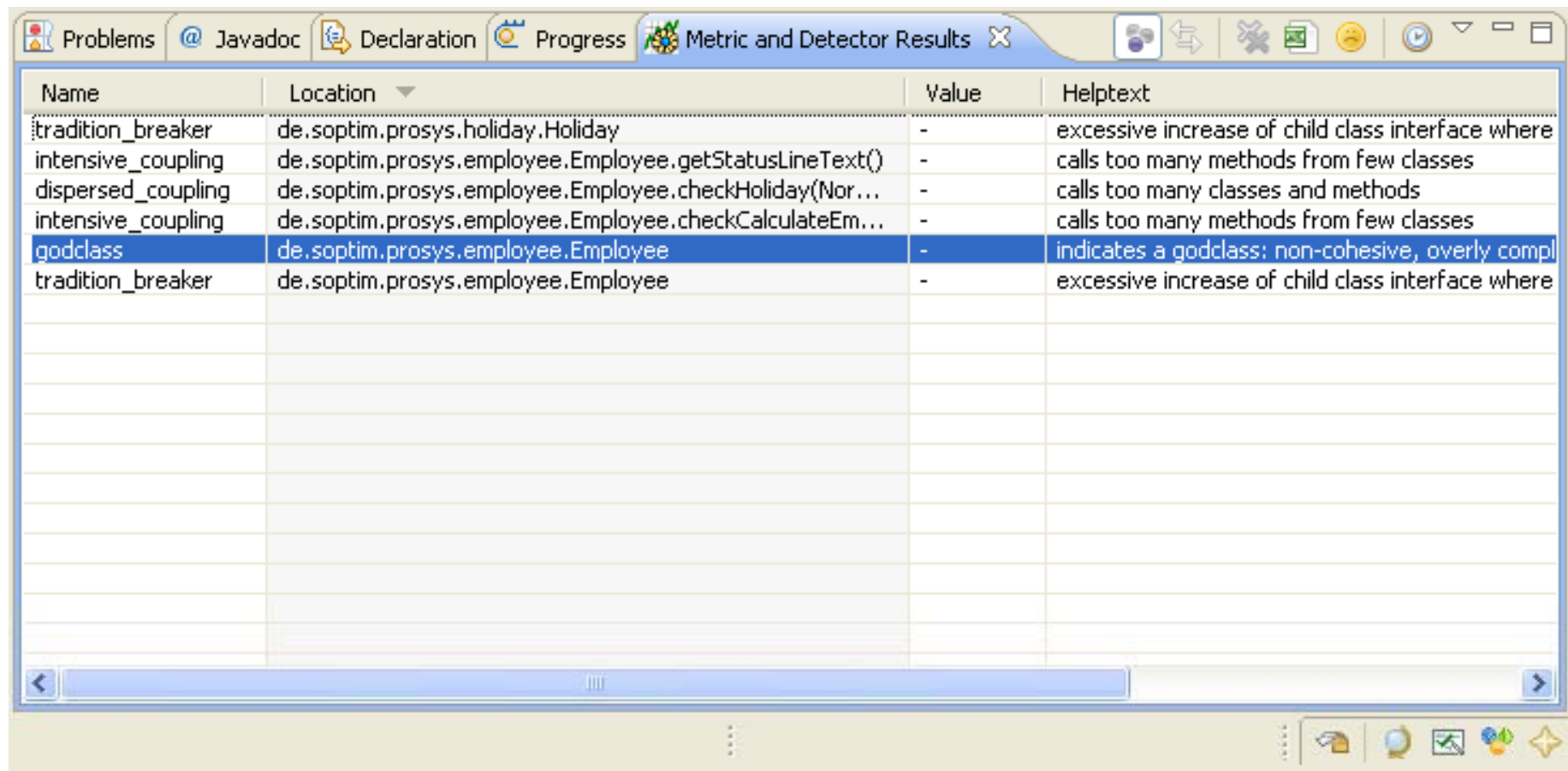
Fokussierung auf aktuelle Aufgabe



„Degree of Interest“ als Relevanz

- Interaktion mit Element A
 - erhöht $DOI(A)$
- Interaktion mit anderen Elementen
 - verringert $DOI(A)$
 - „Decay“
- Wenn $DOI(A) < 0$
 - Bei nächster Interaktion: $DOI(A)$ startet mit 0

