Match algorithm description

Bruno Crotman

18/04/2020

1 Introduction

This document is part of a bigger project. The project is a research about software degradation caused by careless developers' behavior and strategies to deal with it. The strategies to deal with the problem will possibly be inspired by concepts from game theory. At this moment, we think that software degradation can be measured by the number and the types of kludges made by software developers in the code. So, one of the goals of this project at this moment is to study how software projects evolve in terms of number of kludges and kinds of kludges.

Right now we are trying to identify kludges looking at alerts generated by PMD source code analyzer. This tool receives a programming code as input and generates a list of bad programming practices contained in the code.

In order to evaluate how the number of alerts is evolving along the history of a software project, we must be able to analyze two different versions of a code (an old version and a new version) and cateborize each alert as **new**, **fixed** or **not fixed**.

- each PMD alert generated for the old version is either **not fixed** or **fixed** in the new version. When an alert is **not fixed**, it means that it remains in the new version of the code. When an alert is **fixed**, it means that it does not exist anymore in the new version.
- each PMD alert generated for the new version is **not fixed** or **new**. When an alert is **not fixed** it means that the same alert was identified in the old version. When an alert is **new**, it means that the same alert cannot be identified in the old version.

The alerts identified as *not fixed* are the equivalent in both new and old versions. And the intersection between the *fixed* alerts, the *new* alerts and the *not fixed* alerts is empty.

In order to decide if an alert is *not fixed*, *fixed* or *new*, we have to identify if an alert in the old version is equivalent to an alert in the new version.

I describe here two algorithms. The first one, described in Section 2, is a naive algorithm based on matches by lines of code. The second is a more sophisticated algorithm, based on matches by blocks of code.

2 Matches by line of code

In this first algorithm, I match the lines of code of the old version with the lines of code of the new version using information from the output of git's diff command. When an alert with the same features occurs in both matched lines, this alert is declared *not fixed*. The alerts that occur in a not matched line of the old version are declared *fixed* and the alerts located in a not matched line of the new version are declared *new*

These are the steps of the algorithm:

1. Generate a list of alerts from each version (old and new) using PMD Alert

2. Generate the git diff between the two versions
3. Using information from git diff, create a map between the lines
4. Categorize the alerts in *new*, *not fixed* or *fixed*

2.1 Generate a list of alerts for each version

The two codes presented in this Section, named "new version" and "old version", are used in Sections 2.2, 2.3 and 2.4 to describe the algorithm.

The old version, with the alerts generated by PMI, is shown below.

```
/*  1-                                  *//*
/*  2-                                  */ * Copyright (C) 2007 Yusuke Yamamoto
/*  3-                                  */ * Copyright (C) 2011 Twitter, Inc.
/*  4-                                  *//* ...  */
/* 20-                                  *//* ...  */
/* 21-                                  */import twitter4j.conf.Configuration;
/* 22-                                  */
/* 23-                                  */import java.util.*;
/* 24-UnusedImports                     */import java.util.concurrent.ConcurrentHashMap;
/* 25-                                  */
/* 26-                                  */
/* 27-                                  *//**
/* 28-                                   *//* ...  */
/* 32-                                  *//* ...  */
/* 33-                                  */ */
/* 34-                                  */class TwitterImpl extends TwitterBaseImpl implements Twitter {
/* 35-                                  */    private static final long serialVersionUID = 9170943084096085770L;
/* 36-UnusedPrivateField                */    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/* 37-                                  */
/* 38-                                  */    /*package*/
/* 39-                                  */    TwitterImpl(Configuration conf, Authorization auth) {
/* 40-                                  *//* ...  */
/* 74-                                  *//* ...  */
/* 75-                                  */            if (conf.isTweetModeExtended()) {
/* 76-                                  */                params.add(new HttpParameter("tweet_mode", "extended"));
/* 77-                                  */            }
/* 78-OptimizableToArrayCall            */            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size()]);
/* 79-                                  */
/* 80-                                  */            // implicitParamsMap.containsKey() is evaluated in the above if clause.
/* 81-                                  */            // thus implicitParamsStrMap needs to be initialized first
/* 82-                                  *//* ...  */
/* 88-                                  *//* ...  */
/* 89-                                  */
/* 90-                                  */
/* 91-                                  */    @Override
/* 92-FormalParameterNamingConventions  */    public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
/* 93-FormalParameterNamingConventions(2)*/                                                 Boolean sleep_timeEnabled, String start_sleepTime,
/* 94-FormalParameterNamingConventions(2)*/                                                 String end_sleepTime, String time_zone, String lang)
/* 95-                                  */            throws TwitterException {
/* 96-                                  */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/* 97-                                  */        if (trend_locationWoeid != null) {
/* 98-                                  *//* ...  */
```

```
/*112-                                 *//* ...  */
/*113-                                 */            profile.add(new HttpParameter("lang", lang));
/*114-                                 */        }
/*115-                                 */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/settings.json"
/*116-OptimizableToArrayCall           */                , profile.toArray(new HttpParameter[profile.size()])));
/*117-                                 */
/*118-                                 */    }
/*119-                                 */
/*120-                                 */}
```

Table 1 lists the alerts found in the old version.

