Universität Bremen
Fachbereich 3

Bachelorarbeit

im Studiengang Informatik

zur Erlangung des akademischen Grades
Bachelor of Science

Thema: Analyse der Evolution von Bad-Smells in Java-Projekten mithilfe der Repository-Mining-Bibliothek LibVCS4j

Autor: Max-Phillip Bahr <maxbahr@uni-bremen.de>
Matrikel-Nr. 4119656

Version vom: 28. März 2021

1. Gutachter: Prof. Dr. rer.nat. Rainer Koschke
2. Gutachterin: Dr.-Ing. Hui Shi
<table>
<thead>
<tr>
<th>Abbildungsverzeichnis</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Das Datenmodell des olfaction Servers</td>
<td>14</td>
</tr>
<tr>
<td>2. Anzahl analyserter Commits und Zeitspanne zwischen erstem und letzten Commit in Tagen</td>
<td>20</td>
</tr>
<tr>
<td>3. Anzahl gefundener Smells, getrennt nach Analyse-Software</td>
<td>20</td>
</tr>
<tr>
<td>4. Ergebnisse der Auswertung der Lebenszeit als Boxplot</td>
<td>25</td>
</tr>
<tr>
<td>5. Ergebnisse der Auswertung der Änderungen als Boxplot</td>
<td>30</td>
</tr>
<tr>
<td>6. Analysierte Commits mit Ausreißern</td>
<td>45</td>
</tr>
<tr>
<td>8. Ergebnisse der Auswertung der Änderungen als Boxplot mit Ausreißern</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Ergebnisse des Tukey-Tests zur Anzahl der gefundenen Smells.</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>2</td>
<td>Ergebnisse der Auswertung der Lebenszeit.</td>
</tr>
<tr>
<td>3</td>
<td>Ergebnisse des Tukey-Tests für die Auswertung der Lebenszeit.</td>
</tr>
<tr>
<td>4</td>
<td>Ergebnisse der Auswertung der Änderungen.</td>
</tr>
<tr>
<td>5</td>
<td>Signifikante Paare aus den Ergebnissen des Tukey-Tests zu Forschungsfrage 2.</td>
</tr>
<tr>
<td>6</td>
<td>Ergebnisse des Apriori Algorithmus, sortiert nach Lift.</td>
</tr>
<tr>
<td>7</td>
<td>Beispiele für exklusive Klon-Funde.</td>
</tr>
<tr>
<td>8</td>
<td>Die gesamten Analyseergebnisse.</td>
</tr>
<tr>
<td>9</td>
<td>Vollständige Ergebnisse des Tukey-Tests aus Forschungsfrage 2.</td>
</tr>
</tbody>
</table>
Listingverzeichnis

1 Die Datei addrepo.ps1 .................................. 17
2 Die Datei updaterepo.ps1 ................................. 17
3 Die initiale Datenabfrage ............................... 24
4 Die Abfrage der IDs .................................... 28
5 Die Abfrage der Änderungen ........................... 29
6 Die Abfrage der OIDs ................................... 32
7 Die Abfrage der Smell-Typen .......................... 33
Glossar

**CSV** *Comma-Separated Values*, ein Dateiformat bei dem Werte durch ein Trennzeichen, typischerweise ein Komma, getrennt werden.

**JSON** *JavaScript Object Notation*, "ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen [...] und zu generieren ist."\(^1\)

**OID** *Object Identifier*, eine einzigartige Bezeichnung eines Objekts in der Informatik.

**XML** *Extensible Markup Language* "ist ein einfaches, sehr flexibles Textformat, das von SGML (ISO 8879) abgeleitet wurde"\(^2\)

**UUID** *Universally Unique Identifier*, eine 128-bit Zahl die - egal wer sie wann und wo generiert - praktisch einzigartig ist und wie eine OID zur Bezeichnung von Objekten oder Informationen dient. UUIDs werden in RFC 4122 genauer beschrieben.\(^3\)

**GIN** *Generalized Inverted Index*, eine Form von Index die "dafür designed wurde, Fälle zu behandeln, in denen die zu indizierenden Daten sich aus mehreren Werten zusammensetzen und die Anfragen, die vom Index behandelt werden sollen, nach spezifischen Werten in diesen zusammengesetzten Daten suchen".\(^4\)

**HTTP** *Hypertext Transfer Protocol*, ein Protokoll zur Übertragung von Daten auf der Anwendungsebene, genauer beschrieben in RFC 2616.\(^5\)

**API** Ein *Application Programming Interface*, oder im Deutschen auch "eine Programmierschnittstelle dient dazu, Informationen zwischen einer Anwendung und einzelnen Programmteilen standardisiert auszutauschen."\(^6\)

\(^1\) [http://json.org](http://json.org), vgl. \[jso\]
\(^2\) [http://w3.org](http://w3.org), vgl. \[xml\]
\(^3\) Leach et al., vgl. \[LMS05\]
\(^4\) [http://postgresql.org](http://postgresql.org), vgl. \[?\]
\(^5\) Fielding et al., vgl. \[FGM+99\]
\(^6\) Luber, vgl. \[Lub17\]
1 Einleitung

1.1 Motivation


\(^7\)Gnoyke, vgl. [Gno16]
\(^8\)ISO-Standard, vgl. [ISO11] [S.4]
\(^9\)Fowler, vgl. [Fow18] [Preface: Acknowledgements]
\(^10\)Fowler, vgl. [Fow18] [Chapter 3: Bad Smells in Code]

1.2 Aufbau der Arbeit

2 Ziele der Arbeit

2.1 Ziele

Ziel #1: Analyse von rund 200 Java-Projekten

Ziel #2: Auswertung der Analysedaten zur Beantwortung von Forschungsfragen
Nach Abschluss der Analyse sollen gewonnene Daten ausgewertet und mit ihrer Hilfe mehrere Forschungsfragen beantwortet werden. Zum einen möchte ich eine Frage, die F. Becker bereits in seiner Bachelorarbeit im Zuge einer kleineren Analyse von 20 Java-Projekten beantwortet hat, aufgreifen.\textsuperscript{11}

- Wie lange leben bestimmte Arten von Bad-Smells im Durchschnitt?


Zum anderen möchte ich aber auch zwei Fragen beantworten, die Becker in seiner Arbeit nicht untersucht hat.

- Wie oft werden bestimmte Arten von Bad-Smells im Durchschnitt bearbeitet?
- Treten bestimmte Arten von Bad-Smells häufiger zusammen auf als andere?


\textsuperscript{11}Becker, vgl. [Bec20a] [S.35]
\textsuperscript{12}Arcelli Fontana et al. vgl. [AFWZ13] [S.7-8]
Ziel #3: Erweiterung existierender Software
Um die oben beschriebenen Forschungsfragen zu beantworten und die Analyse durchzuführen, bedarf es ein paar Änderungen an der existierenden Software.


- LibVCS4j besitzt weiterhin keine Möglichkeit, Code-Duplikationen zu erkennen. Hierfür sollen die Erkennungsprogramme CPD und IClones in LibVCS4j integriert werden.

- Um die Änderungen von Bad-Smells im Laufe ihrer Lebensdauer abspeichern zu können, soll das Datenmodell des olfaction Datenbankservers um ein Integer Element \textit{changes} erweitert werden, das die Anzahl an Änderungen wiedergibt, die ein Bad-Smell an einem präzisen Punkt in seiner Lebensspanne durchlaufen hat.

2.2 Nicht-Ziele
Es ist nicht Ziel dieser Arbeit Testcode zu analysieren. Es soll also nur der eigentliche Programmcode untersucht werden, da im Testcode in Testläufen vermehrt bestimmte Bad-Smells wie zum Beispiel Code-Duplikationen (\textit{Klone}) gefunden wurden.

Desweiteren ist es nicht Ziel dieser Arbeit, vorhandene Analyseergebnisse aus anderen Quellen in die Datenbank einzubinden. Es existieren zwar Analyseergebnisse, die eingebunden werden könnten, allerdings nicht in einer Form, die automatisch einlesbar ist. Das Einbinden müsste also per Hand erledigt werden und würde damit vom Ausmaß her den Rahmen dieser Arbeit sprengen.


3 Forschungsstand

Code-Smells sowie ihre Erkennung und Evolution sind bereits Gegenstand vielfacher wissenschaftlicher Untersuchungen.

AbuHassan et al. geben in ihrem Review-Artikel *Software smell detection techniques: A systematic literature review* eine Übersicht zu Erkennungstechniken, Tools und Studien, welche die Erkennung von Code-Smells untersuchen. Insgesamt wurden 145 Primärstudien untersucht, die insgesamt 52 Erkennungstools verwendet haben. 13


Nils Göde hat bei der AG Softwaretechnik im Rahmen seiner Diplomarbeit den inkrementellen Software-Klon Detektor IClopes entwickelt. Dieser "basiert auf dem

13AbuHassan et al., vgl. [AAG] [S.1, 4 & 26]
14Steinbeck, vgl. [Ste20] [S.1]
15Schulz, vgl. [Sch19] [S.6]
16Becker, vgl. [Bec20a] [S.46]
17Palomba et al., vgl. [PDNT]+15 [S.1]
18Becker, vgl. [Bec20a] [S.6]
Tool clones aus dem Projekt Bauhaus" und "wurde evaluiert, indem seine Performan-
ce mit clones verglichen wurde".\textsuperscript{19} IClones wurde außerdem von Göde und Rainer
Koschke in dem Paper \textit{Studying clone evolution using incremental clone detection}
verwendet, um die Evolution von Software-Klonen zu untersuchen. Die Forschungs-
fragen des Papers waren wie flüchtig Klone sind, wie oft Klone während ihrer Le-
bensspanne bearbeitet werden und ob Klone konsistent bearbeitet werden.\textsuperscript{20}

Tufano et al. haben in einer Studie die Evolution von verschiedenen Code-Smells
untersucht. Es wurden 200 Projekte mit insgesamt ca. 580.000 Commits untersucht.
Die Studie sollte die Fragen beantworten wann und warum Code-Smells in den Code
gelangen, wie hoch die Überlebenswahrscheinlichkeit von Code-Smells ist und wie
Entwickler Code-Smells entfernen. Tufano et al. kamen hier zu dem Ergebnis, dass
"80% aller Code-Smells am Ende des überblickten Entwicklungsfensters überlebt
haben und das nur 9% der entfernten Smells durch Refactoring entfernt wurden".\textsuperscript{21,22}
"Die Mehrheit der entfernten Code-Smell-Instanzen (40%) werden als Konsequenz
der Löschung des betroffenen Codes entfernt."\textsuperscript{23}

\textsuperscript{19}Göde, vgl. \cite{Goe08} \[S.7 & 75\]
\textsuperscript{20}Göde & Koschke, vgl. \cite{GK13} \[S.166\]
\textsuperscript{21}Tufano et al., vgl. \cite{TPB17} \[S.1, Abstract\]
\textsuperscript{22}Tufano et al., vgl. \cite{TPB17} \[S.14\]
\textsuperscript{23}Tufano et al., vgl. \cite{TPB17} \[S.17\]
4 Verwendete Software

In diesem Kapitel stelle ich die von mir für die Analyse verwendete Software vor.

olfaction:

LibVCS4j:

PMD:
PMD ist ein Tool zur Analyse von Programmcode auf Bad-Smells. PMD erkennt in Version 6.21.0 die wichtigsten Bad-Smells (z.B. GodClasses, DataClasses und verschiedene Arten von Dead Code) und unterstützt inkrementelle Analysen via Cache-Datei. PMD ist außerdem Open-Source, ein von Forschern anerkanntes Tool, bereits in LibVCS4j integriert und deckt alleine eine breite Menge an Bad-Smells ab, weshalb ich mich für diese Software entschieden habe. Der Quellcode der Software ist auf GitHub einsehbar.\textsuperscript{24}

CPD:
CPD ist genau wie PMD ein Analysetool, welches aber speziell den Bad-Smell Code Duplication (oder auch Software Klone) erkennt. CPD ist ein fester Teil von PMD, aber nicht direkt in LibVCS4j integriert, kann also nicht von LibVCS4j zur Analyse genutzt werden, dies werde ich im Laufe der Arbeit ändern.