Filtern nach Relevanz



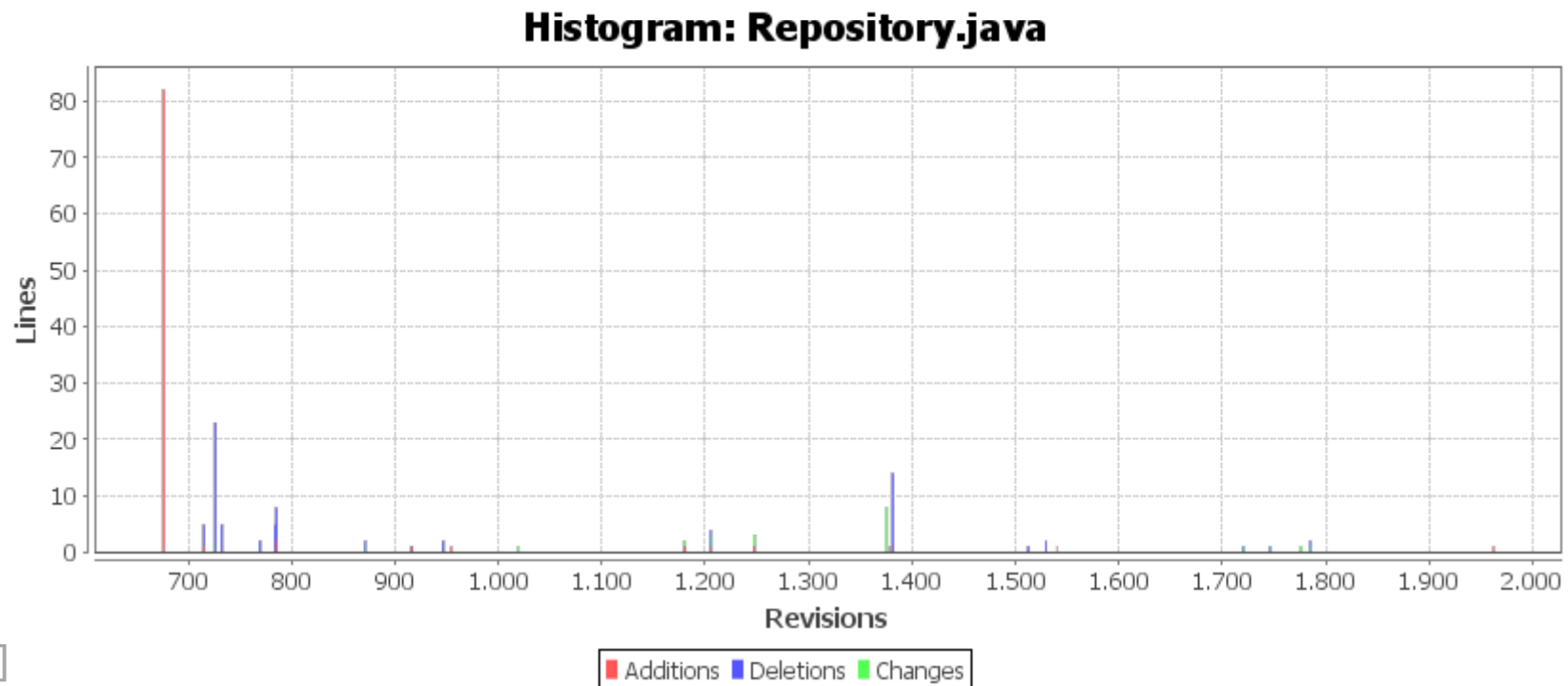
Name	Location	Value	Helptext
tradition_breaker	de.soptim.prosys.holiday.Holiday	-	excessive increase of child class interface where
intensive_coupling	de.soptim.prosys.employee.Employee.getStatusLineText()	-	calls too many methods from few classes
dispersed_coupling	de.soptim.prosys.employee.Employee.checkHoliday(Nor...	-	calls too many classes and methods
intensive_coupling	de.soptim.prosys.employee.Employee.checkCalculateEm...	-	calls too many methods from few classes
godclass	de.soptim.prosys.employee.Employee	-	indicates a godclass: non-cohesive, overly compl
tradition_breaker	de.soptim.prosys.employee.Employee	-	excessive increase of child class interface where

Temporal Degree of Interest

- Adaptiert Kersten's DOI auf Projekt-Historie
- Projekt-Historie: Alle Revisionen im SCM
- Interaktion = Änderung eines Elements
 - Zeilenweise Differenz zwischen Revisionen
 - Hinzufügen, Ändern, Löschen

