

Erkennung von semantisch zusammenhängenden Quelltextabschnitten anhand von Komponententests

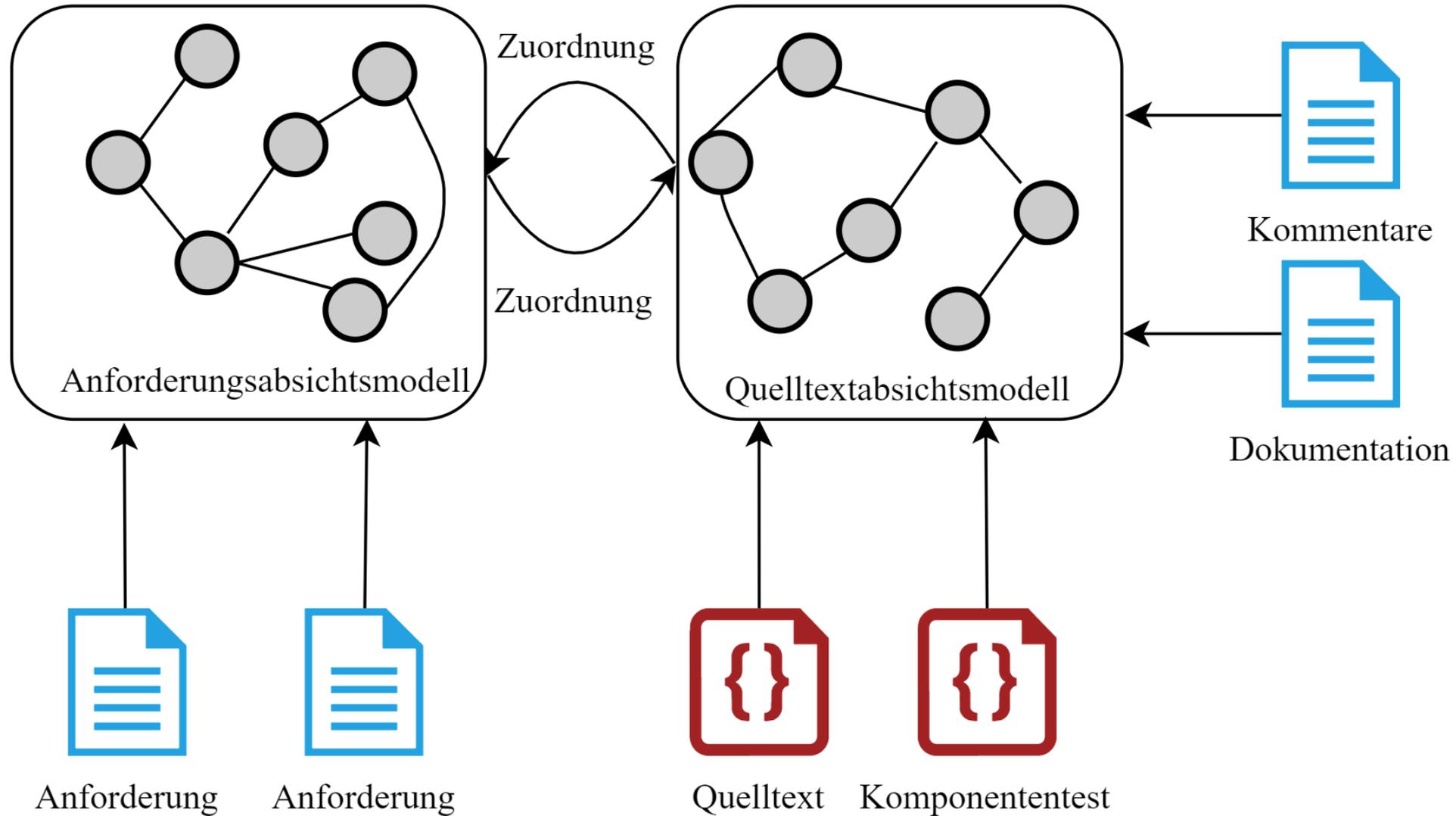
Martin Wittlinger

Betreut von Tobias Hey

IPD Tichy, KIT-Fakultät für Informatik



INDIRECT

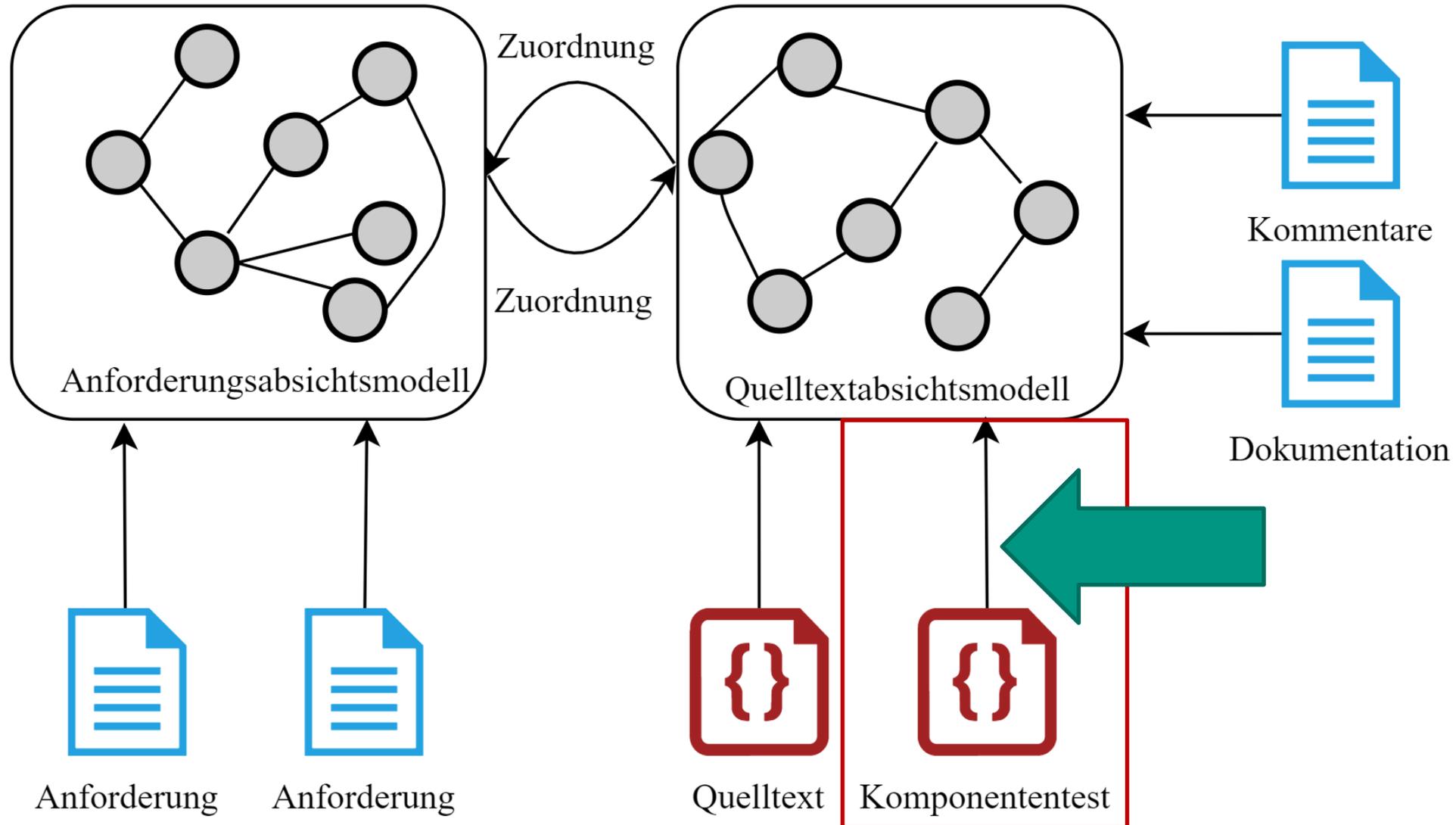


Warum Komponententests?

- Komponententests sind Dokumentation
- Komponententests besitzen eine Intention des Programmierers
- Komponententests sind eine Sicht auf den Code

```
1  @Test
2  public void evaluatesExpression() {
3      Calculator calculator = new Calculator();
4      int sum = calculator.evaluate("1+2+3");
5      assertEquals(6, sum);
6  }
7 }
```