IClones:
IClones ist ein von der AG Softwaretechnik an der Universität Bremen entwickeltes Analysetool, das wie CPD Software-Klone erkennen soll. IClones basiert wie CPD auf Token und besitzt auch wie CPD die Möglichkeit, Typ-1 und 2 Klone zu erken-

\textsuperscript{24}https://github.com/pmd/pmd
nen. Darüber hinaus kann IClones aber auch bis zu einer selbst eingestellten Varianz Typ-3 Klone (near-miss Klone) erkennen, indem es knappe Übereinstimmungen zusammenschließt. IClones ist wie CPD nicht in LibVCS4j integriert und kann noch nicht zur Analyse genutzt werden.
5 Erweiterung der Software

In diesem Kapitel beschreibe ich die Erweiterung der verwendeten Software.

5.1 olfaction


Abbildung 1: Das Datenmodell des olfaction Servers.\(^\text{25}\)

\(^\text{25}\) Becker, vgl. [Bec20b]
Ich habe mich aus Gründen der Abwärtskompatibilität für ein optionales Element entschieden, da die Universität Bremen, um die Daten zu persistieren, ihre Datenbanken verändern muss. Auf diesem Wege muss nur eine Spalte in der Datenbank hinzugefügt werden und alte Daten können behalten werden.


5.2 LibVCS4j

Um sowohl CPD als auch IClones mit LibVCS4j nutzbar zu machen, habe ich am Vorbild von LibVCS4js eingebauter Java-Klasse PMDRunner zwei weitere Klassen, CPDRunner\textsuperscript{26} und IClonesRunner\textsuperscript{27}, sowie deren Unterklassen erstellt. CPDRunner funktioniert wie PMDRunner und führt die main()-Methode vom eingebundenen CPD direkt aus. IClonesRunner hingegen kann dies nicht tun, weil IClones nicht Open-Source ist und deshalb nicht im Projekt enthalten sein darf. Dieses Problem habe ich gelöst, indem ich eine Umgebungsvariable ICLONES angelegt habe, die auf eine ausführbare .jar-Datei von IClones zeigt. Diese Variable wird von IClones-Runner geprüft und wenn die .jar-Datei gefunden wird, wird sie aus Java heraus ausgeführt. Auf diesem Wege kann jeder, der IClones nutzen darf, es selber in LibVCS4j einbinden.

Ich habe mich bei beiden Integrationen dafür entschieden, die eigentliche Analyse-Software XML ausgeben zu lassen, um die Implementierung konstant und das auslesen einfach zu halten. Bei der XML-Ausgabe von IClones fehlen jedoch genaue Positionsdaten. Es werden hier nur Zeilen angegeben, in denen der Klon zu finden ist, nicht genaue Spalten wie bei CPD oder PMD. Aus diesem Grund wird bei IClones immer die ganze Spalte mit in die Datenbank aufgenommen, um sicher zu stellen, dass der gesamte Klon enthalten ist. Ich habe mich bei IClones trotz dieser Limitierung für XML entschieden, da die XML-Parser-API SAX bereits in LibV-

\textsuperscript{26}https://github.com/mbahr94/libvcs4j/tree/cpdrunner
\textsuperscript{27}https://github.com/mbahr94/libvcs4j/tree/iclonesrunner
CS4j eingebunden ist. Um IClones’ *Rich Clone Format* (RCF) zu verwenden, hätte erst die RCF API in LibVCS4j eingebunden werden müssen. Hier wird also etwas Genauigkeit gegen eine einfachere Implementierung aufgewogen.
6 Analyse der Repositories

In diesem Kapitel wird die Analyse selbst, sowie eventuelle Probleme die auftraten, beschrieben.

6.1 Analyse

6.1.1 Repository Daten


```
(repository = Read-Host -Prompt 'Enter the repository name')

cd H:\Bachelorarbeit\Projects\$(repository)
git remote add olfaction "http://localhost:4040/repositories/$(repository).git"
git push olfaction --all

cd H:\Bachelorarbeit\Projects\`
```

Listing 1: Die Datei addrepo.ps1.

```
(repository = Read-Host -Prompt 'Enter the repository name')

cd H:\Bachelorarbeit\Projects\$(repository)
git pull
git push olfaction --all

cd H:\Bachelorarbeit\Projects\`
```

Listing 2: Die Datei updaterepo.ps1.

Das Analyseprogramm benutzt LibVCS4j und die im vorherigen Kapitel beschriebenen Klassen, um die Analyse auf kompletten Repositories auszuführen. Hierbei werden die folgenden Einstellungen verwendet.

6.1.2 PMD Regeln

PMD benutzt die folgenden Regeln bei der Suche nach Bad-Smells:

- GodClass
- DataClass
- SingularField
- UnusedPrivateField
- ExcessiveClassLength
• ExcessiveMethodLength

• ExcessiveParameterList

6.1.3 PMD Parameter

PMD benutzt zusätzlich zur Regelliste folgende Parameter:

-f xml Dieser Parameter setzt das Ausgabeformat von PMD. Hier wurde zum Beispiel bereits von Steinbeck XML festgelegt, um es anschließend einfach zu parsen.

-cache cachefile Dieser Parameter erlaubt eine inkrementelle Analyse eines Repositories mithilfe einer Cache-Datei. Dies steigert die Geschwindigkeit von PMD.

-t threads Dieser Parameter stellt die Anzahl von Threads ein, auf denen PMD läuft und erlaubt eine parallele Analyse mehrerer Dateien. Die Standardeinstellung des PMDRunners ist die Anzahl der verfügbaren Prozessoren.

6.1.4 CPD Parameter

CPD sucht Klone mit den folgenden Parametern:

–language java Dieser Parameter legt die Programmiersprache, die CPD untersucht, auf Java fest.

–minimum-tokens 100 Dieser Parameter legt die Mindestanzahl an Token fest, welche zwischen zwei Codestellen übereinstimmen müssen, damit ein Klon erkannt wird.


–encoding utf-8 Dieser Parameter setzt die Kodierung, die beim Untersuchen von Dateien verwendet wird, auf UTF-8.

–skip-lexical-errors Dieser Parameter sorgt dafür, dass Dateien, die aus irgendwelchen Gründen nicht tokenisiert werden können, ignoriert werden statt das Programm abzubrechen.


–ignore-identifiers Dieser Parameter lässt CPD beim Untersuchen von Dateien Identifier wie z.B. Variablennamen ignorieren.
–ignore-annotations Dieser Parameter lässt CPD beim Untersuchen von Dateien Sprachannotationen ignorieren.


6.1.5 IClones Parameter

IClones sucht Klone mit den folgenden Parametern:

-iformat single Dieser Parameter gibt an, in welchem Format die Repository-Daten geliefert werden und dient dazu, die inkrementelle Analyse von IClones zu deaktivieren, da LibVCS4j die dafür erforderliche Ordnerstruktur nicht automatisch erzeugen kann und IClones damit auf eine einfache Analyse beschränkt ist.

-minclone 100 Dieser Parameter verhält sich ähnlich wie –minimum-tokens bei CPD, es wird die Mindestanzahl an Tokens festgelegt, die übereinstimmen müssen, damit ein Klon erkannt wird.

-minblock 20 Dieser Parameter gibt die "minimale Länge von identischen Token-Sequenzen, die benutzt werden, um Near-Miss Klone zusammenzuführen" an.

-outformat xml Dieser Parameter funktioniert wie –format bei CPD und -f bei PMD. Das Ausgabeformat wird hier auf XML festgelegt, um ein einfaches Parsen der Ausgabe zu ermöglichen.

Neben den Analyseprogrammen, die durch LibVCS4j ausgeführt werden, berechnet mein Analyseprogramm auch die Anzahl der Änderungen, die Bad-Smells in ihrer Lebensspanne durchlaufen und speichert diese mit auf dem Server ab. Die Analyse umfasst 208 Repositories, von denen 206 komplett analysiert wurden. Die folgenden Boxplots beschreiben die Anzahl der analysierten Commits und die Zeitspanne zwischen dem ersten und letzten analysierten Commit in Tagen (Duration in days). Es ist außerdem wichtig zu erwähnen, dass es bei den analysierten Commits ebenfalls Ausreißer gibt, diese jedoch die Skala der Y-Achse soweit verschoben hätten, dass der Plot unleserlich geworden wäre. Ich habe mich deshalb dafür entschieden, die Ausreißer in diesem Plot nicht mit einzubeziehen, die Version mit Ausreißern kann im Anhang unter Boxplot A gefunden werden.

\[28\] AG Softwaretechnik Universität Bremen, vgl. [AS]
Abbildung 2: Anzahl analyserter Commits und Zeitspanne zwischen erstem und letztem Commit in Tagen.


Abbildung 3: Anzahl gefundener Smells, getrennt nach Analyse-Software.

Tabelle A im Anhang gibt eine komplette Übersicht über analysierte Repositories sowie gefundene Smells.

Hierfür habe ich zuerst die Nullhypothese "Es gibt keine statistisch signifikanten Unterschiede in der Anzahl der Code-Smells, die von den verschiedenen Detektoren gefunden wurden." aufgestellt und das Signifikanzniveau \( \alpha \) auf 0.05 festgelegt. Die Varianzanalyse ergab einen F-Wert von 7.519 und einen p-Wert von 0.00059. Da der p-Wert kleiner als \( \alpha = 0.05 \) ist, kann die Nullhypothese abgelehnt werden. Es existieren also statistisch signifikante Unterschiede in der Anzahl der Code-Smells, die jeweils von den verschiedenen Detektoren gefunden wurden. Wo genau diese Unterschiede liegen, zeigt der Tukey-Test. Die folgende Tabelle zeigt die Ergebnisse des Tukey-Tests.

<table>
<thead>
<tr>
<th>group1</th>
<th>group2</th>
<th>meandiff</th>
<th>p-adj</th>
<th>lower</th>
<th>upper</th>
<th>reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPD</td>
<td>IClones</td>
<td>-183367.9951</td>
<td>0.001</td>
<td>-300559.3857</td>
<td>-66176.6046</td>
<td>True</td>
</tr>
<tr>
<td>CPD</td>
<td>PMD</td>
<td>-145007.6942</td>
<td>0.0105</td>
<td>-262199.0847</td>
<td>-27816.3037</td>
<td>True</td>
</tr>
<tr>
<td>IClones</td>
<td>PMD</td>
<td>38360.301</td>
<td>0.7039</td>
<td>-78831.0895</td>
<td>155551.6915</td>
<td>False</td>
</tr>
</tbody>
</table>

Tabelle 1: Ergebnisse des Tukey-Tests zur Anzahl der gefundenen Smells.