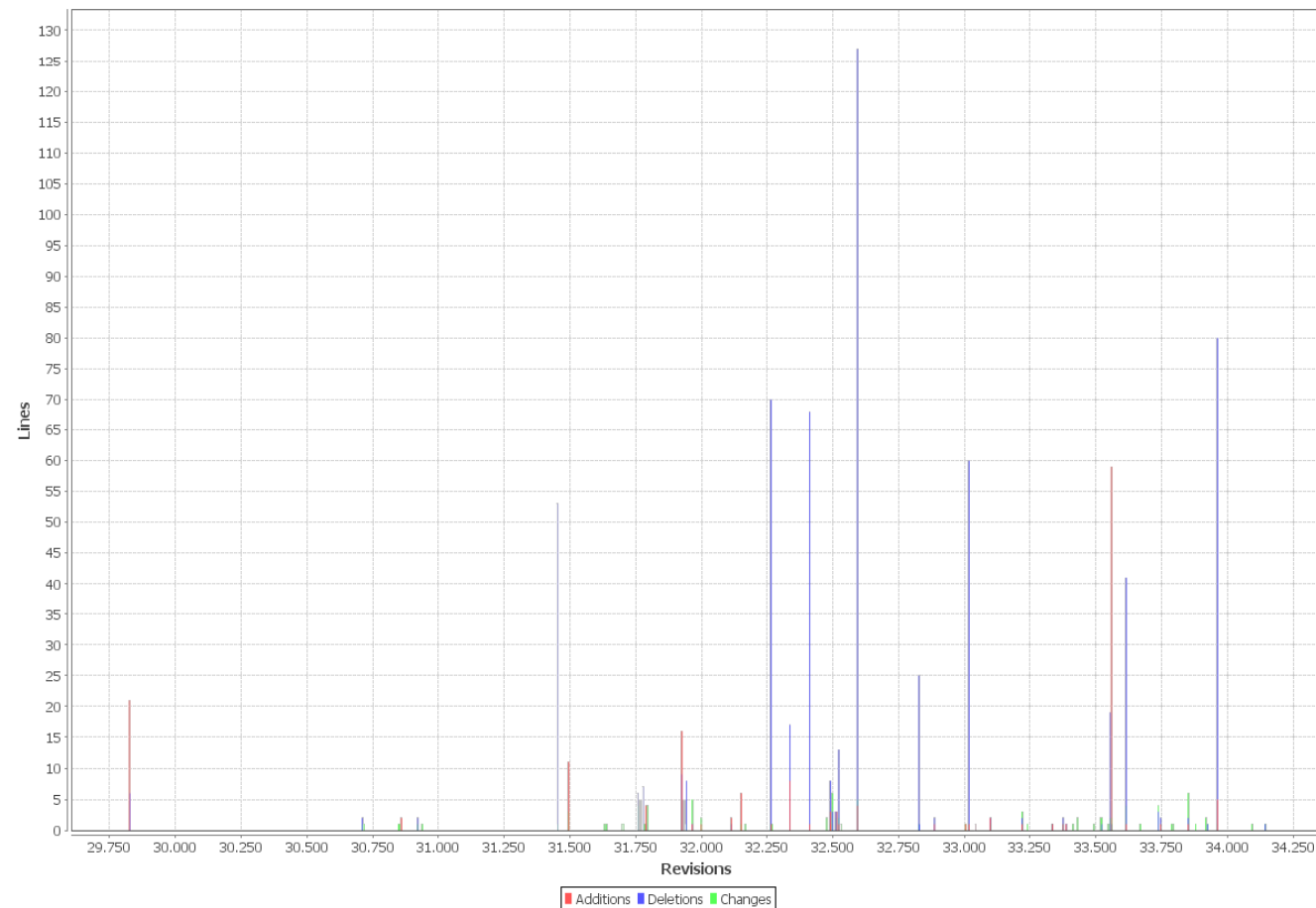
Relevanz gemessen an der Historie

- Temporal Stabiles Element
- DOI $\ll 0$

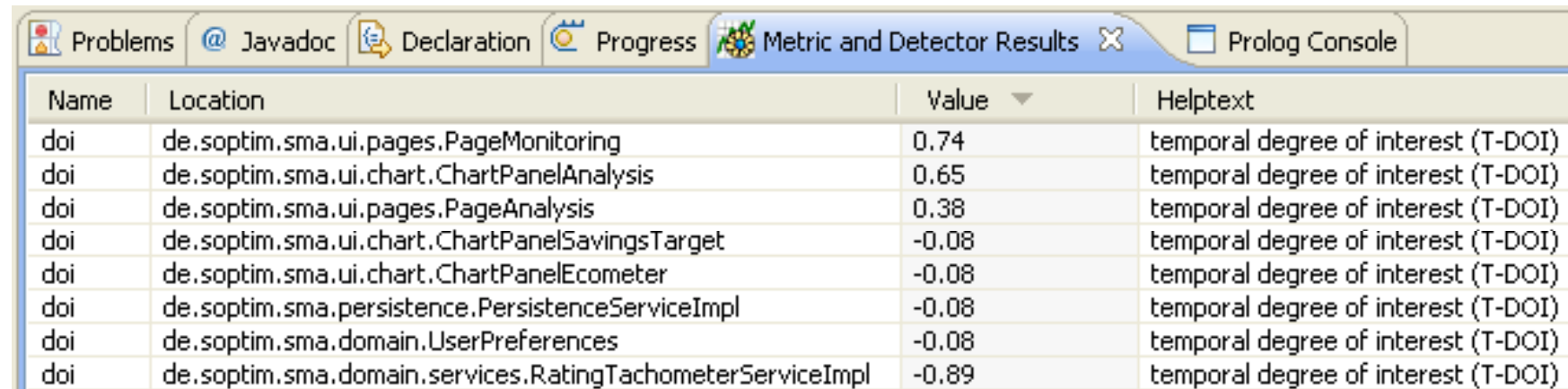


Relevanz gemessen an der Historie

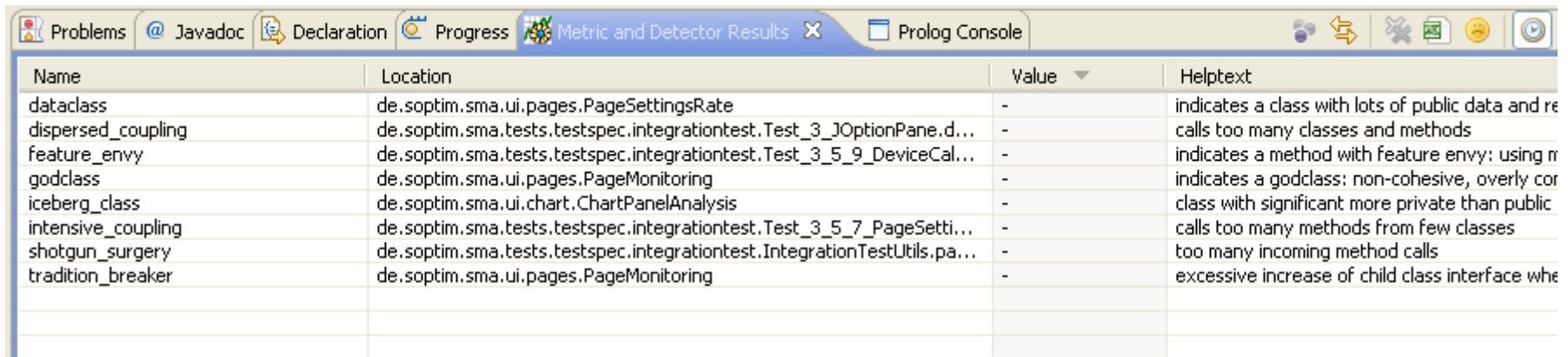
- Temporal Instabiles Element
- $DOI > 0$



Filtern nach Relevanz



Name	Location	Value	Help text
doi	de.soptim.sma.ui.pages.PageMonitoring	0.74	temporal degree of interest (T-DOI)
doi	de.soptim.sma.ui.chart.ChartPanelAnalysis	0.65	temporal degree of interest (T-DOI)
doi	de.soptim.sma.ui.pages.PageAnalysis	0.38	temporal degree of interest (T-DOI)
doi	de.soptim.sma.ui.chart.ChartPanelSavingsTarget	-0.08	temporal degree of interest (T-DOI)
doi	de.soptim.sma.ui.chart.ChartPanelEcometer	-0.08	temporal degree of interest (T-DOI)
doi	de.soptim.sma.persistence.PersistenceServiceImpl	-0.08	temporal degree of interest (T-DOI)
doi	de.soptim.sma.domain.UserPreferences	-0.08	temporal degree of interest (T-DOI)
doi	de.soptim.sma.domain.services.RatingTachometerServiceImpl	-0.89	temporal degree of interest (T-DOI)



Name	Location	Value	Help text
dataclass	de.soptim.sma.ui.pages.PageSettingsRate	-	indicates a class with lots of public data and re
dispersed_coupling	de.soptim.sma.tests.testspec.integrationtest.Test_3_10OptionPane.d...	-	calls too many classes and methods
feature_envy	de.soptim.sma.tests.testspec.integrationtest.Test_3_5_9_DeviceCal...	-	indicates a method with feature envy: using m
godclass	de.soptim.sma.ui.pages.PageMonitoring	-	indicates a godclass: non-cohesive, overly cor
iceberg_class	de.soptim.sma.ui.chart.ChartPanelAnalysis	-	class with significant more private than public
intensive_coupling	de.soptim.sma.tests.testspec.integrationtest.Test_3_5_7_PageSetti...	-	calls too many methods from few classes
shotgun_surgery	de.soptim.sma.tests.testspec.integrationtest.IntegrationTestUtils.pa...	-	too many incoming method calls
tradition_breaker	de.soptim.sma.ui.pages.PageMonitoring	-	excessive increase of child class interface whe