Table 1: Alerts in the old version

id	beginline	ruleset	rule	package	class	method	variable
1	24	Best Practices	UnusedImports	twitter4j	TwitterImpl	No method	No variable
2	36	Best Practices	UnusedPrivateField	twitter4j	TwitterImpl	No method	logger
3	78	Performance	OptimizableToArrayCall	twitter4j	TwitterImpl	TwitterImpl	No variable
4	92	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	trend_locationWoeid
5	93	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	sleep_timeEnabled
6	93	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	start_sleepTime
7	94	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	end_sleepTime
8	94	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	time_zone
9	116	Performance	OptimizableToArrayCall	twitter4j	TwitterImpl	updateAccountSettings	No variable

The new version has the following changes in relation to the old version:

- line 24 of the old version "import java.util.concurrent.ConcurrentHashMap", that was a "Unused Import", was removed. So the alert related to this line must be declared *fixed*.
- line 92 of the old version (91 of the new version) was fixed by changing the name of the parameter. This must be classified as another *fixed* alert.
- line 119 was included in the new version, with an unused private field. This must be categorized as a *new* alert

```
/*  1-                                  *//*
/*  2-                                  */ * Copyright (C) 2007 Yusuke Yamamoto
/*  3-                                  */ * Copyright (C) 2011 Twitter, Inc.
/*  4-                                  *//* ...  */
/* 31-                                  *//* ...  */
/* 32-                                  */ */
/* 33-                                  */class TwitterImpl extends TwitterBaseImpl implements Twitter {
/* 34-                                  */    private static final long serialVersionUID = 9170943084096085770L;
/* 35-UnusedPrivateField                */    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/* 36-                                  */
/* 37-                                  */    /*package*/
/* 38-                                  */    TwitterImpl(Configuration conf, Authorization auth) {
/* 39-                                  *//* ...  */
/* 73-                                  *//* ...  */
/* 74-                                  */            if (conf.isTweetModeExtended()) {
/* 75-                                  */                params.add(new HttpParameter("tweet_mode", "extended"));
/* 76-                                  */            }
/* 77-OptimizableToArrayCall            */            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size()]);
/* 78-                                  */
/* 79-                                  */            // implicitParamsMap.containsKey() is evaluated in the above if clause.
/* 80-                                  */            // thus implicitParamsStrMap needs to be initialized first
/* 81-                                  *//* ...  */
/* 88-                                  *//* ...  */
/* 89-                                  */
/* 90-                                  */    @Override
/* 91-                                  */    public AccountSettings updateAccountSettings(Integer trendlocationWoeid,
/* 92-FormalParameterNamingConventions(2)*/                                                 Boolean sleep_timeEnabled, String start_sleepTime,
/* 93-FormalParameterNamingConventions(2)*/                                                 String end_sleepTime, String time_zone, String lang)
/* 94-                                  */            throws TwitterException {
/* 95-                                  */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/* 96-                                  */        if (trendlocationWoeid != null) {
/* 97-                                  *//* ...  */
/*111-                                  *//* ...  */
/*112-                                  */            profile.add(new HttpParameter("lang", lang));
/*113-                                  */        }
/*114-                                  */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/settings.json"
/*115-OptimizableToArrayCall            */                , profile.toArray(new HttpParameter[profile.size()])));
/*116-                                  */
/*117-                                  */    }
/*118-                                  */
/*119-                                  */
/*120-                                  */    @Override
```

```
/*121-                                    */    public AccountSettings updateAccountSettings2(Integer trendlocationWoeid,
/*122-FormalParameterNamingConventions(2)*/                                                  Boolean sleep_timeEnabled, String start_sleepTime,
/*123-FormalParameterNamingConventions(2)*/                                                  String end_sleepTime, String time_zone, String lang)
/*124-                                    */            throws TwitterException {
/*125-                                    */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/*126-                                    */        if (trendlocationWoeid != null) {
/*127-                                    *//* ...  */
/*141-                                    *//* ...  */
/*142-                                    */            profile.add(new HttpParameter("lang", lang));
/*143-                                    */        }
/*144-                                    */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/settings.json"
/*145-OptimizableToArrayCall              */                , profile.toArray(new HttpParameter[profile.size()])));
/*146-                                    */
/*147-                                    */    }
/*148-                                    */
/*149-                                    */
/*150-UnusedPrivateField                  */    private int not_used = 0;
/*151-                                    */
/*152-                                    */}
```

Table 2 lists the alerts found in the new version.