### 6.2 Probleme

CPD und IClones konnten nicht auf allen Repositories zu Ende ausgeführt werden. CPD lief auf jruby nicht zu Ende und IClones sowohl auf jruby als auch auf jython. Die Programme froren im Laufe der Analyse ein, ohne Fehler zu melden. Als
Grund vermute ich die Anzahl und Größe der zu überprüfenden Dateien in Kombination mit limitierten Systemressourcen. Beide Repositories haben viele (teilweise große) Quellcode-Dateien und in jrubys Fall sogar stark verzweigte Unterordner. Dazu kommt, dass jeder Fund zusätzlich vom Analyseprogramm auf Änderungen untersucht werden muss. Hierfür wird jeder Bereich, in dem der Smell gefunden wurde, auf inhaltliche Änderungen untersucht, was zu starker Last führen kann. Aus diesem Grund wurden die beiden Repositories aus der oben beschriebenen Varianzanalyse und dem Tukey-Test ausgeschlossen, um das Ergebnis nicht zu verfälschen.


Das Repository jitwatch sollte ursprünglich auch analysiert werden, dies wurde jedoch aufgrund eines Fehlers sowohl unter Linux als auch unter Windows unmöglich gemacht. LibVCS4j meldet beim Auschecken der zweiten Revision bereits einen Checkout Konflikt, was allerdings keinen Sinn ergibt, da das Repository in einen neuen temporären Ordner ausgecheckt wird und die Dateinhalte nicht bearbeitet werden.
7 Auswertung der Analysedaten

In diesem Kapitel werde ich die in Kapitel 2 vorgestellten Forschungsfragen mithilfe der abgespeicherten Analysedaten beantworten.

- Wie lange leben bestimmte Arten von Bad-Smells im Durchschnitt?
- Wie oft werden bestimmte Arten von Bad-Smells im Durchschnitt bearbeitet?
- Treten bestimmte Arten von Bad-Smells häufiger zusammen auf als andere?


7.1 Forschungsfrage 1: Wie lange leben bestimmte Arten von Bad-Smells im Durchschnitt?


Um die abgefragten Daten auszuwerten wird als erstes aus den Commit-Daten des Repositories eine ISO 8601 Zeitspanne berechnet. Dann wird die ISO 8601 Zeitspanne, die im abgefragten `duration` Feld jeder Lebensspanne gespeichert ist, durch die für das Repository berechnete Zeitspanne geteilt, um herauszufinden, wie lange die Lebensspanne im Vergleich zum Repository existiert hat. Die Ergebnisse dieser Division werden dann als Dezimalzahl zwischen 0 und 1 in einer Liste abgelegt und, wenn es noch mehr Lebensspannen desselben Typs im Repository gibt, werden die nächsten 700 Lebensspannen vom Server abgefragt und das ganze wiederholt, bis es keine Lebensspannen mehr gibt. Aus den Dezimalzahlen in der Liste wird im An-
schluss der Durchschnitt und die Standardabweichung berechnet. Die Dezimalzahlen werden hierbei in Prozenten umgewandelt.

<table>
<thead>
<tr>
<th>Smell-Typ</th>
<th>Anzahl</th>
<th>Durchschnittliche Lebenszeit</th>
<th>Standardabweichung</th>
<th>Min. Lebenszeit</th>
<th>Max. Lebenszeit</th>
</tr>
</thead>
<tbody>
<tr>
<td>GodClass</td>
<td>6.866</td>
<td>23,43%</td>
<td>26,06%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
<tr>
<td>DataClass</td>
<td>3.018</td>
<td>25,84%</td>
<td>28,83%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
<tr>
<td>SingularField</td>
<td>3.217</td>
<td>12,00%</td>
<td>20,10%</td>
<td>0,00%</td>
<td>99,99%</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>4.331</td>
<td>11,00%</td>
<td>19,67%</td>
<td>0,00%</td>
<td>99,98%</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>2.450</td>
<td>22,12%</td>
<td>25,24%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>7.407</td>
<td>18,52%</td>
<td>24,81%</td>
<td>0,00%</td>
<td>99,99%</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>2.395</td>
<td>14,95%</td>
<td>18,81%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
<tr>
<td>cpd-clone</td>
<td>361.819</td>
<td>6,37%</td>
<td>14,90%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
<tr>
<td>iclones-clone</td>
<td>27.874</td>
<td>16,88%</td>
<td>24,56%</td>
<td>0,00%</td>
<td>100,00%</td>
</tr>
</tbody>
</table>

Tabelle 2: Ergebnisse der Auswertung der Lebenszeit.

Der folgende Boxplot stellt die Ergebnisse der Auswertung übersichtlich dar. Die Y-Achse stellt die prozentuale Lebensdauer der Smells dar und ist mit Dezimalzahlen von 0 bis 1 beschriftet, die Prozentsätze von 0% bis 100% darstellen. Ich habe mich hier für eine Version mit Ausreißern entschieden, da die Y-Achse von 0 bis 1 geht und der Plot auch mit Ausreißern gut lesbar ist. Ein Entfernen der Ausreißer würde hingegen die Skala verkleinern und weniger sinnvoll machen.

Abbildung 4: Ergebnisse der Auswertung der Lebenszeit als Boxplot.


Es ist bei dieser Art der Datenauswertung jedoch anzumerken, dass keine Aussage über die Zukunft getroffen werden kann. Das heißt, dass Lebensspannen, die in den letzten analysierten Commits eines Repositories zum ersten mal aufgetaucht sind, natürlich in der Berechnung eine eher kleine prozentuale Lebensdauer haben und so die Ergebnisse nach unten verfälschen könnten.

Um zu zeigen, dass die Unterschiede in der Lebensdauer der verschiedenen Code-Smell-Arten statistisch signifikant sind, wurden hier ebenfalls eine Varianzanalyse und ein Tukey-Test durchgeführt.

Hierzu habe ich die Nullhypothese ”Es gibt keine statistisch signifikanten Unterschiede zwischen den Lebensdauern der Smell-Arten” aufgestellt und das Signifikanzniveau $\alpha$ auf 0.05 festgelegt. Der errechnete p-Wert liegt bei 0.0 bei einem F-Wert von 3333.302, was bedeutet, dass der p-Wert kleiner ist, als ihn das Programm ausdrücken kann und damit auch kleiner als $\alpha = 0.05$. Somit ist gezeigt, dass es statistisch signifikante Unterschiede in den Lebensdauern der analysierten
Arten von Code-Smells gibt, die Nullhypothese wird also abgelehnt. Der Tukey-Test zeigt im folgenden, wo genau diese Unterschiede liegen.

Die folgende Tabelle zeigt die vollständigen Ergebnisse des Tukey-Tests zur ersten Forschungsfrage.

<table>
<thead>
<tr>
<th>group1</th>
<th>group2</th>
<th>meandiff</th>
<th>p-adj</th>
<th>lower</th>
<th>upper</th>
<th>reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataClass</td>
<td>ExcessiveClassLength</td>
<td>-0.0372</td>
<td>0.001</td>
<td>-0.0511</td>
<td>-0.0232</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveMethodLength</td>
<td>-0.0731</td>
<td>0.001</td>
<td>-0.0842</td>
<td>-0.0621</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveParameterList</td>
<td>-0.1089</td>
<td>0.001</td>
<td>-0.1229</td>
<td>-0.0949</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>GodClass</td>
<td>-0.0241</td>
<td>0.001</td>
<td>-0.0355</td>
<td>-0.0129</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>SingularField</td>
<td>-0.1384</td>
<td>0.001</td>
<td>-0.1514</td>
<td>-0.1255</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>UnusedPrivateField</td>
<td>-0.1483</td>
<td>0.001</td>
<td>-0.1605</td>
<td>-0.1362</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>cpd-clone</td>
<td>-0.1947</td>
<td>0.001</td>
<td>-0.204</td>
<td>-0.1853</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveMethodLength</td>
<td>-0.036</td>
<td>0.001</td>
<td>-0.0479</td>
<td>-0.0241</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveParameterList</td>
<td>-0.0717</td>
<td>0.001</td>
<td>-0.0865</td>
<td>-0.057</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>GodClass</td>
<td>0.013</td>
<td>0.0227</td>
<td>0.001</td>
<td>0.0251</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>SingularField</td>
<td>-0.1013</td>
<td>0.001</td>
<td>-0.115</td>
<td>-0.0876</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>UnusedPrivateField</td>
<td>-0.1112</td>
<td>0.001</td>
<td>-0.1241</td>
<td>-0.0982</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>cpd-clone</td>
<td>-0.1575</td>
<td>0.001</td>
<td>-0.1679</td>
<td>-0.1472</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>iclones-clone</td>
<td>-0.0525</td>
<td>0.001</td>
<td>-0.0633</td>
<td>-0.0417</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>ExcessiveParameterList</td>
<td>-0.0357</td>
<td>0.001</td>
<td>-0.0478</td>
<td>-0.0237</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>GodClass</td>
<td>0.049</td>
<td>0.001</td>
<td>0.0404</td>
<td>0.0576</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>SingularField</td>
<td>-0.0653</td>
<td>0.001</td>
<td>-0.0761</td>
<td>-0.0545</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>UnusedPrivateField</td>
<td>-0.0752</td>
<td>0.001</td>
<td>-0.085</td>
<td>-0.0654</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>cpd-clone</td>
<td>-0.1215</td>
<td>0.001</td>
<td>-0.1276</td>
<td>-0.1155</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>iclones-clone</td>
<td>-0.0165</td>
<td>0.001</td>
<td>-0.0232</td>
<td>-0.0098</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>GodClass</td>
<td>0.0848</td>
<td>0.001</td>
<td>0.0726</td>
<td>0.0969</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>SingularField</td>
<td>-0.0295</td>
<td>0.001</td>
<td>-0.0434</td>
<td>-0.0157</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>UnusedPrivateField</td>
<td>-0.0394</td>
<td>0.001</td>
<td>-0.0525</td>
<td>-0.0264</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>cpd-clone</td>
<td>-0.0858</td>
<td>0.001</td>
<td>-0.0963</td>
<td>-0.0753</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>iclones-clone</td>
<td>0.0193</td>
<td>0.001</td>
<td>0.0084</td>
<td>0.0302</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>SingularField</td>
<td>-0.1143</td>
<td>0.001</td>
<td>-0.1253</td>
<td>-0.1034</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>UnusedPrivateField</td>
<td>-0.1242</td>
<td>0.001</td>
<td>-0.1341</td>
<td>-0.1143</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>cpd-clone</td>
<td>-0.1706</td>
<td>0.001</td>
<td>-0.1768</td>
<td>-0.1643</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>iclones-clone</td>
<td>-0.0655</td>
<td>0.001</td>
<td>-0.0724</td>
<td>-0.0586</td>
<td>True</td>
</tr>
<tr>
<td>SingularField</td>
<td>UnusedPrivateField</td>
<td>-0.0099</td>
<td>0.1985</td>
<td>-0.0218</td>
<td>0.002</td>
<td>False</td>
</tr>
<tr>
<td>SingularField</td>
<td>cpd-clone</td>
<td>-0.0562</td>
<td>0.001</td>
<td>-0.0653</td>
<td>-0.0472</td>
<td>True</td>
</tr>
<tr>
<td>SingularField</td>
<td>iclones-clone</td>
<td>0.0488</td>
<td>0.001</td>
<td>0.0393</td>
<td>0.0583</td>
<td>True</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>cpd-clone</td>
<td>-0.0464</td>
<td>0.001</td>
<td>-0.0542</td>
<td>-0.0385</td>
<td>True</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>iclones-clone</td>
<td>0.0587</td>
<td>0.001</td>
<td>0.0403</td>
<td>0.0761</td>
<td>True</td>
</tr>
<tr>
<td>cpd-clone</td>
<td>iclones-clone</td>
<td>0.1051</td>
<td>0.001</td>
<td>0.1019</td>
<td>0.1082</td>
<td>True</td>
</tr>
</tbody>
</table>

Tabelle 3: Ergebnisse des Tukey-Tests für die Auswertung der Lebenszeit.