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

Smell Detection in Context

KO-LOKATION

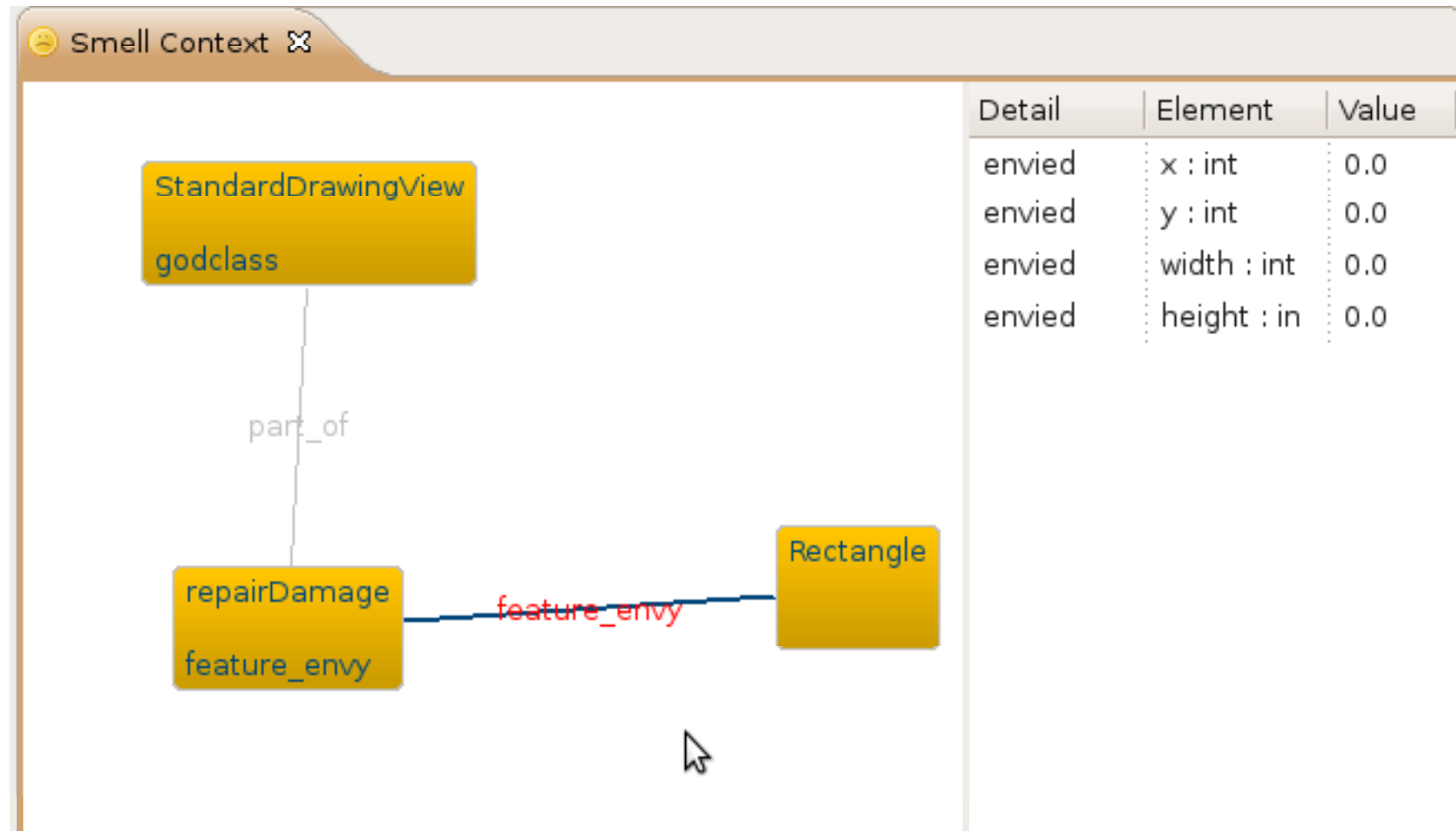
Ko-Lokation: Benachbarte Smells darstellen

- Fast kein Design Problem existiert in Isolation
- Neid
 - Beneide ich eine Datenklasse ?
- Shotgun Surgery
 - Welche Typen und Methoden sind Aufrufer ?
- „Gottklasse“
 - Was sind die komplexen Methoden ?

Kontext eines Smells

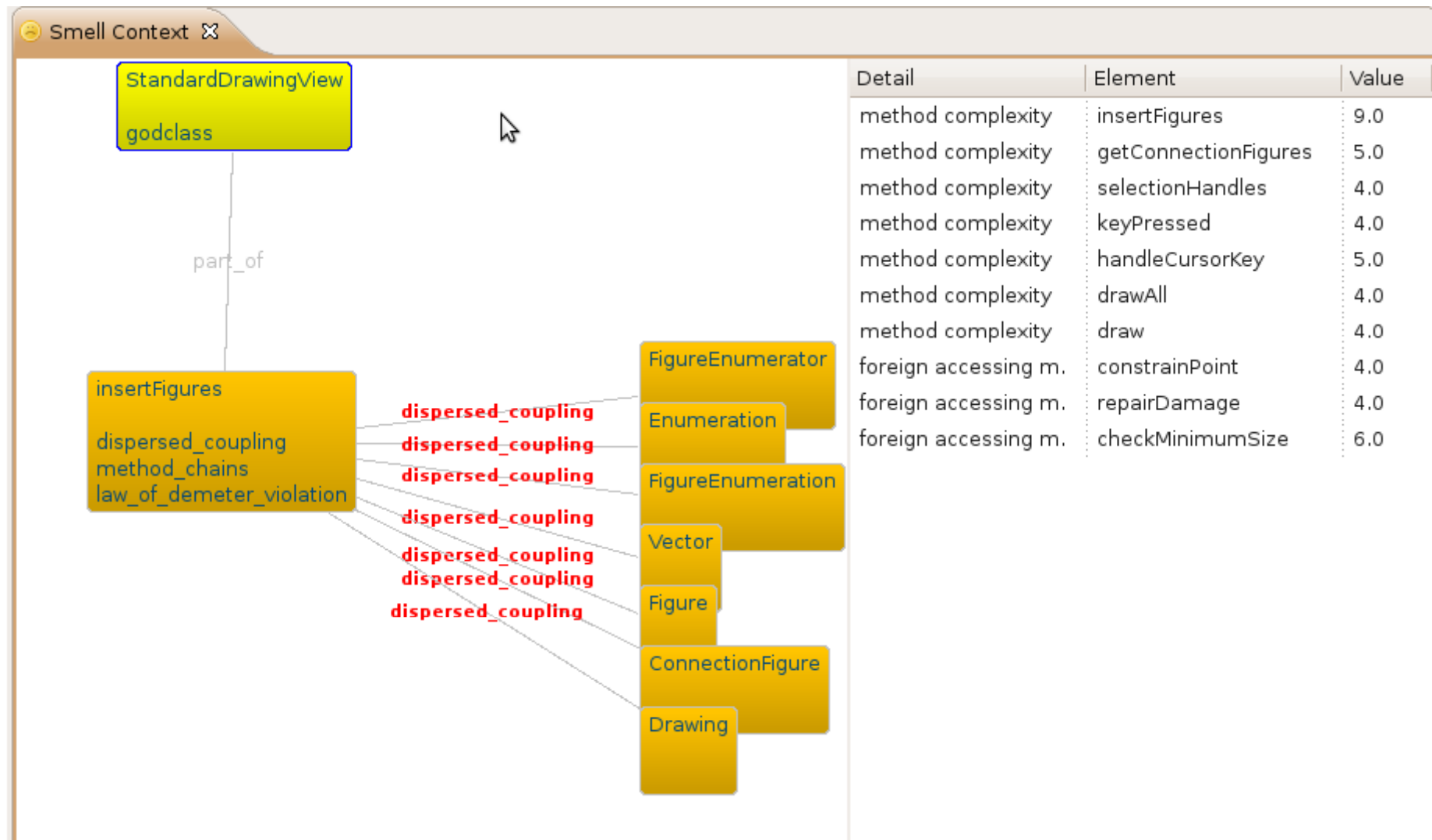
- Relationen und Details sind von Interesse
- Jeder Smell definiert Relationen
 - auf denen zB Metriken der Detection Strategies arbeiten
- Jeder Smell definiert Details
 - für ein Element (mit Smell)
 - für Relation