Table 2: Alerts in the new version

id	beginline	ruleset	rule	package	class	method	variable
1	35	Best Practices	UnusedPrivateField	twitter4j	TwitterImpl	No method	logger
2	77	Performance	OptimizableToArrayCall	twitter4j	TwitterImpl	TwitterImpl	No variable
3	92	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	sleep_timeEnabled
4	92	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	start_sleepTime
5	93	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	end_sleepTime
6	93	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings	time_zone
7	115	Performance	OptimizableToArrayCall	twitter4j	TwitterImpl	updateAccountSettings	No variable
8	122	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings2	sleep_timeEnabled
9	122	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings2	start_sleepTime
10	123	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings2	end_sleepTime
11	123	Code Style	FormalParameterNamingConventions	twitter4j	TwitterImpl	updateAccountSettings2	time_zone
12	145	Performance	OptimizableToArrayCall	twitter4j	TwitterImpl	updateAccountSettings2	No variable
13	150	Best Practices	UnusedPrivateField	twitter4j	TwitterImpl	No method	not_used

2.2 Generate the git diff between the two versions

The command git diff is executed between the two versions with the option –patience.

The result of the git diff operation between this two versions is shown below:

```
36  4   j/match_algorithm_description/{old => new}/code.java

diff --git a/j/match_algorithm_description/old/code.java b/j/match_algorithm_description/new/code.java
index 33a26b4..375268c 100644
--- a/j/match_algorithm_description/old/code.java
+++ b/j/match_algorithm_description/new/code.java
@@ -24 +23,0 @@ import java.util.*;
-import java.util.concurrent.ConcurrentHashMap;
@@ -92 +91 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
-    public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
+    public AccountSettings updateAccountSettings(Integer trendlocationWoeid,
@@ -97,2 +96,2 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
-        if (trend_locationWoeid != null) {
-            profile.add(new HttpParameter("trend_location_woeid", trend_locationWoeid));
+        if (trendlocationWoeid != null) {
+            profile.add(new HttpParameter("trend_location_woeid", trendlocationWoeid));
@@ -119,0 +119,33 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
+
+    @Override
+    public AccountSettings updateAccountSettings2(Integer trendlocationWoeid,
+                                                  Boolean sleep_timeEnabled, String start_sleepTime,
+                                                  String end_sleepTime, String time_zone, String lang)
+            throws TwitterException {
+        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
+        if (trendlocationWoeid != null) {
+            profile.add(new HttpParameter("trend_location_woeid", trendlocationWoeid));
+        }
+        if (sleep_timeEnabled != null) {
+            profile.add(new HttpParameter("sleep_time_enabled", sleep_timeEnabled.toString()));
+        }
+        if (start_sleepTime != null) {
+            profile.add(new HttpParameter("start_sleep_time", start_sleepTime));
+        }
+        if (end_sleepTime != null) {
+            profile.add(new HttpParameter("end_sleep_time", end_sleepTime));
+        }
```

```
+        if (time_zone != null) {
+            profile.add(new HttpParameter("time_zone", time_zone));
+        }
+        if (lang != null) {
+            profile.add(new HttpParameter("lang", lang));
+        }
+        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/settings.json"
+                , profile.toArray(new HttpParameter[profile.size()])));
+
+    }
+
+
+    private int not_used = 0;
+
```

2.3 Using information from git diff, create a map between the lines

Using this information from git's diff, it's possible to create a map between the lines of the old version and the new version.

For each difference stated in the git diff output (the sections of the diff file starting with "@@"), there is an indication of the number of lines removed from the old version and the number of lines added to the new version. The line in which the lines are removed from the old version and the line at which the lines are added is indicated, too. Following this information it´s possible to create a map between the lines of the old version and the equivalent lines in the new version.