7.2 Forschungsfrage 2: Wie oft werden bestimmte Arten von Bad-Smells im Durchschnitt bearbeitet?


```java
final JSONObject analyzedLifespansResult = makeGraphQLRequest(
    new String.join(
        "\n",
        "query($analysis: String!, $kind: String!) {",
        "  analysis(name: $analysis) {",
        "    codeSmellLifespans(kind: $kind) {",
        "      edges {",
        "        node {",
        "          id",
        "        }",
        "      }",
        "    }",
        "  }",
        "}",
        "})",
    new JSONObject(
        Map.ofEntries(
            Map.entry("analysis", analysisName),
            Map.entry("kind", codeSmellKind)
        )
    ),
    client,
    endpoint
);
```

Listing 4: Die Abfrage der IDs.

Im Anschluss werden dann für jede ID in der Liste alle Code-Smell-Instanzen in der jeweiligen Lebensspanne abgefragt und die höchste Anzahl an Änderungen in der Lebensspanne ermittelt.
Die Abfrage der Daten für die Beantwortung von Forschungsfrage 2 inklusive Berechnung nahm insgesamt 1 Tag, 13 Stunden, 24 Minuten und 8 Sekunden in Anspruch. Da hier nicht mit Pagination gearbeitet wurde, kann genau gesagt werden, dass neun HTTP-Anfragen an den Server gestellt wurden, um die Listen der IDs der Lebensspannen zu generieren und 419.377 HTTP-Anfragen, um an die einzelnen Smells der Lebensspannen zu kommen und die Änderungen auszuwerten. Die Laufzeit ist aufgrund der vielen HTTP-Anfragen entsprechend lang.

Aus den höchsten Änderungen jeder Lebensspanne wird danach der Durchschnitt und die Standardabweichung berechnet.

<table>
<thead>
<tr>
<th>Smell-Typ</th>
<th>Anzahl</th>
<th>Durchschnittliche Anzahl Änderungen</th>
<th>Standardabweichung</th>
<th>Min. Änderungen</th>
<th>Max. Änderungen</th>
</tr>
</thead>
<tbody>
<tr>
<td>GodClass</td>
<td>6.866</td>
<td>15.458928</td>
<td>36.734061</td>
<td>0</td>
<td>881</td>
</tr>
<tr>
<td>DataClass</td>
<td>3.018</td>
<td>2.608550</td>
<td>11.664391</td>
<td>0</td>
<td>478</td>
</tr>
<tr>
<td>SingularField</td>
<td>3.217</td>
<td>0.000622</td>
<td>0.035262</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>4.331</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>2.450</td>
<td>22.447347</td>
<td>60.002946</td>
<td>0</td>
<td>1240</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>7.407</td>
<td>4.267855</td>
<td>9.266617</td>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>2.395</td>
<td>0.375783</td>
<td>1.324815</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>cpd-clone</td>
<td>361.819</td>
<td>0.174551</td>
<td>0.635378</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>iclones-clone</td>
<td>27.874</td>
<td>0.459676</td>
<td>1.057036</td>
<td>0</td>
<td>21</td>
</tr>
</tbody>
</table>

Tabelle 4: Ergebnisse der Auswertung der Änderungen.

Abbildung 5: Ergebnisse der Auswertung der Änderungen als Boxplot.


Auch bei diesen Ergebnissen gilt wieder, dass keine Aussage über die Zukunft getroffen werden kann. Genauer gesagt könnten auch hier Lebensspannen, die in späteren analysierten Commits auftreten, die Ergebnisse nach unten verfälschen.

29 Göde & Koschke, vgl. [GK13] [S.187-188]
Um statistisch signifikante Unterschiede in der Anzahl der Änderungen zwischen Smell-Arten zu zeigen, wurden auch bei dieser Forschungsfrage eine Varianzanalyse und ein Tukey-Test durchgeführt. Für die Varianzanalyse wurde die Nullhypothese "Es gibt keine statistisch signifikanten Unterschiede zwischen den Anzahlen der Änderungen der analysierten Smell-Arten" aufgestellt und das Signifikanzniveau $\alpha$ auf 0.05 festgelegt. Der errechnete p-Wert liegt hier ebenfalls bei 0.0 und der F-Wert bei 7776.617. Es gilt also auch hier, dass $p < \alpha$ und somit wird die Nullhypothese abgelehnt. Damit ist gezeigt, dass es statistisch signifikante Unterschiede bei der Anzahl der Änderungen zwischen den Smell-Arten gibt. Der Tukey-Test zeigt im folgenden, wo diese Unterschiede liegen.

Die folgende Tabelle zeigt die für diese Forschungsfrage signifikanten Paare bei denen im Tukey-Test die Nullhypothese abgelehnt wurde.

<table>
<thead>
<tr>
<th>group1</th>
<th>group2</th>
<th>meandiff</th>
<th>p-adj</th>
<th>lower</th>
<th>upper</th>
<th>reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataClass</td>
<td>ExcessiveClassLength</td>
<td>19.839</td>
<td>0.001</td>
<td>19.2667</td>
<td>20.4113</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveMethodLength</td>
<td>1.6595</td>
<td>0.001</td>
<td>1.205</td>
<td>2.114</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveParameterList</td>
<td>-2.2326</td>
<td>0.001</td>
<td>-2.8085</td>
<td>-1.6566</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>GodClass</td>
<td>12.8506</td>
<td>0.001</td>
<td>12.3909</td>
<td>13.3102</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>SingularField</td>
<td>-2.6077</td>
<td>0.001</td>
<td>-3.1411</td>
<td>-2.0744</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>UnusedPrivateField</td>
<td>-2.6083</td>
<td>0.001</td>
<td>-3.1074</td>
<td>-2.1093</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>cpd-clone</td>
<td>-2.4338</td>
<td>0.001</td>
<td>-2.8185</td>
<td>-2.0491</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>iclones-clone</td>
<td>-2.1487</td>
<td>0.001</td>
<td>-2.552</td>
<td>-1.7454</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveMethodLength</td>
<td>-18.1879</td>
<td>0.001</td>
<td>-18.67</td>
<td>-17.689</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveParameterList</td>
<td>-22.0716</td>
<td>0.001</td>
<td>-22.67</td>
<td>-21.468</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>GodClass</td>
<td>-6.9884</td>
<td>0.001</td>
<td>-7.4837</td>
<td>-6.4931</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>SingularField</td>
<td>-22.4467</td>
<td>0.001</td>
<td>-23.0111</td>
<td>-21.8824</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>UnusedPrivateField</td>
<td>-22.4473</td>
<td>0.001</td>
<td>-22.9794</td>
<td>-21.9135</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>cpd-clone</td>
<td>-22.2728</td>
<td>0.001</td>
<td>-22.6994</td>
<td>-21.8461</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>iclones-clone</td>
<td>-21.9877</td>
<td>0.001</td>
<td>-22.4312</td>
<td>-21.5442</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>ExcessiveParameterList</td>
<td>-3.8921</td>
<td>0.001</td>
<td>-4.3868</td>
<td>-3.3973</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>GodClass</td>
<td>11.1911</td>
<td>0.001</td>
<td>10.8385</td>
<td>11.5437</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>SingularField</td>
<td>-4.2672</td>
<td>0.001</td>
<td>-4.7116</td>
<td>-3.8228</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>UnusedPrivateField</td>
<td>-4.2679</td>
<td>0.001</td>
<td>-4.6704</td>
<td>-3.8633</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>cpd-clone</td>
<td>-4.0933</td>
<td>0.001</td>
<td>-4.3403</td>
<td>-3.8463</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>iclones-clone</td>
<td>-3.8082</td>
<td>0.001</td>
<td>-4.0833</td>
<td>-3.5331</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>GodClass</td>
<td>15.0831</td>
<td>0.001</td>
<td>14.5837</td>
<td>15.5826</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>SingularField</td>
<td>-15.4583</td>
<td>0.001</td>
<td>-15.908</td>
<td>-15.0086</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>UnusedPrivateField</td>
<td>-15.4589</td>
<td>0.001</td>
<td>-15.8673</td>
<td>-15.0505</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>cpd-clone</td>
<td>-15.2844</td>
<td>0.001</td>
<td>-15.5408</td>
<td>-15.028</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>iclones-clone</td>
<td>-14.9993</td>
<td>0.001</td>
<td>-15.2828</td>
<td>-14.7157</td>
<td>True</td>
</tr>
<tr>
<td>SingularField</td>
<td>iclones-clone</td>
<td>0.4591</td>
<td>0.0086</td>
<td>0.0672</td>
<td>0.851</td>
<td>True</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>iclones-clone</td>
<td>0.4597</td>
<td>0.0011</td>
<td>0.1159</td>
<td>0.8034</td>
<td>True</td>
</tr>
<tr>
<td>cpd-clone</td>
<td>iclones-clone</td>
<td>0.2851</td>
<td>0.001</td>
<td>0.1543</td>
<td>0.416</td>
<td>True</td>
</tr>
</tbody>
</table>


Insgesamt wurde die Nullhypothese für sieben Paare nicht abgelehnt. Eine Tabelle mit den vollständigen Ergebnissen ist im Anhang unter Tabelle B zu finden.
7.3 Forschungsfrage 3: Treten bestimmte Arten von Bad-Smells häufiger zusammen auf als andere?


```
final JSONObject analyzedCommitsResult = makeGraphQLRequest(
    String.join(
        "\n",
        "query($analysis: String!, $repository: String!) {
          analysis(name: $analysis) {
            analyzedCommits(repository: $repository) {
              edges {
                node {
                  oid,
                  }
              }
          }
        }\n    ),
    new JSONObject(
        Map.ofEntries(
            Map.entry("analysis", analysisName),
            Map.entry("repository", repoData[0])
        )
    ),
    client,
    endpoint
);  
```

Listing 6: Die Abfrage der OIDs.