INDIRECT

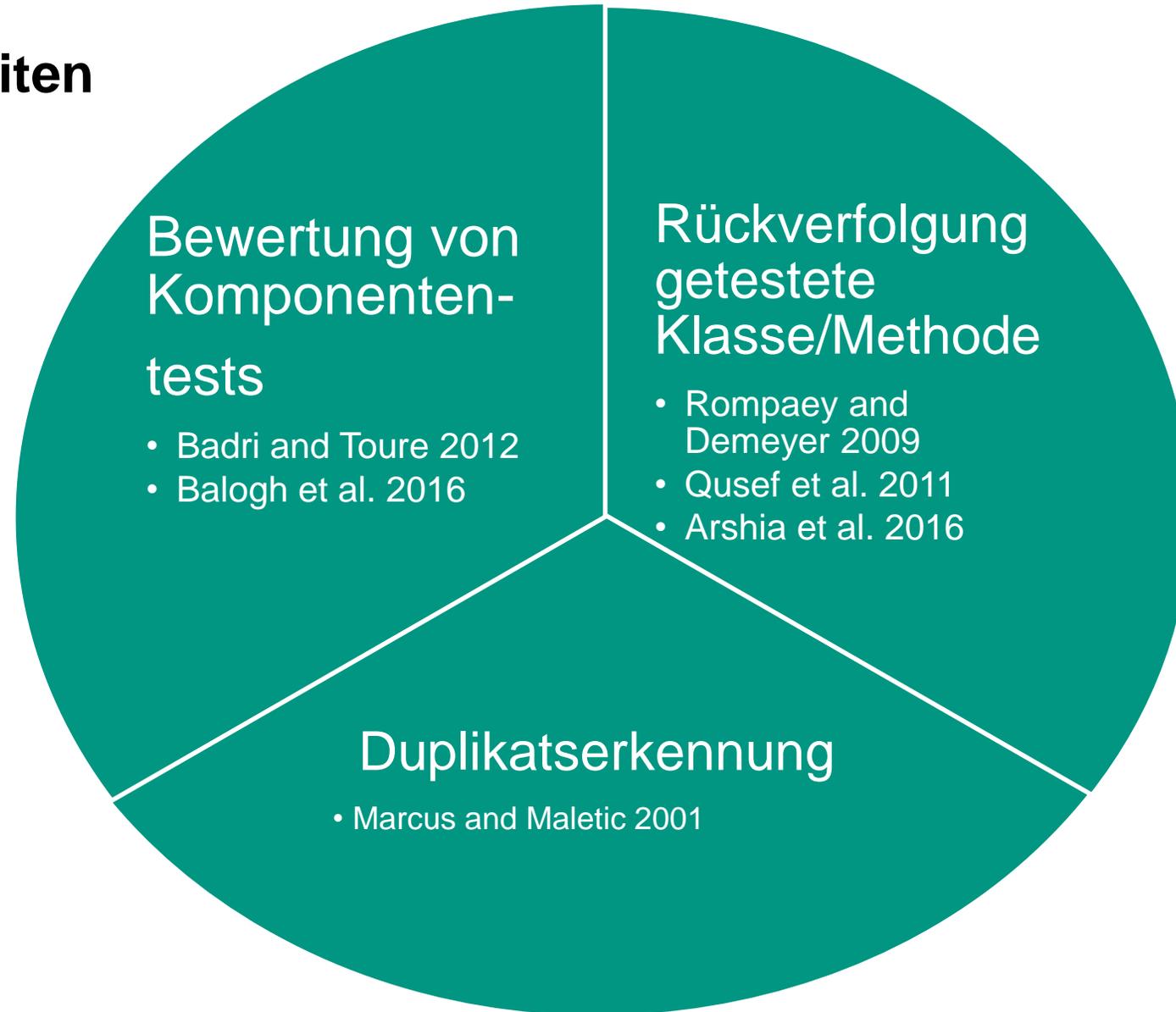


Beispiel

```
1 @Test
2 void testNearestUnit() {
3     Board b = parser.parseMap(Lists.newArrayList("#####", "# ..#", "#####"))
4         .getBoard();
5     Square s1 = b.squareAt(1, 1);
6     Square s2 = b.squareAt(2, 1);
7     Square result = Navigation.findNearest(Pellet.class, s1).getSquare();
8     assertThat(result).isEqualTo(s2);
9 }
```

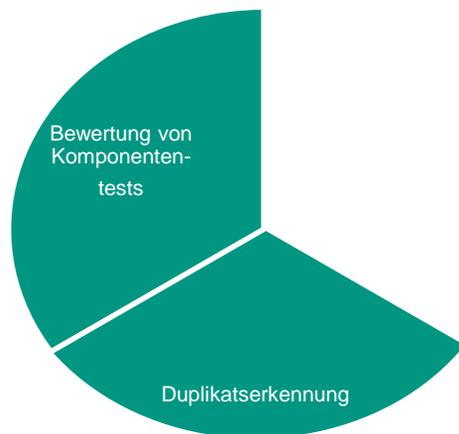
- `Navigation.findNearest` \Leftrightarrow `b.squareAt`
- `parser.parseMap` \Leftrightarrow `b.squareAt`

Verwandte Arbeiten



Rückverfolgung getestete Methode/Klasse

- Textuelle Verfahren
 - Dokumentation Antoniol et al. 2002
 - Dokumenten Csuvik, Kicsi, and Vidács 2019
- Semantische Verfahren
 - Datenfluss Qusef et al. 2011
 - Klassenvariablen Ghafari, Ghezzi, and Rubinov 2015
 - Quelltextmetriken Arshia, Mostafa, and Hadi 2016



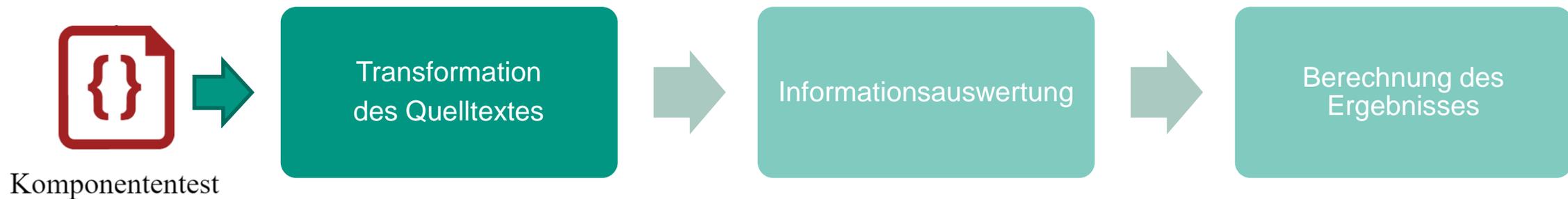
Ansatz

- Heuristisches Verfahren
- Nutzung von verschiedenen Teilverfahren
- Extraktion von Informationen aus dem Quelltext
- Semantische und Textuelle Verfahren