For the new and old versions presented in Section 2.1, the map is shown in Table 3

Table 3: Map between lines of the old version and lines of the new version

old	new
1	1
2	2
...	...
...	...
22	22
23	23
24	NA
25	24
26	25
...	...
...	...
90	89
91	90
NA	91
92	NA
93	92
94	93
95	94
96	95
NA	96
NA	97
97	NA
98	NA
99	98
100	99

Table 3: Map between lines of the old version and lines of the new version *(continued)*

old	new
...	...
...	...
118	117
119	118
NA	119
NA	120
NA	121
NA	122
NA	123
NA	124
NA	125
NA	126
NA	127
NA	128
NA	129
NA	130
NA	131
NA	132
NA	133
NA	134
NA	135
NA	136
NA	137
NA	138
NA	139
NA	140
NA	141
NA	142
NA	143
NA	144
NA	145
NA	146
NA	147
NA	148
NA	149
NA	150
NA	151
120	152

Now we can connect the to versions and plugin the alerts

```
/*  1-                     *//*
/*  2-                     */ * Copyright (C) 2007 Yusuke Yamamoto
/*  3-                     */ * Copyright (C) 2011 Twitter, Inc.
/*  4-                     *//* ...  */
/* 20-                     *//* ...  */
/* 21-                     */import twitter4j.conf.Configuration;
/* 22-                     */
/* 23-                     */import java.util.*;
/* 24-UnusedImports        */import java.util.concurrent.ConcurrentHashMap;
/* 25-                     */
/* 26-                     */
/* 27-                     *//**
/* 28-                      *//* ...  */
/* 32-                     *//* ...  */
/* 33-                     */ */
/* 34-                     */class TwitterImpl extends TwitterBaseImpl implements Twitter {
/* 35-                     */    private static final long serialVersionUID = 9170943084096085770L;
/* 36-UnusedPrivateField   */    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/* 37-                     */
/* 38-                     */    /*package*/
/* 39-                     */    TwitterImpl(Configuration conf, Authorization auth) {
/* 40-                     *//* ...  */
/* 74-                     *//* ...  */
/* 75-                     */            if (conf.isTweetModeExtended()) {
/* 76-                     */                params.add(new HttpParameter("tweet_mode", "extended"));
/* 77-                     */            }
/* 78-OptimizableToArrayCa */            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size(
/* 79-                     */
/* 80-                     */            // implicitParamsMap.containsKey() is evaluated in the above if clause.
/* 81-                     */            // thus implicitParamsStrMap needs to be initialized first
/* 82-                     *//* ...  */
/* 89-                     *//* ...  */
/* 90-                     */
/* 91-                     */    @Override
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/* 92-FormalParameterNamin */    public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
/* 93-FormalParameterNamin(2)*/                                              Boolean sleep_timeEnabled, String start_s
/* 94-FormalParameterNamin(2)*/                                              String end_sleepTime, String time_zone, S
/* 95-                     */            throws TwitterException {
/* 96-                     */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//* ...  */
/*112-                     *//* ...  */
/*113-                     */            profile.add(new HttpParameter("lang", lang));
/*114-                     */        }
/*115-                     */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/setting/
/*116-OptimizableToArrayCa */                , profile.toArray(new HttpParameter[profile.size()])));
/*117-                     */
/*118-                     */    }
/*119-                     */
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//* ...  */
/*   -                     *//* ...  */
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/*120-                     */}
```

```
/*  1-                     *//*
/*  2-                     */ * Copyright (C) 2007 Yusuke Yamamoto
/*  3-                     */ * Copyright (C) 2011 Twitter, Inc.
/*  4-                     *//* ...  */
/* 20-                     *//* ...  */
/* 21-                     */import twitter4j.conf.Configuration;
/* 22-                     */
/* 23-                     */import java.util.*;
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/* 24-                     */
/* 25-                     */
/* 26-                     *//**
/* 27-                     *//* ...  */
/* 31-                     *//* ...  */
/* 32-                     */ */
/* 33-                     */class TwitterImpl extends TwitterBaseImpl implements Twitter {
/* 34-                     */    private static final long serialVersionUID = 9170943084096085770L;
/* 35-UnusedPrivateField   */    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/* 36-                     */
/* 37-                     */    /*package*/
/* 38-                     */    TwitterImpl(Configuration conf, Authorization auth) {
/* 39-                     *//* ...  */
/* 73-                     *//* ...  */
/* 74-                     */            if (conf.isTweetModeExtended()) {
/* 75-                     */                params.add(new HttpParameter("tweet_mode", "extended"));
/* 76-                     */            }
/* 77-OptimizableToArrayCa */            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size(
/* 78-                     */
/* 79-                     */            // implicitParamsMap.containsKey() is evaluated in the above if clause.
/* 80-                     */            // thus implicitParamsStrMap needs to be initialized first
/* 81-                     *//* ...  */
/* 88-                     *//* ...  */
/* 89-                     */
/* 90-                     */    @Override
/* 91-                     */    public AccountSettings updateAccountSettings(Integer trendlocationWoeid,
/*   -                     *//*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
/* 92-FormalParameterNamin(2)*/                                              Boolean sleep_timeEnabled, String start_s
/* 93-FormalParameterNamin(2)*/                                              String end_sleepTime, String time_zone, S
/* 94-                     */            throws TwitterException {
/* 95-                     */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/* 96-                     */        if (trendlocationWoeid != null) {
/* 97-                     *//* ...  */
/*111-                     *//* ...  */
/*112-                     */            profile.add(new HttpParameter("lang", lang));
/*113-                     */        }
/*114-                     */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/setting
/*115-OptimizableToArrayCa */                , profile.toArray(new HttpParameter[profile.size()])));
/*116-                     */
/*117-                     */    }
/*118-                     */
/*119-                     */
/*120-                     */    @Override
/*121-                     */    public AccountSettings updateAccountSettings2(Integer trendlocationWoeid,
/*122-FormalParameterNamin(2)*/                                              Boolean sleep_timeEnabled, String start_s
/*123-FormalParameterNamin(2)*/                                              String end_sleepTime, String time_zone, S
/*124-                     */            throws TwitterException {
/*125-                     */        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/*126-                     */        if (trendlocationWoeid != null) {
/*127-                     *//* ...  */
/*141-                     *//* ...  */
/*142-                     */            profile.add(new HttpParameter("lang", lang));
/*143-                     */        }
/*144-                     */        return factory.createAccountSettings(post(conf.getRestBaseURL() + "account/setting
/*145-OptimizableToArrayCa */                , profile.toArray(new HttpParameter[profile.size()])));
/*146-                     */
/*147-                     */    }
/*148-                     */
/*149-                     */
/*150-UnusedPrivateField   */    private int not_used = 0;
/*151-                     */
/*152-                     */}
```