Dann werden für jede OID alle Code-Smells gesucht, die in dem jeweiligen Commit vorkommen und die einzigartigen Typen der Code-Smells jeweils in eine neue Zeile des Datensets geschrieben.
Listing 7: Die Abfrage der Smell-Typen.

Das daraus resultierende Datenset wird im Anschluss im Python-Programm durch den Apriori Algorithmus ausgewertet. Die nachfolgende Tabelle stellt die Ergebnisse des Algorithmus dar, es ist jedoch wichtig zu erwähnen, dass die Commits der Repositories jruby und jython aufgrund der zuvor beschriebenen unvollständigen Analyse nicht in das Datenset mit aufgenommen wurden, da dies die Daten verfälschen würde.

Die Abfrage der Daten und das Schreiben des Datensets zur Beantwortung von Forschungsfrage 3 hat insgesamt 18 Stunden, 20 Minuten und 39 Sekunden in Anspruch genommen. Es wurde eine HTTP-Anfrage gestellt, um die OIDs der Commits zu sammeln und dann eine HTTP-Anfrage für jeden der 125.993 Commits, die gesammelt wurden.
Tabelle 6: Ergebnisse des Apriori Algorithmus, sortiert nach Lift.
Die gefundenen Regeln sind in der Tabelle nach Lift sortiert. Die Relation zwischen *UnusedPrivateField* und *SingularField* ist hierbei am interessantesten und sticht mit einem Lift-Wert von 1,454 stark hervor. Die beiden Smells kommen zusammen in 55,5% der Commits vor und die beiden Regeln treffen in ca. 90% der Fälle zu. Alle anderen Regeln bewegen sich in einem Lift-Bereich von 1.016 bis 1.183, was aufzeigt, dass einzelne Code-Smell-Typen sich nicht gegenseitig ausschließen, da in diesem Fall ein Lift-Wert von unter eins zu sehen wäre. Die unteren Regeln der Liste sind jedoch eher uninteressant, da der Lift-Wert gegen eins geht, was bedeutet das eher keine Relation zwischen den Typen vorliegt bzw. diese Typen nur schwach in Beziehung zueinander stehen.

Auch interessant ist die Relation zwischen CPD- und IClones-Klonen. Diese wurde vom Algorithmus im Hinblick auf das gesamte Datenset als eher schwach mit einem Lift-Wert von 1,029 beurteilt, obwohl die beiden Typen in 88,5% aller Commits zusammen zu finden sind und die Regeln zu je 97,8% und 93% zutreffen. Das Ergebnis lässt vermuten, dass CPD und IClones teilweise unterschiedliche Klone finden und nicht jeder Fund eines Klons von einem Programm auch vom anderen gemeldet wird. Dies bestätigt einen meiner vorherigen Verdacht und ist sehr gut in der vollständigen Tabelle der Analyseergebnisse im Anhang zu sehen (Tabelle A). Einige Repositories listen für IClones keine Funde, während für CPD Funde verzeichnet sind, und umgekehrt. Hier ist eine Auswahl an Beispielen dieser Repositories.

<table>
<thead>
<tr>
<th>CPD Fund → Kein IClones Fund</th>
<th>IClones Fund → Kein CPD Fund</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td>lambda-behave</td>
</tr>
<tr>
<td>libvcs4j</td>
<td>oryx</td>
</tr>
<tr>
<td>mapsforge</td>
<td>urnlib</td>
</tr>
</tbody>
</table>

Tabelle 7: Beispiele für exklusive Klon-Funde.

### 7.4 Performance der Datenbank


jedoch gegen den möglichen Flexibilitätsverlust aufgewogen werden, der damit einhergehen kann.\(^{30}\) Becker hat ebenfalls bei der Beantwortung seiner Forschungsfragen entdeckt, dass der Server, um die Performance zu verbessern "mit mehreren Instanzen und Lastverteilung" horizontal skaliert werden kann.\(^{31}\)

\(^{30}\) Becker, vgl. [Bec20a] [S.46]

\(^{31}\) Becker, vgl. [Bec20a] [S.37]
8 Fazit


Es wurden außerdem drei Forschungsfragen beantwortet. Die Themen der Fragen umfassten Lebensdauer von verschiedenen Bad-Smells, Änderungen von verschiedenen Bad-Smells und das gemeinsame Auftreten verschiedener einzelner Bad-Smells.


Bei der Beantwortung von Frage 3 wurde eine wahrscheinliche Beziehung zwischen
SingularField und UnusedPrivateField gefunden. Es wurde außerdem entdeckt, dass sich im Hinblick auf die Analysedaten keine einzelnen Code-Smell-Typen gegenseitig ausschließen. Auch interessant war die Relation von CPD-Klonen zu IClones-Klonen. Diese wurde vom Apriori Algorithmus als eher schwach eingestuft, was darauf schließen lässt, dass die Typen eher unabhängig voneinander sind, also jedes der beiden Erkennungstools exklusive Klone findet. Dieser Verdacht wurde dann mit einem Blick auf die Analyseergebnisse bestätigt und einige Beispiele für exklusive Klone-Funde genannt.


8.1 Ausblick


Beim olfaction Server sind die Anfragen für die Berechnung von Änderungen sehr langsam, da durch die einzelnen Instanzen einer Lebensspanne iteriert werden muss, um die höchste Anzahl an Änderungen zu finden. Hier könnte man noch ein changes Element auf der Lebensspannen Ebene hinzufügen um die Berechnung zu vereinfachen.

Des Weiteren könnte man sich genauer ansehen, welche Klone exklusiv von CPD und IClones gefunden werden und woran es genau liegt, dass bestimmte Klone von einer Software nicht erkannt werden.
Literaturverzeichnis

[AAG]  

[AFWZ13]  
Arcelli Fontana, Francesca ; Walter, Bartosz ; Zanoni, Marco: Code Smells and Micro Patterns Correlations, 2013

[AS]  

[Bec20a]  
Becker, Felix: Development of a Server for Code Smell Data. 2 2020

[Bec20b]  

[FGM+99]  

[Fow18]  

[Gö08]  
Göde, Nils: Incremental Clone Detection. 9 2008

[GK13]  

[Gno16]  

[ISO11]  

[jso]  
Einführung in JSON. https://www.json.org/json-de.html Abruf: 22.02.2021

[LMS05]  

[PDNT+15] PALOMBA, Fabio; DI Nucci, Dario; TUFANO, Michele; BAVOTA, Gabriele; OLIVETO, Rocco; POSHYVANYK, Denys; LUCIA, Andrea: Landfill: an Open Dataset of Code Smells with Public Evaluation, 2015


[TPB+17] TUFANO, Michele; PALOMBA, Fabio; BAVOTA, Gabriele; OLIVETO, Rocco; DI PENTA, Massimiliano; LUCIA, Andrea; POSHYVANYK, Denys: When and Why Your Code Starts to Smell Bad (and Whether the Smells Go Away). In: IEEE Transactions on Software Engineering PP (2017), 01, S. 1–1. [http://dx.doi.org/10.1109/TSE.2017.2653105]. – DOI 10.1109/TSE.2017.2653105