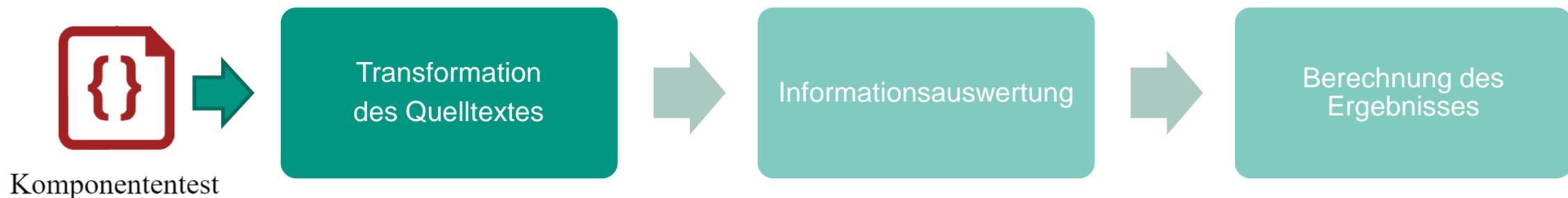
Transformation des Quelltextes

- Umwandlung des Quelltextes
- Aufteilung der Teilinformationen



Extrahierte Informationen

- Paketbezeichner
- Importanweisungen
- Methodename
- aufgerufene Methoden
- Überprüfungen
- Auf/Abbaumethode



Informationsauswertung: Überprüfung

```
1 @Test
2 void testNearestUnit() {
3     Board b = parser.parseMap(Lists.newArrayList("#####", "# ..#", "#####"))
4         .getBoard();
5     Square s1 = b.squareAt(1, 1);
6     Square s2 = b.squareAt(2, 1);
7     Square result = Navigation.findNearest(Pellet.class, s1).getSquare();
8     assertThat(result).isEqualTo(s2);
9 }
```



Komponententest



Transformation
des Quelltextes



Informationsauswertung



Berechnung des
Ergebnisses

Informationsauswertung: Methodename

- Textuell
- Existieren Konventionen
- Ähnlichkeitsalgorithmen
 - fastText
 - Cosinus
 - JaroWinkler
 - FuzzyScore
 - Längste Gleiche Sequenz
 - Einfache Gleichheit

testSubscriptionLookups



SubscriptionLookups



Subscription::lookup



Komponententest



Transformation
des Quelltextes



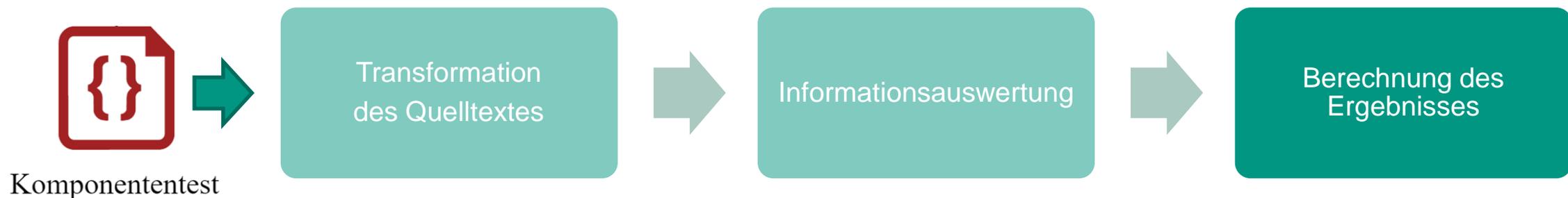
Informationsauswertung



Berechnung des
Ergebnisses

Berechnung des Ergebnisses

- Konfidenzwerte für jede Information
- Unterschiedlich genaue Teilergebnisse
- Summierung der Konfidenzwerte



Evaluation

- Rahmenbedingungen
 - JPacman, Gitiles und Roller
 - Gold Standard selbst erstellt
 - Parameter auf Teilmenge eingestellt
- Metriken
 - Präzision, Ausbeute und F_1 -Maß
 - Richtige, falsche und fehlende Methodenpaare

Roller

	Klassen	Methoden	Quelltextzeilen
Haupt	537	4616	47124
Test	50	222	5326
Gesamt	587	4838	52450