„Smell Context View“



Implementierung aus [J10] in [CV]

„Smell Context View“



MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
KO-LOKATION
CASE STUDY

Smell Detection in Context

CASE STUDY

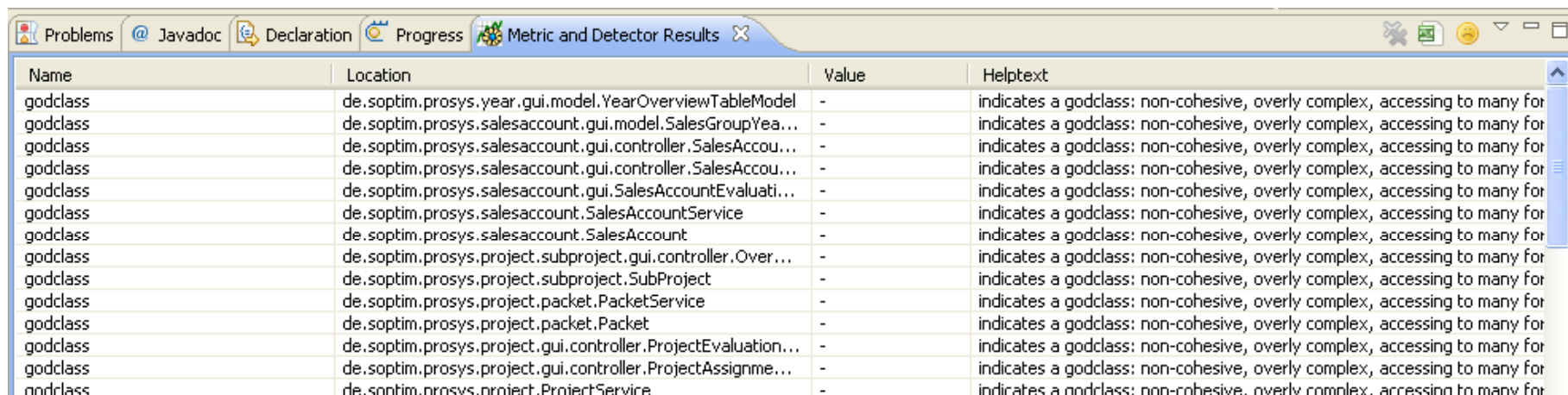
„Case Study“ ist eine empirische Methode

- Untersucht zeitnah Phänomene in ihrem Kontext im echten Leben
- Single Case vs. **Multiple Case Study**
- Design einer Studie kann enthalten:
 - Triangulation (Datenquellen, Arten, Beobachter)
 - Beweisketten
 - Muster, Rivalisierende Erklärungen
 - „Critical Cases“, Reproduktion der Cases
- Dokumentation im „Case Study Protocol“

1. Fall „Klassische“ Smell Detection

1. Interviews zum Hintergrund der Teilnehmer
2. Schulung „Design Smells“ + Tooling
3. Beobachtung der Entwickler
 - „Think-aloud“ Protokolle
4. Interviews am Ende

[J10]



The screenshot shows a table of detected 'godclass' smells in an IDE. The table has four columns: Name, Location, Value, and Helptext. The 'Value' column contains dashes, and the 'Helptext' column contains a consistent description for each entry: 'indicates a godclass: non-cohesive, overly complex, accessing to many for...'. The 'Name' column lists 'godclass' for 13 entries and 'nndclass' for the last entry. The 'Location' column shows various package and class paths.

Name	Location	Value	Helptext
godclass	de.soptim.prosys.year.gui.model.YearOverviewTableModel	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.gui.model.SalesGroupYea...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.gui.controller.SalesAccou...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.gui.controller.SalesAccou...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.gui.SalesAccountEvaluati...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.SalesAccountService	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.salesaccount.SalesAccount	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.subproject.gui.controller.Over...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.subproject.SubProject	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.packet.PacketService	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.packet.Packet	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.gui.controller.ProjectEvaluation...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	de.soptim.prosys.project.gui.controller.ProjectAssignme...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
nndclass	de.soptim.prosys.project.ProjectService	-	indicates a nndclass: non-cohesive, overly complex, accessing to many for

Ergebnisse „Klassisch“

- Identifikation der Probleme „manuell“
 - Suchen im ResultView
 - Korrespondierenden Quelltext lesen
 - Eclipse Call Graph und Reference Search
- Falsch-Positive fallen kaum auf oder werden einfach ignoriert

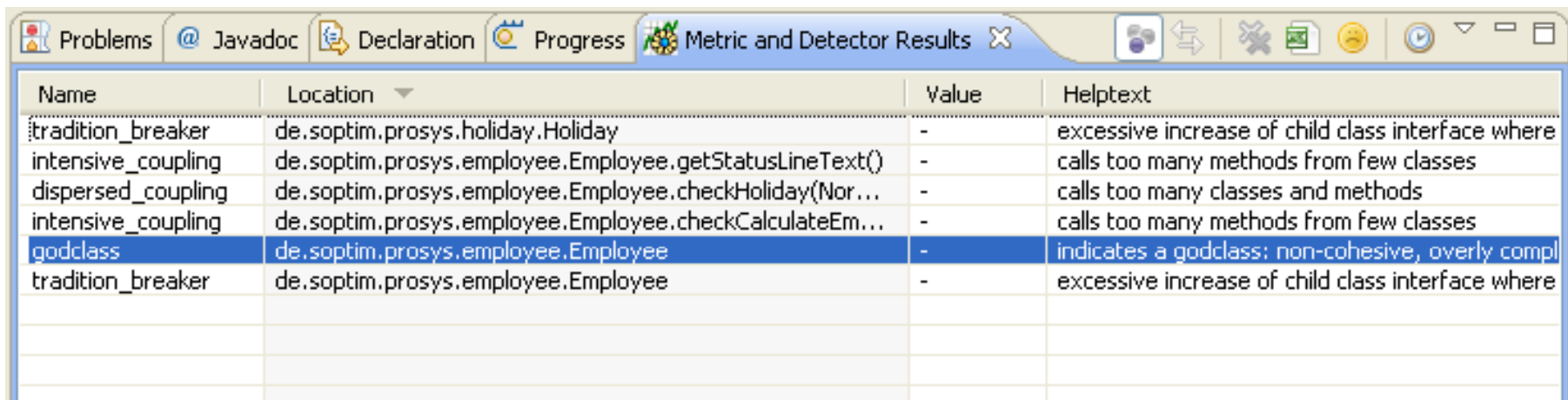
“Iteration über alle Ergebnisse, Caller, usw.“

“Lästige Handarbeit“

2. Fall „Smell Detection in Context“

1. Schulung des Toolings
2. Beobachtung der Entwickler
3. Interviews am Ende
4. Weiteres Interview 2 Wochen später

[J10]



Name	Location	Value	Help text
tradition_breaker	de.soptim.prosys.holiday.Holiday	-	excessive increase of child class interface where
intensive_coupling	de.soptim.prosys.employee.Employee.getStatusLineText()	-	calls too many methods from few classes
dispersed_coupling	de.soptim.prosys.employee.Employee.checkHoliday(Nor...	-	calls too many classes and methods
intensive_coupling	de.soptim.prosys.employee.Employee.checkCalculateEm...	-	calls too many methods from few classes
godclass	de.soptim.prosys.employee.Employee	-	indicates a godclass: non-cohesive, overly compl
tradition_breaker	de.soptim.prosys.employee.Employee	-	excessive increase of child class interface where

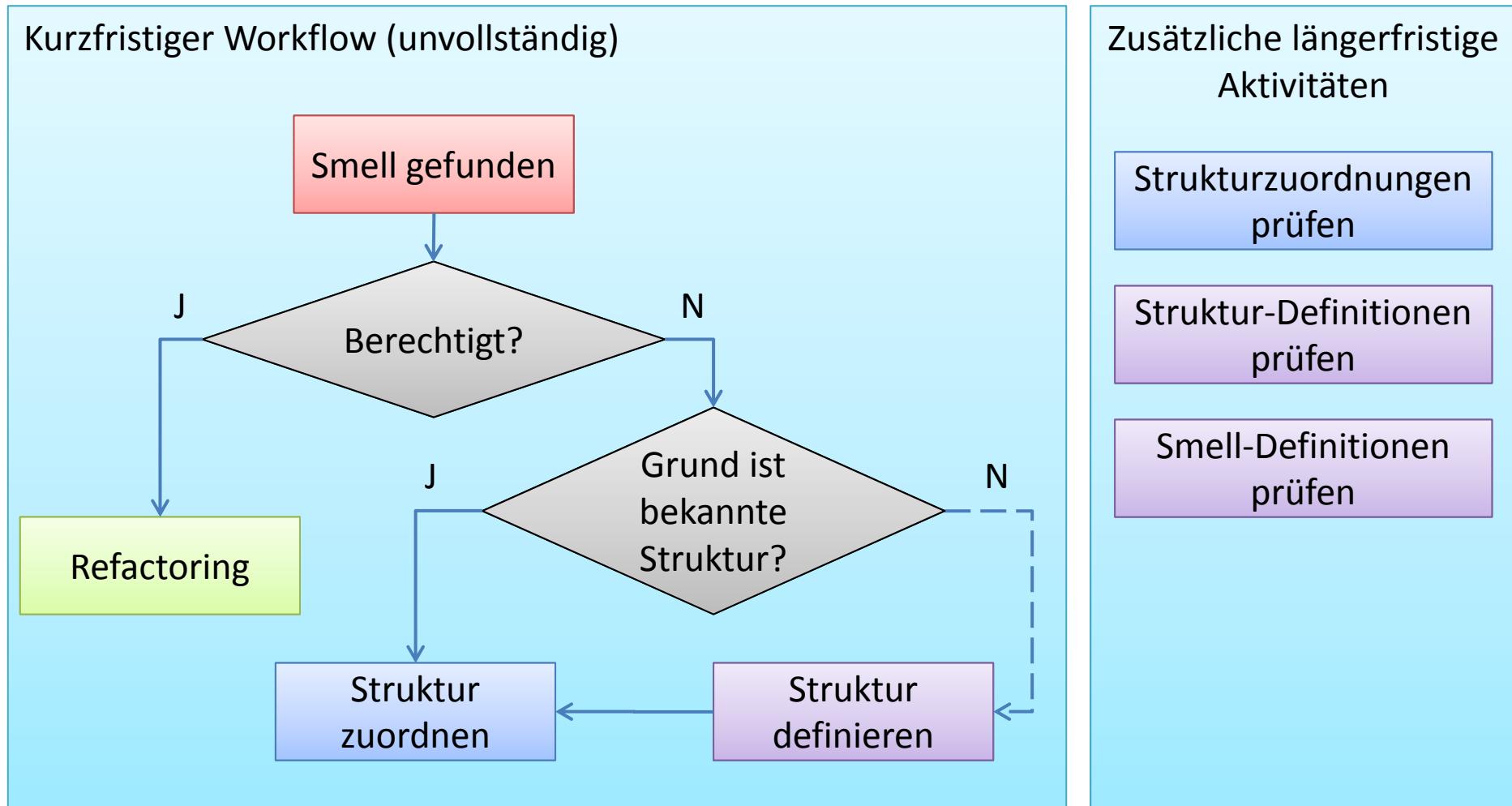
Ergebnisse „SDIC“

- Ausnutzung der Relevanz-Filterung (Aufgaben)
- Smell Context View für konkrete Probleme konsultiert
- Struktur-Definitionen werden ausgenutzt
- Temporales Filtern wird nicht genutzt
 - anscheinend ist die Ergebniss-Menge pro Aufgabe schon klein genug

Fazit

- Annahme „Kontext fehlt“ bestätigt sich in der Studie des klassischen Prototyps.
- Strukturen sind Abwägungen von Design-Prinzipien und haben Einfluss auf die Erkennung von Smells
- Relevanz-Filterung pro Aufgabe macht Smell Detection effizient genug für Floss Refactoring
- Kontext eines Smells unterstützt das subjektive Verständnis

(Coding-)Kultur-Evolution



Q & A

Literatur

- [J10] S. Jancke. **Smell detection in context, Diploma thesis. University of Bonn, 2010.**
- [Ker07] M. Kersten. Focusing knowledge work with task context. PhD thesis, University of British Columbia, 2007.
- [LM06] M. Lanza and R. Marinescu. Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems. Springer, 1 edition, 9 2006.
- [LHR88] K. Lieberherr, I. Holland, and A. Riel. Object-oriented programming: An objective sense of style. ACM SIGPLAN Notices, 23(11):323-334, 1988.
- [MH09] E. Murphy-Hill. Programmer Friendly Refactoring Tools. PhD thesis, Portland State University, 2009.
- [RDGM04] Daniel Ratiu, Stéphane Ducasse, Tudor Gîrba, and Radu Marinescu. Using history information to improve design flaws detection. In CSMR '04: Proceedings of the Eighth Euromicro Working Conference on Software Maintenance and Reengineering (CSMR'04), page 223, Washington, DC, USA, 2004. IEEE Computer Society.
- [SRK07] D. Speicher, T. Rho, and G. Kniesel. JTransformer - eine logikbasierte Infrastruktur zur Codeanalyse. In Proc. 9. Ws. Software-Reengineering (WSR 2007), pages 21-22, 2007.
- [Ecl] Eclipse Project, <http://eclipse.org/>
- [CV] Cultivate Project, <http://sewiki.iai.uni-bonn.de/projects/cultivate/start>
- [dsp] Eigene Hintergrundüberlegungen