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Analyzierte Commits</th>
<th>PMD Smells</th>
<th>CPD Smells</th>
<th>IClones Smells</th>
<th>Gesamt Smells</th>
</tr>
</thead>
<tbody>
<tr>
<td>adt4j</td>
<td>67</td>
<td>179</td>
<td>633</td>
<td>756</td>
<td>1568</td>
</tr>
<tr>
<td>arc4j</td>
<td>2051</td>
<td>15455</td>
<td>70734</td>
<td>16545</td>
<td>122722</td>
</tr>
<tr>
<td>agrona</td>
<td>196</td>
<td>3602</td>
<td>12732</td>
<td>11711</td>
<td>28045</td>
</tr>
<tr>
<td>airline</td>
<td>62</td>
<td>243</td>
<td>160</td>
<td>139</td>
<td>542</td>
</tr>
<tr>
<td>almanac-converter</td>
<td>144</td>
<td>1737</td>
<td>2041</td>
<td>43</td>
<td>3821</td>
</tr>
<tr>
<td>arangodb-java-driver</td>
<td>484</td>
<td>18092</td>
<td>39906</td>
<td>2210</td>
<td>59543</td>
</tr>
<tr>
<td>Arc4j</td>
<td>15</td>
<td>901</td>
<td>1967</td>
<td>75</td>
<td>2162</td>
</tr>
<tr>
<td>asyncc-http-client</td>
<td>580</td>
<td>16716</td>
<td>19688</td>
<td>2096</td>
<td>38500</td>
</tr>
<tr>
<td>attic-aurora</td>
<td>935</td>
<td>14043</td>
<td>12091</td>
<td>2140</td>
<td>28274</td>
</tr>
<tr>
<td>auto</td>
<td>80</td>
<td>16</td>
<td>141</td>
<td>0</td>
<td>157</td>
</tr>
<tr>
<td>autonomic</td>
<td>80</td>
<td>34</td>
<td>2</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>azure</td>
<td>754</td>
<td>33102</td>
<td>6926</td>
<td>4125</td>
<td>44153</td>
</tr>
<tr>
<td>availability</td>
<td>150</td>
<td>226</td>
<td>61</td>
<td>81</td>
<td>368</td>
</tr>
<tr>
<td>big釜ne</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>blade</td>
<td>258</td>
<td>7799</td>
<td>17645</td>
<td>2619</td>
<td>28063</td>
</tr>
<tr>
<td>BiotaFaces-OSP</td>
<td>1066</td>
<td>81139</td>
<td>1682745</td>
<td>94317</td>
<td>1858261</td>
</tr>
<tr>
<td>boot</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>caffeine</td>
<td>426</td>
<td>1292</td>
<td>3473</td>
<td>3553</td>
<td>8298</td>
</tr>
<tr>
<td>cglib</td>
<td>37</td>
<td>435</td>
<td>545</td>
<td>148</td>
<td>1128</td>
</tr>
<tr>
<td>checkstyle</td>
<td>2739</td>
<td>100627</td>
<td>725361</td>
<td>22908</td>
<td>848906</td>
</tr>
<tr>
<td>Chronicl-Map</td>
<td>1140</td>
<td>27245</td>
<td>26938</td>
<td>16224</td>
<td>70707</td>
</tr>
<tr>
<td>Clojure</td>
<td>60</td>
<td>2710</td>
<td>15025</td>
<td>18185</td>
<td>36370</td>
</tr>
<tr>
<td>CodenameOne</td>
<td>2440</td>
<td>994786</td>
<td>1572761</td>
<td>336464</td>
<td>2904011</td>
</tr>
<tr>
<td>cogcomp-nlp</td>
<td>90</td>
<td>1916</td>
<td>4950</td>
<td>3052</td>
<td>9918</td>
</tr>
<tr>
<td>commons-csv</td>
<td>711</td>
<td>2009</td>
<td>1239</td>
<td>147</td>
<td>3395</td>
</tr>
<tr>
<td>commons-lang</td>
<td>1992</td>
<td>98922</td>
<td>361990</td>
<td>43869</td>
<td>586781</td>
</tr>
<tr>
<td>concurrent-commit</td>
<td>32</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>config</td>
<td>317</td>
<td>4355</td>
<td>1142</td>
<td>951</td>
<td>6448</td>
</tr>
<tr>
<td>consul-api</td>
<td>149</td>
<td>5315</td>
<td>15601</td>
<td>391</td>
<td>21307</td>
</tr>
<tr>
<td>cqengine</td>
<td>107</td>
<td>1424</td>
<td>6978</td>
<td>3970</td>
<td>12372</td>
</tr>
<tr>
<td>crawlerj</td>
<td>37</td>
<td>786</td>
<td>283</td>
<td>0</td>
<td>1054</td>
</tr>
<tr>
<td>cucumber-jvm</td>
<td>720</td>
<td>4093</td>
<td>5199</td>
<td>308</td>
<td>10108</td>
</tr>
<tr>
<td>cubes-rest</td>
<td>38</td>
<td>33</td>
<td>110</td>
<td>8</td>
<td>151</td>
</tr>
<tr>
<td>cyclops</td>
<td>226</td>
<td>11178</td>
<td>158412</td>
<td>111240</td>
<td>280830</td>
</tr>
<tr>
<td>derived4j</td>
<td>85</td>
<td>280</td>
<td>1027</td>
<td>1949</td>
<td>3256</td>
</tr>
<tr>
<td>Design</td>
<td>234</td>
<td>5402</td>
<td>1572</td>
<td>2328</td>
<td>5982</td>
</tr>
<tr>
<td>docker</td>
<td>235</td>
<td>5572</td>
<td>8038</td>
<td>691</td>
<td>14211</td>
</tr>
<tr>
<td>drjava</td>
<td>2795</td>
<td>455831</td>
<td>697741</td>
<td>514969</td>
<td>1668541</td>
</tr>
<tr>
<td>dropwizard-circuitbreaker</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>eclipse-collections</td>
<td>14</td>
<td>797</td>
<td>22463</td>
<td>6375</td>
<td>29665</td>
</tr>
<tr>
<td>Ehcache</td>
<td>3</td>
<td>9</td>
<td>19</td>
<td>5</td>
<td>33</td>
</tr>
<tr>
<td>error-prone</td>
<td>2105</td>
<td>28685</td>
<td>543662</td>
<td>31828</td>
<td>662355</td>
</tr>
<tr>
<td>eureka</td>
<td>252</td>
<td>2699</td>
<td>788</td>
<td>1446</td>
<td>4933</td>
</tr>
<tr>
<td>failures</td>
<td>265</td>
<td>230</td>
<td>1756</td>
<td>284</td>
<td>2190</td>
</tr>
<tr>
<td>FakeSMTP</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>fast-xml-serialization</td>
<td>389</td>
<td>12151</td>
<td>12821</td>
<td>1820</td>
<td>34204</td>
</tr>
<tr>
<td>fastjson</td>
<td>446</td>
<td>60446</td>
<td>175758</td>
<td>134108</td>
<td>352152</td>
</tr>
<tr>
<td>fast-pax</td>
<td>28</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>feather</td>
<td>21</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>feign</td>
<td>189</td>
<td>1041</td>
<td>15</td>
<td>40</td>
<td>1096</td>
</tr>
<tr>
<td>fileusage</td>
<td>9550</td>
<td>285581</td>
<td>4186405</td>
<td>115311</td>
<td>6407817</td>
</tr>
<tr>
<td>fixture-factory</td>
<td>31</td>
<td>16</td>
<td>30</td>
<td>30</td>
<td>76</td>
</tr>
<tr>
<td>flatbuffers</td>
<td>80</td>
<td>159</td>
<td>17</td>
<td>62</td>
<td>238</td>
</tr>
<tr>
<td>flexy-pool</td>
<td>79</td>
<td>123</td>
<td>0</td>
<td>0</td>
<td>121</td>
</tr>
<tr>
<td>flyingcairn</td>
<td>86</td>
<td>9589</td>
<td>79764</td>
<td>5871</td>
<td>95224</td>
</tr>
<tr>
<td>FlexBuilder</td>
<td>389</td>
<td>4433</td>
<td>48036</td>
<td>146837</td>
<td>69016</td>
</tr>
<tr>
<td>frescod</td>
<td>15388</td>
<td>3653874</td>
<td>4706074</td>
<td>1434482</td>
<td>9814430</td>
</tr>
<tr>
<td>gate-core</td>
<td>228</td>
<td>76294</td>
<td>1183427</td>
<td>352161</td>
<td>1611882</td>
</tr>
<tr>
<td>governor</td>
<td>202</td>
<td>916</td>
<td>759</td>
<td>334</td>
<td>2009</td>
</tr>
<tr>
<td>graphhopper</td>
<td>1756</td>
<td>57408</td>
<td>81689</td>
<td>34834</td>
<td>150931</td>
</tr>
<tr>
<td>greendia</td>
<td>1462</td>
<td>9877</td>
<td>10115</td>
<td>2582</td>
<td>22574</td>
</tr>
<tr>
<td>gson</td>
<td>745</td>
<td>4152</td>
<td>8849</td>
<td>2861</td>
<td>15862</td>
</tr>
<tr>
<td>guava</td>
<td>3767</td>
<td>187287</td>
<td>3867299</td>
<td>280387</td>
<td>4334973</td>
</tr>
<tr>
<td>guice</td>
<td>351</td>
<td>4201</td>
<td>5416</td>
<td>1033</td>
<td>10650</td>
</tr>
<tr>
<td>h2database</td>
<td>6763</td>
<td>1403654</td>
<td>1697071</td>
<td>376847</td>
<td>3678272</td>
</tr>
<tr>
<td>hibernate</td>
<td>480</td>
<td>9769</td>
<td>2297</td>
<td>0</td>
<td>12896</td>
</tr>
<tr>
<td>hibernate-orm</td>
<td>4050</td>
<td>1783774</td>
<td>6381484</td>
<td>476408</td>
<td>8641666</td>
</tr>
<tr>
<td>HikariJSON</td>
<td>40</td>
<td>36</td>
<td>35</td>
<td>50</td>
<td>121</td>
</tr>
<tr>
<td>hotspot</td>
<td>48</td>
<td>687</td>
<td>1308</td>
<td>59</td>
<td>2054</td>
</tr>
<tr>
<td>HotswapAgent</td>
<td>163</td>
<td>1216</td>
<td>27602</td>
<td>10602</td>
<td>50309</td>
</tr>
<tr>
<td>Repository Name</td>
<td>Analyseierte Commits</td>
<td>PMD Smells</td>
<td>CPD Smells</td>
<td>IClones Smells</td>
<td>Smells Gesamt</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------</td>
<td>------------</td>
<td>------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Hystrix</td>
<td>50</td>
<td>1287</td>
<td>3550</td>
<td>680</td>
<td>5437</td>
</tr>
<tr>
<td>soot</td>
<td>486</td>
<td>22692</td>
<td>27483</td>
<td>1782</td>
<td>59877</td>
</tr>
<tr>
<td>imgcadr</td>
<td>68</td>
<td>301</td>
<td>49</td>
<td>37</td>
<td>387</td>
</tr>
<tr>
<td>g2objc</td>
<td>1646</td>
<td>61600</td>
<td>566738</td>
<td>5319</td>
<td>633657</td>
</tr>
<tr>
<td>jdbct</td>
<td>148</td>
<td>30</td>
<td>43</td>
<td>0</td>
<td>73</td>
</tr>
<tr>
<td>jackson-dataforms-text</td>
<td>50</td>
<td>722</td>
<td>244</td>
<td>0</td>
<td>1206</td>
</tr>
<tr>
<td>jackson-dataformat-money</td>
<td>44</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>jacoco</td>
<td>404</td>
<td>1087</td>
<td>441</td>
<td>0</td>
<td>1528</td>
</tr>
<tr>
<td>jaxms</td>
<td>758</td>
<td>41548</td>
<td>24098</td>
<td>16</td>
<td>65572</td>
</tr>
<tr>
<td>jacksonw</td>
<td>4</td>
<td>87</td>
<td>2048</td>
<td>60</td>
<td>2145</td>
</tr>
<tr>
<td>jaxrs</td>
<td>628</td>
<td>208635</td>
<td>128648</td>
<td>157753</td>
<td></td>
</tr>
<tr>
<td>javamelody</td>
<td>1540</td>
<td>27590</td>
<td>61843</td>
<td>3717</td>
<td>93150</td>
</tr>
<tr>
<td>javaparser</td>
<td>378</td>
<td>8949</td>
<td>1118572</td>
<td>66980</td>
<td>1194481</td>
</tr>
<tr>
<td>javapaint</td>
<td>200</td>
<td>792</td>
<td>602</td>
<td>149</td>
<td>1543</td>
</tr>
<tr>
<td>jasmin</td>
<td>683</td>
<td>31232</td>
<td>67837</td>
<td>12013</td>
<td>111082</td>
</tr>
<tr>
<td>javaVerbalExpressions</td>
<td>32</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>jdoc</td>
<td>31</td>
<td>758</td>
<td>4382</td>
<td>36</td>
<td>5176</td>
</tr>
<tr>
<td>jcalend</td>
<td>121</td>
<td>864</td>
<td>66</td>
<td>33</td>
<td>963</td>
</tr>
<tr>
<td>jCTools</td>
<td>333</td>
<td>2469</td>
<td>12965</td>
<td>11601</td>
<td>27035</td>
</tr>
<tr>
<td>jeso</td>
<td>519</td>
<td>7451</td>
<td>102829</td>
<td>7096</td>
<td>134838</td>
</tr>
<tr>
<td>jinu</td>
<td>3</td>
<td>516</td>
<td>18</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>jeromq</td>
<td>100</td>
<td>738</td>
<td>547</td>
<td>0</td>
<td>1285</td>
</tr>
<tr>
<td>jenx</td>
<td>56</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>jetcd</td>
<td>55</td>
<td>351</td>
<td>495</td>
<td>0</td>
<td>846</td>
</tr>
<tr>
<td>jfactory</td>
<td>60</td>
<td>656</td>
<td>303</td>
<td>0</td>
<td>961</td>
</tr>
<tr>
<td>jgit</td>
<td>941</td>
<td>182971</td>
<td>565041</td>
<td>28332</td>
<td>776444</td>
</tr>
<tr>
<td>jgraphq</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>jgraphx</td>
<td>138</td>
<td>19491</td>
<td>36145</td>
<td>2479</td>
<td>58115</td>
</tr>
<tr>
<td>jHippcup</td>
<td>50</td>
<td>153</td>
<td>0</td>
<td>0</td>
<td>153</td>
</tr>
<tr>
<td>jhdtree</td>
<td>51</td>
<td>5605</td>
<td>19355</td>
<td>6102</td>
<td>36012</td>
</tr>
<tr>
<td>jilion</td>
<td>855</td>
<td>32335</td>
<td>223486</td>
<td>18737</td>
<td>274558</td>
</tr>
<tr>
<td>jimsfs</td>
<td>174</td>
<td>726</td>
<td>643</td>
<td>1623</td>
<td>2992</td>
</tr>
<tr>
<td>jjwt</td>
<td>23</td>
<td>46</td>
<td>70</td>
<td>23</td>
<td>139</td>
</tr>
<tr>
<td>jns</td>
<td>98</td>
<td>23755</td>
<td>171</td>
<td>254</td>