2.4 Categorize the alerts

The alert in the old version is classified as **not fixed** if there is an alert in the new version in the corresponding line with the same rule, same method name and same variable name. Otherwise, the alert is categorised as **fixed**

Table **??** shows the classification of the alerts in the old version. The alerts 1 and 4

Table 4: Classifications of the alerts in the old version

id	line	rule	class	method	variable	idnew	linenew	category
1	24	UnusedImports	TwitterImpl	No method	No variable	NA	NA	Fixed
2	36	UnusedPrivateField	TwitterImpl	No method	logger	1	35	Not fixed
3	78	OptimizableToArrayCall	TwitterImpl	TwitterImpl	No variable	2	77	Not fixed
4	92	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	trend_locationWoeid	NA	NA	Fixed
5	93	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	sleep_timeEnabled	3	92	Not fixed
6	93	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	start_sleepTime	4	92	Not fixed
7	94	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	end_sleepTime	5	93	Not fixed
8	94	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	time_zone	6	93	Not fixed
9	116	OptimizableToArrayCall	TwitterImpl	updateAccountSettings	No variable	7	115	Not fixed

Table **??** shows the classification of the alerts in the new version.

Table 5: Classifications of the alerts in the new version

id	line	rule	class	method	variable	idold	lineold	category
1	35	UnusedPrivateField	TwitterImpl	No method	logger	2	36	Not fixed
2	77	OptimizableToArrayCall	TwitterImpl	TwitterImpl	No variable	3	78	Not fixed
3	92	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	sleep_timeEnabled	5	93	Not fixed
4	92	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	start_sleepTime	6	93	Not fixed
5	93	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	end_sleepTime	7	94	Not fixed
6	93	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings	time_zone	8	94	Not fixed
7	115	OptimizableToArrayCall	TwitterImpl	updateAccountSettings	No variable	9	116	Not fixed
8	122	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings2	sleep_timeEnabled	NA	NA	New
9	122	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings2	start_sleepTime	NA	NA	New
10	123	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings2	end_sleepTime	NA	NA	New
11	123	FormalParameterNamingConventions	TwitterImpl	updateAccountSettings2	time_zone	NA	NA	New
12	145	OptimizableToArrayCall	TwitterImpl	updateAccountSettings2	No variable	NA	NA	New
13	150	UnusedPrivateField	TwitterImpl	No method	not_used	NA	NA	New

3 Matches by Block