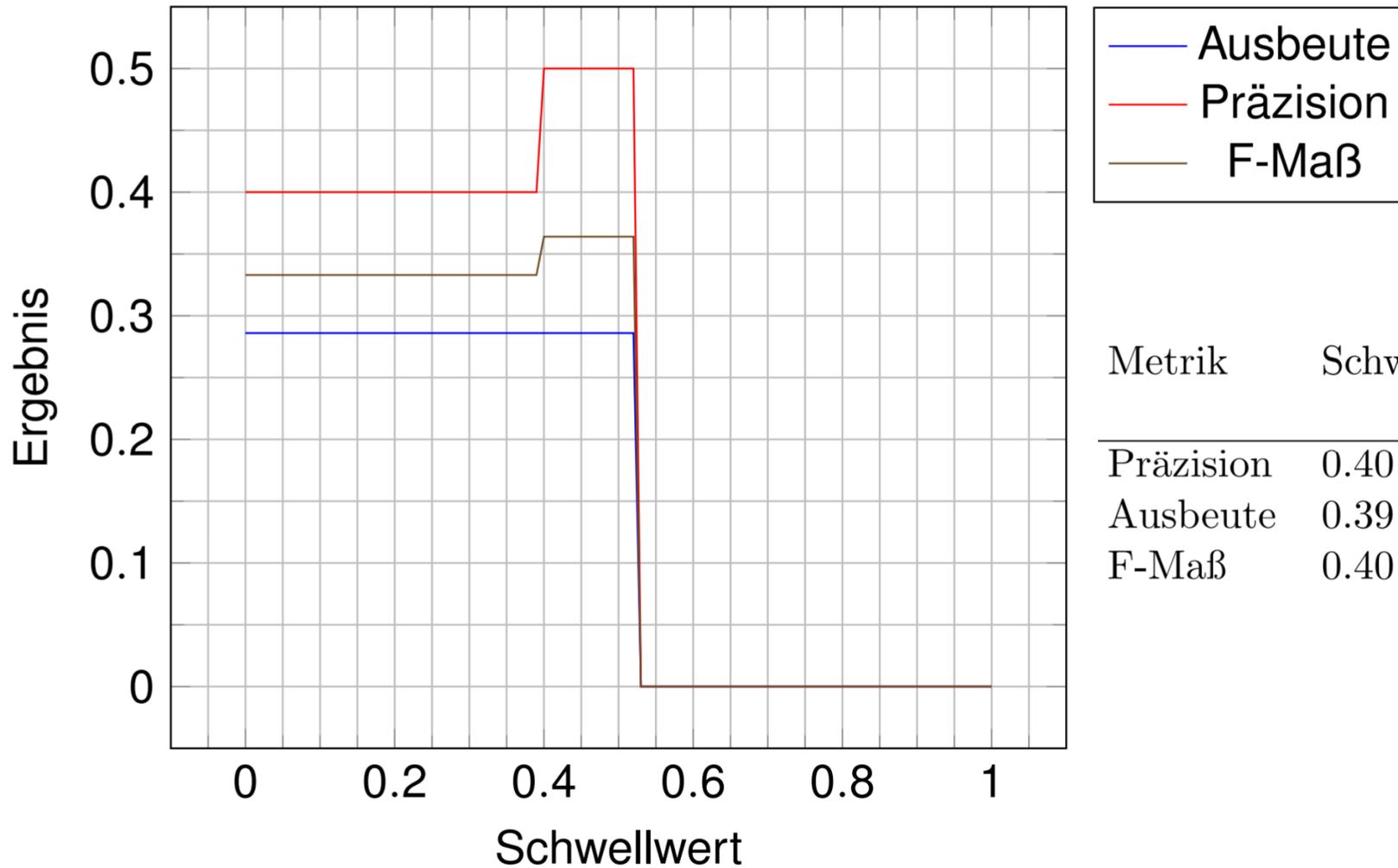
JPacman

	Klassen	Methoden	Quelltextzeilen
Haupt	46	201	2013
Test	14	54	613
Gesamt	60	255	2626

Gitiles

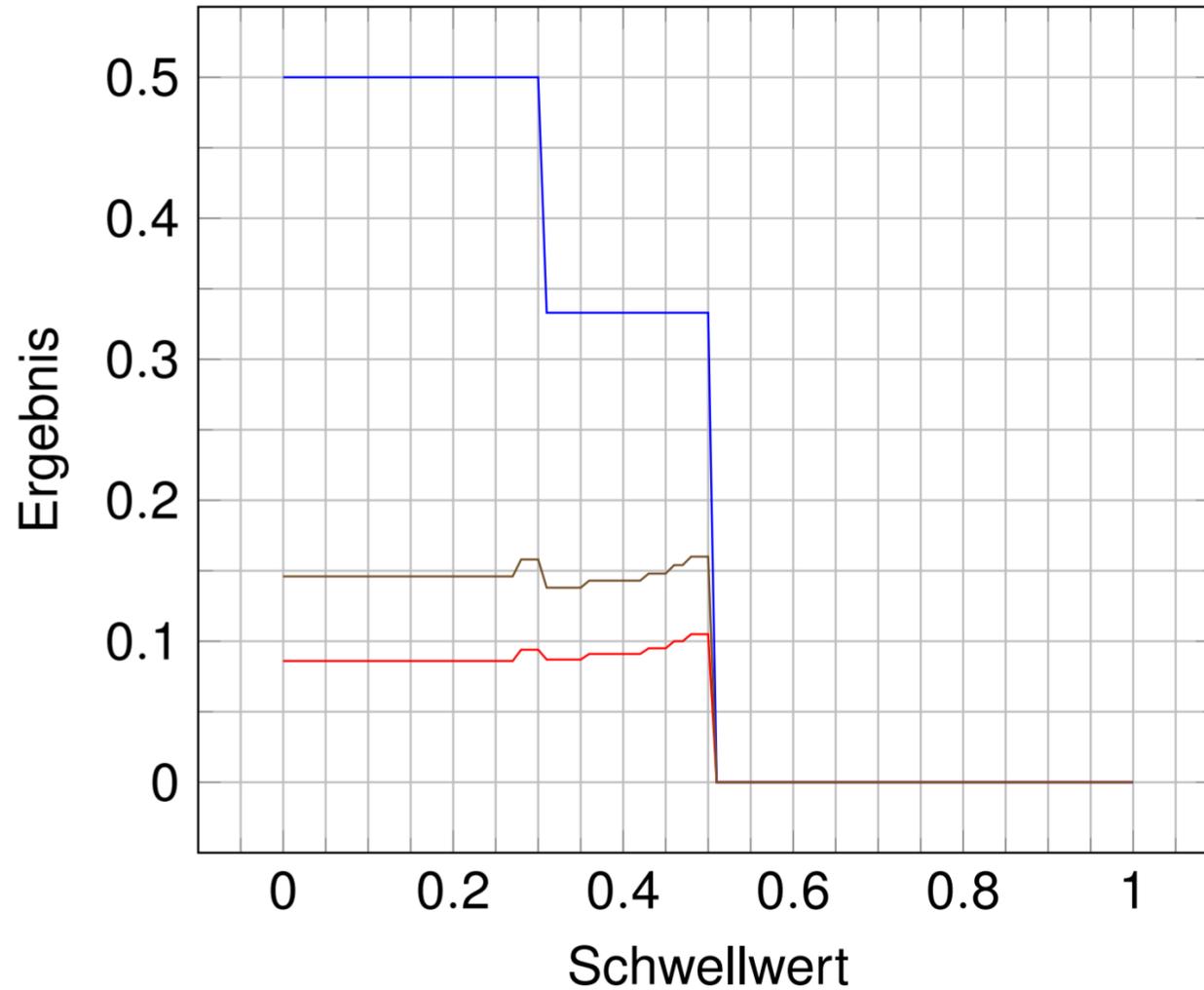
	Klassen	Methoden	Quelltextzeilen
Haupt	91	666	9812
Test	34	389	4835
Gesamt	125	1055	14647

JPacman Ergebnisse



Metrik	Schwellwert	Anzahl Methodenpaare		
		richtige	falsche	fehlende
Präzision	0.40	2	2	5
Ausbeute	0.39	2	3	5
F-Maß	0.40	2	2	5

Gitiles Ergebnisse



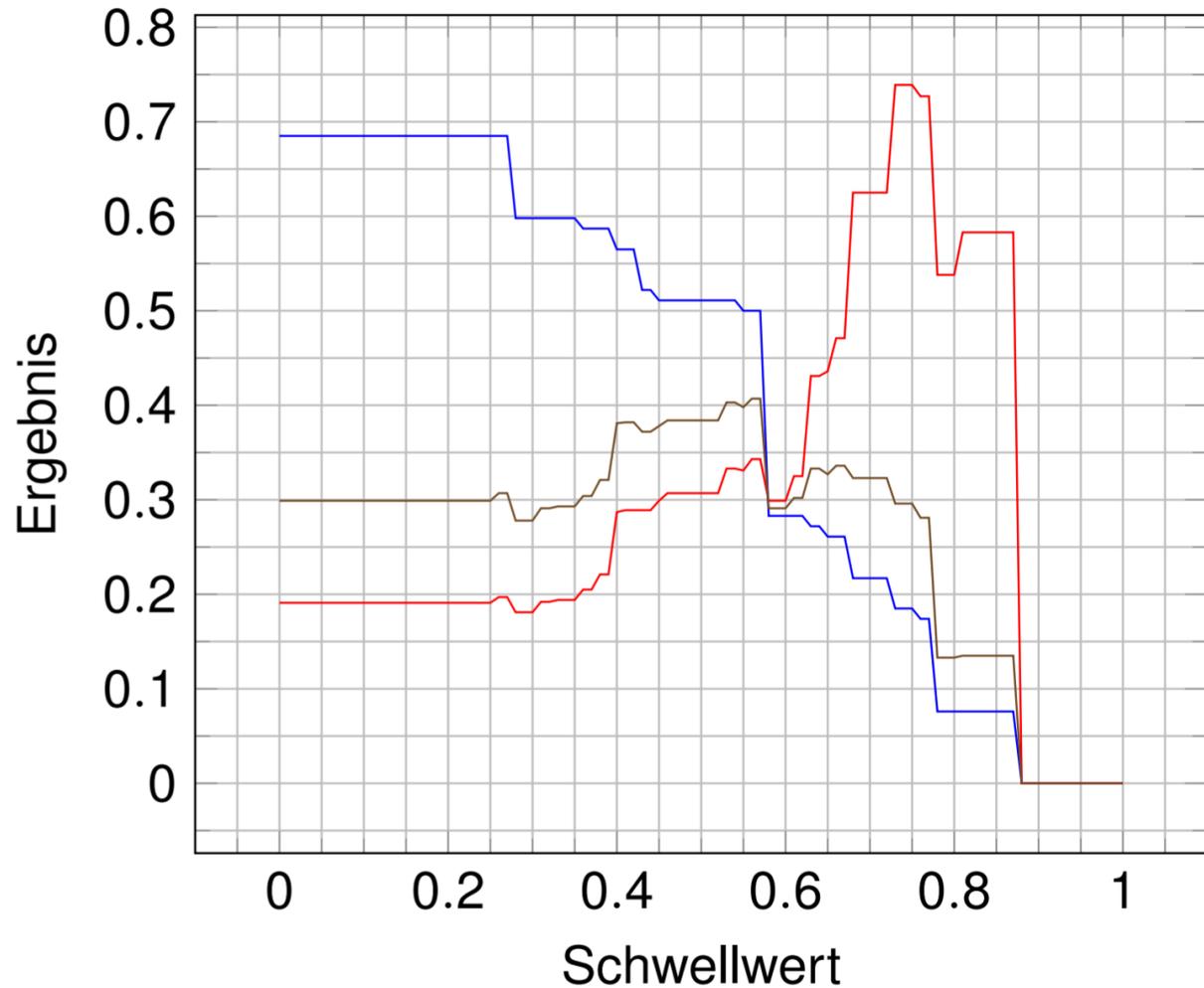
Metrik	Schwellwert	Anzahl Methodenpaare		
		richtige	falsche	fehlende
Präzision	0.48	2	17	4
Ausbeute	0.00	3	32	3
F-Maß	0.30	3	29	3

Gitiles Probleme

- Erstellermuster
- Modernes Java
- private Methoden
- Testattrappen

```
1 | @Test
2 | public void basicLog() throws Exception {
3 |     RevCommit commit = repo.branch("HEAD").commit().create();
4 |     Log response = buildJson(LOG, "/repo/+log");
5 |     assertThat(response.log).hasSize(1);
6 |     verifyJsonCommit(response.log.get(0), commit);
7 |     assertThat(response.log.get(0).treeDiff).isNull();
8 | }
```

Roller Ergebnisse



Metrik	Schwellwert	Anzahl Methodenpaare		
		richtige	falsche	fehlende
Präzision	0.74	17	6	75
Ausbeute	0.27	63	256	29
F-Maß	0.57	46	88	46

Zeichenkettenähnlichkeitsalgorithmenvergleich

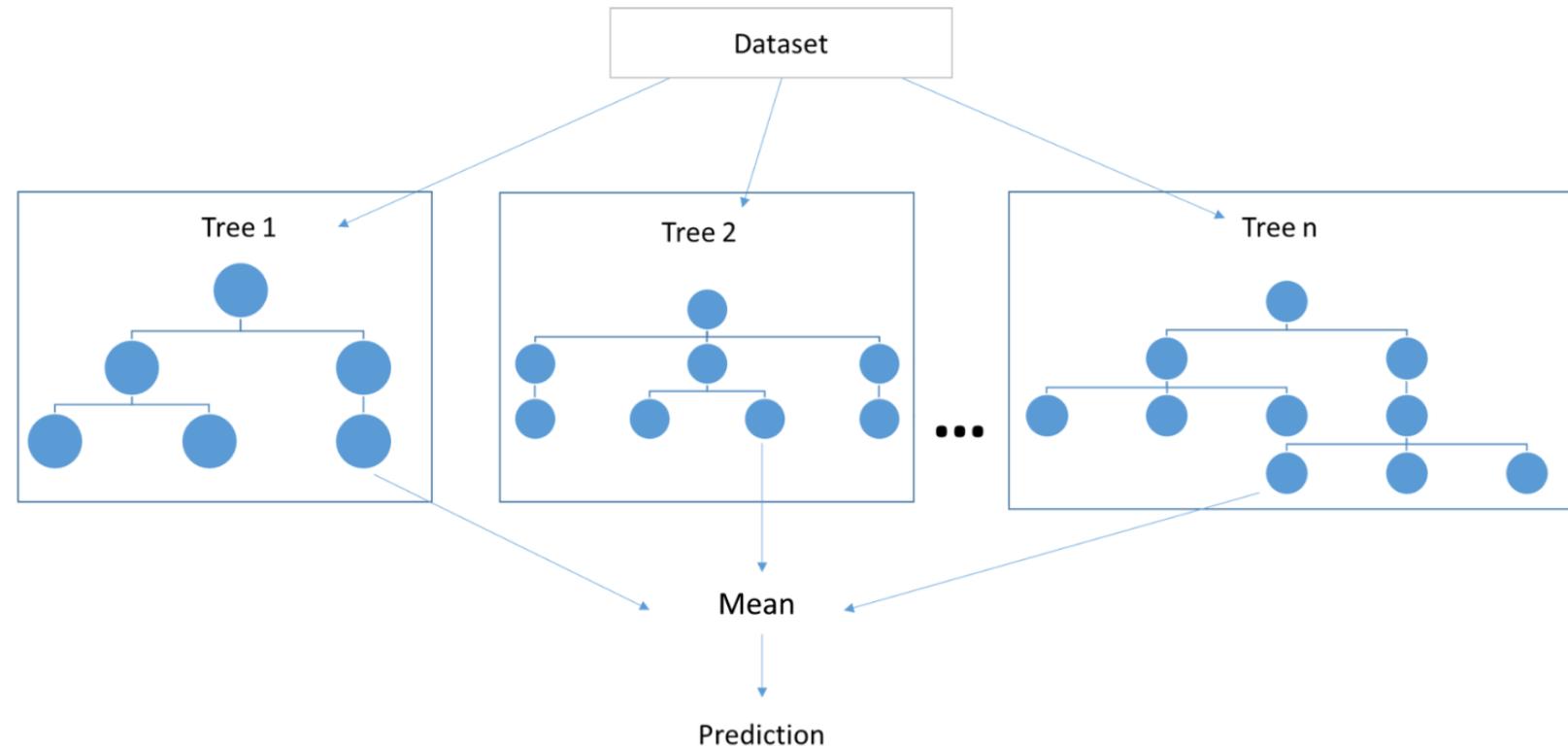
Algorithmus	Schwellwert 0.40			Schwellwert 0.60		
	Präzision	Ausbeute	F-Maß	Präzision	Ausbeute	F-Maß
fastText 1 Epoche	0.33	0.52	0.40	0.34	0.17	0.23
fastText 25 Epochen	0.32	0.51	0.40	0.34	0.19	0.24
Kosinus	0.21	0.49	0.29	0.28	0.15	0.20
FuzzyScore	0.21	0.55	0.30	0.33	0.36	0.34
JaroWinkler	0.20	0.67	0.31	0.41	0.50	0.45
Längste Gleiche Sequenz (LGS)	0.20	0.53	0.29	0.52	0.37	0.43
Gleichheit	0.37	0.42	0.40	0.35	0.10	0.15
JaroWinkler+LGS+fastText	0.22	0.67	0.33	0.41	0.53	0.46

Zusammenfassung

- Verwandte Arbeiten
- Verfahren
- Teilverfahren
- Evaluation
- Probleme
- Ausblick

Ausblick

- Testattrappen
- Erstellermuster
- Private Methoden
- Dokumentation
- Random Forest



Literatur

- [VRD09] Rompaey and Demeyer 2009
- [QBO+11] Qusef et al. 2011
- [AMH16] Arshia et al. 2016
- [MM01] Marcus and Maletic 2001
- [BT12] Badri and Toure 2012
- [BGBG16] Balogh et al. 2016
- [CKV19] Csuvik, Kicsi, and Vidács 2019
- [GGR15] Ghafari, Ghezzi, and Rubinov 2015

Informationsauswertung: Konfidenzwerte

Rückschluss	Konfidenzwert
Gleiches Paket	0.05
Ähnlicher Klassenname	0.2
Ähnlicher Methodename	0.3
Methoden in @Before	0.1
Methoden in @After	0.1
Felder in @Before	0.1
Felder in @After	0.1
Methoden die Felder nutzen in @Before	0.1
Methoden die Felder nutzen in @After	0.1
Methoden die das Objekt in der Überprüfung nutzen	0.2
Getestete Klasse in der Überprüfung	0.1
Getestete Methode in der Überprüfung	0.1

Teilverfahren 0.25

Teilverfahren	Schwellwert 0.25		
	Präzision	Ausbeute	F-Maß
überprüfte Methode/Klasse/Objekt	0.37	0.21	0.27
überprüfte Methode/Klasse/Objekt + Auf/Abbau	0.20	0.53	0.29
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassenname	0.17	0.55	0.26
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Methodennamen	0.18	0.60	0.28
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Paket/Methodenname	0.19	0.69	0.30

Teilverfahren 0.40

Teilverfahren	Schwellwert 0.40		
	Präzision	Ausbeute	F-Maß
überprüfte Methode/Klasse/Objekt	0.00	0.00	0.00
überprüfte Methode/Klasse/Objekt + Auf/Abbau	0.37	0.21	0.27
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassenname	0.25	0.29	0.27
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Methodennamen	0.29	0.52	0.37
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Paket/Methodennamen	0.29	0.57	0.38

Teilverfahren 0.60

Teilverfahren	Schwellwert 0.60		
	Präzision	Ausbeute	F-Maß
überprüfte Methode/Klasse/Objekt	0.00	0.00	0.00
überprüfte Methode/Klasse/Objekt + Auf/Abbau	0.00	0.00	0.00
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassenname	0.45	0.10	0.16
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Methodennamen	0.43	0.27	0.33
überprüfte Methode/Klasse/Objekt + Auf/Abbau + Klassen/Paket/Methodennamen	0.30	0.28	0.29

Informationsauswertung: Paketbezeichner

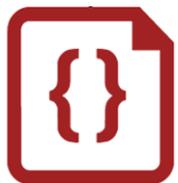
- Testklasse im selben Paket wie getestete Klasse
- Javakonvention



Informationsauswertung: Importanweisungen

- Geladenen Klassen

```
import org.jfree.chart.util;  
import org.jfree.chart.*;  
import static java.lang.Math.*;  
import static java.lang.System.out;
```



Komponententest



Transformation
des Quelltextes



Informationsauswertung



Berechnung des
Ergebnisses

Informationsauswertung: aufgerufene Methoden

- Genutzten Methoden



Informationsauswertung: Auf/Abbaumethoden

- Methodenname
- Zuweisungen
- Methodenaufrufe