<td>2600</td>
</tr>
<tr>
<td>jprof-fl</td>
<td>362</td>
<td>12238</td>
<td>49926</td>
<td>72402</td>
<td>89544</td>
</tr>
<tr>
<td>joda-time</td>
<td>4</td>
<td>252</td>
<td>1752</td>
<td>116</td>
<td>2120</td>
</tr>
<tr>
<td>jpl</td>
<td>69</td>
<td>96</td>
<td>216</td>
<td>156</td>
<td>462</td>
</tr>
<tr>
<td>jruby</td>
<td>3694</td>
<td>1683792</td>
<td>0</td>
<td>0</td>
<td>1663752</td>
</tr>
<tr>
<td>jSAX</td>
<td>1382</td>
<td>1697912</td>
<td>312244</td>
<td>10676</td>
<td>588720</td>
</tr>
<tr>
<td>jenx</td>
<td>294</td>
<td>4913</td>
<td>8379</td>
<td>1099</td>
<td>14491</td>
</tr>
<tr>
<td>jenx-java</td>
<td>217</td>
<td>5486</td>
<td>982</td>
<td>624</td>
<td>7092</td>
</tr>
<tr>
<td>jenxPath</td>
<td>83</td>
<td>406</td>
<td>1643</td>
<td>300</td>
<td>2349</td>
</tr>
<tr>
<td>jenxSurfer</td>
<td>75</td>
<td>54</td>
<td>1358</td>
<td>69</td>
<td>1481</td>
</tr>
<tr>
<td>jk</td>
<td>18</td>
<td>6439</td>
<td>55156</td>
<td>15864</td>
<td>76059</td>
</tr>
<tr>
<td>jenx-dataprovider</td>
<td>18</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>jenx4</td>
<td>634</td>
<td>1404</td>
<td>3121</td>
<td>79</td>
<td>4604</td>
</tr>
<tr>
<td>jython</td>
<td>2231</td>
<td>688668</td>
<td>2321053</td>
<td>0</td>
<td>3009721</td>
</tr>
<tr>
<td>kolohoke</td>
<td>20</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>kryo</td>
<td>578</td>
<td>6862</td>
<td>36982</td>
<td>20796</td>
<td>66460</td>
</tr>
<tr>
<td>language-behave</td>
<td>67</td>
<td>38</td>
<td>0</td>
<td>33</td>
<td>71</td>
</tr>
<tr>
<td>languagetools</td>
<td>2690</td>
<td>88803</td>
<td>78291</td>
<td>55887</td>
<td>222481</td>
</tr>
<tr>
<td>lanterna</td>
<td>1491</td>
<td>20851</td>
<td>84881</td>
<td>16366</td>
<td>122098</td>
</tr>
<tr>
<td>LatencyUtils</td>
<td>36</td>
<td>43</td>
<td>7</td>
<td>18</td>
<td>68</td>
</tr>
<tr>
<td>lithgix</td>
<td>500</td>
<td>138743</td>
<td>308356</td>
<td>98797</td>
<td>541196</td>
</tr>
<tr>
<td>lithgix-ri</td>
<td>103</td>
<td>709</td>
<td>260</td>
<td>0</td>
<td>960</td>
</tr>
<tr>
<td>log4j</td>
<td>143</td>
<td>4283</td>
<td>3165</td>
<td>931</td>
<td>8379</td>
</tr>
<tr>
<td>loghook</td>
<td>73</td>
<td>79</td>
<td>974</td>
<td>0</td>
<td>1053</td>
</tr>
<tr>
<td>lombook</td>
<td>869</td>
<td>34502</td>
<td>118551</td>
<td>29764</td>
<td>182817</td>
</tr>
<tr>
<td>mapsforge</td>
<td>144</td>
<td>167</td>
<td>484</td>
<td>0</td>
<td>651</td>
</tr>
<tr>
<td>maps prolific</td>
<td>88</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MariaDB4j</td>
<td>53</td>
<td>184</td>
<td>0</td>
<td>0</td>
<td>184</td>
</tr>
<tr>
<td>maven-wrapper</td>
<td>26</td>
<td>78</td>
<td>0</td>
<td>0</td>
<td>78</td>
</tr>
<tr>
<td>micro-server</td>
<td>101</td>
<td>254</td>
<td>107</td>
<td>39</td>
<td>400</td>
</tr>
<tr>
<td>minio-java</td>
<td>199</td>
<td>13077</td>
<td>8807</td>
<td>1637</td>
<td>23521</td>
</tr>
<tr>
<td>minio</td>
<td>361</td>
<td>8142</td>
<td>643</td>
<td>368</td>
<td>9153</td>
</tr>
<tr>
<td>mockito</td>
<td>626</td>
<td>5123</td>
<td>3126</td>
<td>1088</td>
<td>9337</td>
</tr>
<tr>
<td>mooc</td>
<td>1522</td>
<td>2767</td>
<td>2597</td>
<td>195</td>
<td>5559</td>
</tr>
<tr>
<td>modelmapper</td>
<td>213</td>
<td>907</td>
<td>286</td>
<td>161</td>
<td>1354</td>
</tr>
<tr>
<td>modernizer-maven-plugin</td>
<td>15</td>
<td>15</td>
<td>5</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>moshi</td>
<td>213</td>
<td>992</td>
<td>578</td>
<td>122</td>
<td>1792</td>
</tr>
<tr>
<td>MutabilityDetector</td>
<td>181</td>
<td>422</td>
<td>4861</td>
<td>0</td>
<td>5283</td>
</tr>
<tr>
<td>nakadi</td>
<td>15</td>
<td>390</td>
<td>195</td>
<td>15</td>
<td>600</td>
</tr>
<tr>
<td>nanohttpd</td>
<td>184</td>
<td>734</td>
<td>0</td>
<td>0</td>
<td>734</td>
</tr>
<tr>
<td>nblucax</td>
<td>19</td>
<td>171</td>
<td>1097</td>
<td>0</td>
<td>1178</td>
</tr>
<tr>
<td>nifty-gui</td>
<td>213</td>
<td>8876</td>
<td>50307</td>
<td>1205</td>
<td>61388</td>
</tr>
<tr>
<td>Repository Name</td>
<td>Analyzierte Commits</td>
<td>PMD Smells</td>
<td>CPD Smells</td>
<td>IClones Smells</td>
<td>Smells Gesamt</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------</td>
<td>------------</td>
<td>------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>NullAway</td>
<td>160</td>
<td>666</td>
<td>10109</td>
<td>140</td>
<td>10535</td>
</tr>
<tr>
<td>omfquick</td>
<td>1082</td>
<td>18693</td>
<td>22737</td>
<td>148</td>
<td>42576</td>
</tr>
<tr>
<td>Orientre</td>
<td>710</td>
<td>15591</td>
<td>243953</td>
<td>2512</td>
<td>264736</td>
</tr>
<tr>
<td>orika</td>
<td>172</td>
<td>1978</td>
<td>1359</td>
<td>907</td>
<td>4244</td>
</tr>
<tr>
<td>osyx</td>
<td>57</td>
<td>68</td>
<td>0</td>
<td>57</td>
<td>125</td>
</tr>
<tr>
<td>owner</td>
<td>86</td>
<td>129</td>
<td>0</td>
<td>0</td>
<td>129</td>
</tr>
<tr>
<td>parsi</td>
<td>235</td>
<td>4043</td>
<td>1772</td>
<td>7</td>
<td>5852</td>
</tr>
<tr>
<td>picocli</td>
<td>973</td>
<td>18487</td>
<td>19509</td>
<td>1838</td>
<td>37834</td>
</tr>
<tr>
<td>pinpoint</td>
<td>513</td>
<td>8650</td>
<td>15992</td>
<td>7628</td>
<td>32290</td>
</tr>
<tr>
<td>pm4d</td>
<td>227</td>
<td>7900</td>
<td>7429</td>
<td>1921</td>
<td>17250</td>
</tr>
<tr>
<td>polygiet-maven</td>
<td>16</td>
<td>295</td>
<td>104</td>
<td>24</td>
<td>423</td>
</tr>
<tr>
<td>mocka</td>
<td>2</td>
<td>234</td>
<td>322</td>
<td>282</td>
<td>836</td>
</tr>
<tr>
<td>powermock</td>
<td>41</td>
<td>135</td>
<td>23</td>
<td>0</td>
<td>158</td>
</tr>
<tr>
<td>preto</td>
<td>8739</td>
<td>1471534</td>
<td>429663</td>
<td>885988</td>
<td>6652135</td>
</tr>
<tr>
<td>primfaces</td>
<td>1138</td>
<td>215289</td>
<td>2470185</td>
<td>214316</td>
<td>2899790</td>
</tr>
<tr>
<td>protobuf</td>
<td>68</td>
<td>3514</td>
<td>3063</td>
<td>8029</td>
<td>26406</td>
</tr>
<tr>
<td>protostack</td>
<td>52</td>
<td>44</td>
<td>12</td>
<td>58</td>
<td>66</td>
</tr>
<tr>
<td>quartu</td>
<td>150</td>
<td>7997</td>
<td>22155</td>
<td>6518</td>
<td>36688</td>
</tr>
<tr>
<td>raml-tester</td>
<td>184</td>
<td>1104</td>
<td>396</td>
<td>157</td>
<td>1654</td>
</tr>
<tr>
<td>randomizedtesting</td>
<td>143</td>
<td>1330</td>
<td>407</td>
<td>182</td>
<td>1919</td>
</tr>
<tr>
<td>reactive-streams-jym</td>
<td>84</td>
<td>152</td>
<td>4132</td>
<td>307</td>
<td>4721</td>
</tr>
<tr>
<td>restful-java</td>
<td>513</td>
<td>11106</td>
<td>20215</td>
<td>6312</td>
<td>37633</td>
</tr>
<tr>
<td>redisson</td>
<td>1616</td>
<td>123316</td>
<td>211577</td>
<td>1166975</td>
<td></td>
</tr>
<tr>
<td>registry</td>
<td>314</td>
<td>1935</td>
<td>17122</td>
<td>1390</td>
<td>20447</td>
</tr>
<tr>
<td>resilience4j</td>
<td>43</td>
<td>0</td>
<td>55</td>
<td>8</td>
<td>63</td>
</tr>
<tr>
<td>rest-assured</td>
<td>79</td>
<td>1249</td>
<td>5569</td>
<td>6749</td>
<td>13567</td>
</tr>
<tr>
<td>restful-i</td>
<td>181</td>
<td>2519</td>
<td>1087</td>
<td>181</td>
<td>4087</td>
</tr>
<tr>
<td>RestExpress</td>
<td>26</td>
<td>442</td>
<td>26</td>
<td>0</td>
<td>468</td>
</tr>
<tr>
<td>ribbon</td>
<td>157</td>
<td>969</td>
<td>3872</td>
<td>173</td>
<td>5014</td>
</tr>
<tr>
<td>richfaces</td>
<td>94</td>
<td>1504</td>
<td>5184</td>
<td>188</td>
<td>6876</td>
</tr>
<tr>
<td>riptide</td>
<td>5</td>
<td>510</td>
<td>10</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>rocketmq</td>
<td>107</td>
<td>9180</td>
<td>33293</td>
<td>223</td>
<td>42096</td>
</tr>
<tr>
<td>solFlo</td>
<td>67</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>service-proxy</td>
<td>50</td>
<td>6153</td>
<td>7470</td>
<td>971</td>
<td>14594</td>
</tr>
<tr>
<td>simple-binary-encoding</td>
<td>694</td>
<td>21945</td>
<td>132036</td>
<td>29743</td>
<td>183724</td>
</tr>
<tr>
<td>squeak</td>
<td>55</td>
<td>396</td>
<td>231</td>
<td>0</td>
<td>486</td>
</tr>
<tr>
<td>stackoverflow4j</td>
<td>262</td>
<td>679</td>
<td>2314</td>
<td>0</td>
<td>2993</td>
</tr>
<tr>
<td>Slackstream-Mod-Manager</td>
<td>235</td>
<td>7153</td>
<td>4539</td>
<td>2518</td>
<td>14210</td>
</tr>
<tr>
<td>Snack</td>
<td>16</td>
<td>432</td>
<td>96</td>
<td>16</td>
<td>544</td>
</tr>
<tr>
<td>sonarqube</td>
<td>3701</td>
<td>40640</td>
<td>342867</td>
<td>4127</td>
<td>387634</td>
</tr>
<tr>
<td>spring-pom</td>
<td>142</td>
<td>72765</td>
<td>231025</td>
<td>62869</td>
<td>366159</td>
</tr>
<tr>
<td>ssit-indflow</td>
<td>968</td>
<td>26590</td>
<td>19194</td>
<td>5281</td>
<td>51065</td>
</tr>
<tr>
<td>SourcererCC</td>
<td>17</td>
<td>561</td>
<td>289</td>
<td>136</td>
<td>986</td>
</tr>
<tr>
<td>spatial4j</td>
<td>65</td>
<td>399</td>
<td>216</td>
<td>253</td>
<td>888</td>
</tr>
<tr>
<td>spaceon</td>
<td>1785</td>
<td>85581</td>
<td>1592839</td>
<td>16313</td>
<td>1694534</td>
</tr>
<tr>
<td>spark-boot</td>
<td>386</td>
<td>12699</td>
<td>62794</td>
<td>76051</td>
<td></td>
</tr>
<tr>
<td>stagnanton</td>
<td>470</td>
<td>1958</td>
<td>2559</td>
<td>0</td>
<td>4557</td>
</tr>
<tr>
<td>streamex</td>
<td>241</td>
<td>4675</td>
<td>12484</td>
<td>6314</td>
<td>23473</td>
</tr>
<tr>
<td>swaktit</td>
<td>474</td>
<td>210925</td>
<td>374602</td>
<td>134172</td>
<td>710690</td>
</tr>
<tr>
<td>tablesaw</td>
<td>803</td>
<td>12826</td>
<td>130042</td>
<td>19948</td>
<td>162816</td>
</tr>
<tr>
<td>tap</td>
<td>67</td>
<td>0</td>
<td>37</td>
<td>1</td>
<td>37</td>
</tr>
<tr>
<td>tastes4j</td>
<td>145</td>
<td>1149</td>
<td>1350</td>
<td>1606</td>
<td>4105</td>
</tr>
<tr>
<td>testcontainers-java</td>
<td>408</td>
<td>857</td>
<td>176</td>
<td>83</td>
<td>1116</td>
</tr>
<tr>
<td>thumbinator</td>
<td>22</td>
<td>248</td>
<td>121</td>
<td>22</td>
<td>391</td>
</tr>
<tr>
<td>Time4J</td>
<td>258</td>
<td>39132</td>
<td>150844</td>
<td>33235</td>
<td>223211</td>
</tr>
<tr>
<td>tindel</td>
<td>517</td>
<td>5271</td>
<td>115000</td>
<td>5980</td>
<td>128151</td>
</tr>
<tr>
<td>trove</td>
<td>1</td>
<td>4</td>
<td>17</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>truth</td>
<td>719</td>
<td>2889</td>
<td>5364</td>
<td>3011</td>
<td>11264</td>
</tr>
<tr>
<td>underscore-java</td>
<td>461</td>
<td>1929</td>
<td>12314</td>
<td>4330</td>
<td>18573</td>
</tr>
<tr>
<td>university-parsers</td>
<td>576</td>
<td>11910</td>
<td>14136</td>
<td>352</td>
<td>26598</td>
</tr>
<tr>
<td>war</td>
<td>63</td>
<td>659</td>
<td>0</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>xing</td>
<td>207</td>
<td>12308</td>
<td>10051</td>
<td>10018</td>
<td>32377</td>
</tr>
<tr>
<td>Gesamt</td>
<td>131918</td>
<td>17351979</td>
<td>47192107</td>
<td>7097373</td>
<td>71641513</td>
</tr>
</tbody>
</table>

Tabelle 8: Die gesamten Analyseergebnisse.

Die rot unterlegten Felder bedeuten, dass die jeweilige Analyse auf dem jeweiligen Repository nicht ausgeführt werden konnte. Alle Zahlen zu gefundenen Smells beziehen sich auf die Menge gefundenen Smells über alle analysierten Commits.
### Tabelle B

<table>
<thead>
<tr>
<th>group1</th>
<th>group2</th>
<th>meandiff</th>
<th>p-adj</th>
<th>lower</th>
<th>upper</th>
<th>reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataClass</td>
<td>ExcessiveClassLength</td>
<td>19.839</td>
<td>0.001</td>
<td>19.2667</td>
<td>20.4113</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveMethodLength</td>
<td>1.6595</td>
<td>0.001</td>
<td>1.205</td>
<td>2.114</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveParameterList</td>
<td>-2.2326</td>
<td>0.001</td>
<td>-2.8085</td>
<td>-1.6566</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>GodClass</td>
<td>12.8506</td>
<td>0.001</td>
<td>12.3909</td>
<td>13.3102</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>SingularField</td>
<td>-2.6077</td>
<td>0.001</td>
<td>-3.1411</td>
<td>-2.0744</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>ExcessiveMethodLength</td>
<td>-2.4338</td>
<td>0.001</td>
<td>-2.8185</td>
<td>-2.0491</td>
<td>True</td>
</tr>
<tr>
<td>DataClass</td>
<td>idlones-clone</td>
<td>-2.1487</td>
<td>0.001</td>
<td>-2.552</td>
<td>-1.7434</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveMethodLength</td>
<td>-18.1795</td>
<td>0.001</td>
<td>-18.67</td>
<td>-17.689</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>ExcessiveParameterList</td>
<td>-22.0716</td>
<td>0.001</td>
<td>-22.6763</td>
<td>-21.4668</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>GodClass</td>
<td>-6.9884</td>
<td>0.001</td>
<td>-7.4837</td>
<td>-6.4931</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>SingularField</td>
<td>-22.4467</td>
<td>0.001</td>
<td>-23.0111</td>
<td>-21.8824</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>UnusedPrivateField</td>
<td>-22.4473</td>
<td>0.001</td>
<td>-22.9794</td>
<td>-21.9153</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveClassLength</td>
<td>idlones-clone</td>
<td>-21.9877</td>
<td>0.001</td>
<td>-22.4312</td>
<td>-21.5442</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>ExcessiveParameterList</td>
<td>-3.8921</td>
<td>0.001</td>
<td>-4.3868</td>
<td>-3.3973</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>GodClass</td>
<td>11.1911</td>
<td>0.001</td>
<td>10.8385</td>
<td>11.5437</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>SingularField</td>
<td>-1.2672</td>
<td>0.001</td>
<td>-4.7116</td>
<td>-3.8238</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>UnusedPrivateField</td>
<td>-1.2679</td>
<td>0.001</td>
<td>-4.6704</td>
<td>-3.8653</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>idlones-clone</td>
<td>-1.0933</td>
<td>0.001</td>
<td>-4.3403</td>
<td>-3.8463</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveMethodLength</td>
<td>ExcessiveParameterList</td>
<td>-3.8082</td>
<td>0.001</td>
<td>-4.0833</td>
<td>-3.5331</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>GodClass</td>
<td>15.0831</td>
<td>0.001</td>
<td>14.5837</td>
<td>15.5826</td>
<td>True</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>SingularField</td>
<td>-0.3752</td>
<td>0.5075</td>
<td>-0.9432</td>
<td>0.1929</td>
<td>False</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>UnusedPrivateField</td>
<td>-0.3758</td>
<td>0.4253</td>
<td>-0.9117</td>
<td>0.1602</td>
<td>False</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>idlones-clone</td>
<td>-0.2012</td>
<td>0.8729</td>
<td>-0.6327</td>
<td>0.2303</td>
<td>False</td>
</tr>
<tr>
<td>ExcessiveParameterList</td>
<td>GodClass</td>
<td>0.0839</td>
<td>0.9</td>
<td>-0.3643</td>
<td>0.5321</td>
<td>False</td>
</tr>
<tr>
<td>GodClass</td>
<td>SingularField</td>
<td>-15.4583</td>
<td>0.001</td>
<td>-15.908</td>
<td>-15.0086</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>UnusedPrivateField</td>
<td>-15.4589</td>
<td>0.001</td>
<td>-15.8673</td>
<td>-15.0505</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>idlones-clone</td>
<td>-15.2844</td>
<td>0.001</td>
<td>-15.5408</td>
<td>-15.028</td>
<td>True</td>
</tr>
<tr>
<td>GodClass</td>
<td>idlones-clone</td>
<td>-14.9993</td>
<td>0.001</td>
<td>-15.2828</td>
<td>-14.7157</td>
<td>True</td>
</tr>
<tr>
<td>SingularField</td>
<td>UnusedPrivateField</td>
<td>-0.0006</td>
<td>0.9</td>
<td>-0.4905</td>
<td>0.4893</td>
<td>False</td>
</tr>
<tr>
<td>SingularField</td>
<td>idlones-clone</td>
<td>0.1739</td>
<td>0.8724</td>
<td>-0.1988</td>
<td>0.5466</td>
<td>False</td>
</tr>
<tr>
<td>SingularField</td>
<td>cpd-clone</td>
<td>0.4591</td>
<td>0.0086</td>
<td>0.0672</td>
<td>0.851</td>
<td>True</td>
</tr>
<tr>
<td>UnusedPrivateField</td>
<td>idlones-clone</td>
<td>0.4597</td>
<td>0.001</td>
<td>0.1159</td>
<td>0.8034</td>
<td>True</td>
</tr>
<tr>
<td>cpd-clone</td>
<td>idlones-clone</td>
<td>0.2851</td>
<td>0.001</td>
<td>0.1543</td>
<td>0.416</td>
<td>True</td>
</tr>
</tbody>
</table>


Die rot unterlegten Zeilen bedeuten, dass die Nullhypothese für das jeweilige Paar nicht abgelehnt werden konnte.
**Boxplot A**

Abbildung 6: Analysierte Commits mit Ausreißern.

**Boxplot B**

Bei diesem Boxplot ist zu beachten, dass die Y-Achse mit $1e6$ beschriftet ist. Die Zahlen an der Y-Achse müssen also in Millionen gelesen werden.

Abbildung 8: Ergebnisse der Auswertung der Änderungen als Boxplot mit Ausreißern.
Eidesstattliche Erklärung zur Bachelorarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Unterschrift: 

Ort, Datum:
