

MLCQ: Industry-Relevant Code Smell Data Set

Lech Madeyski and Tomasz Lewowski

Wrocław University of Science and Technology
Faculty of Computer Science and Management
Department of Applied Informatics

September 15, 2021



HR EXCELLENCE IN RESEARCH



Wrocław University
of Science and Technology

Acknowledgement

This work has been supported by the National Centre for Research and Development (NCBiR) project POIR.01.01.01-00-0792/16 with the aim to improve the codebeat platform (<http://codebeat.co>) by *code quest sp. z o.o.*
We would like to thank all of the developers involved in the study.



Agenda

1. Code smells
2. Research context
3. Obtaining samples
4. Grading staff
5. Grading samples
6. Industry-relevance



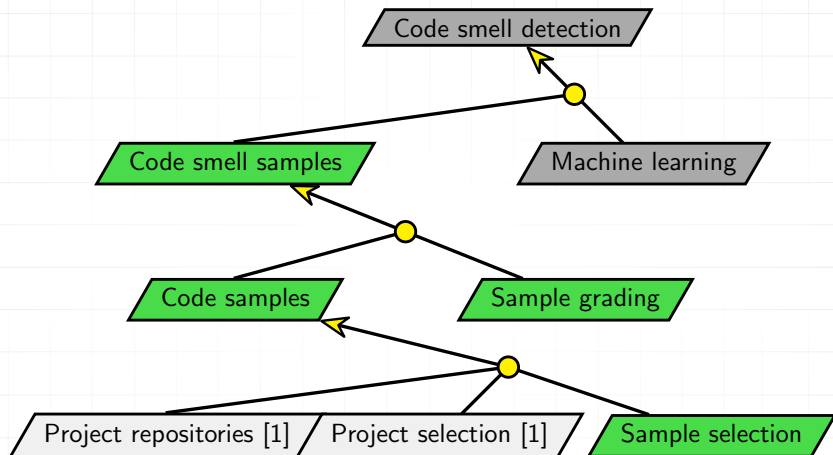
Code smells

Code smells:

- ▶ code fragments,
- ▶ symptoms of troublesome design,
- ▶ not necessarily problems themselves!



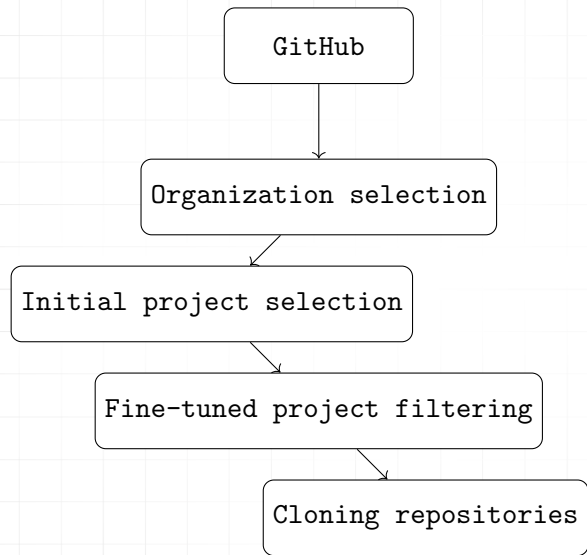
Research context



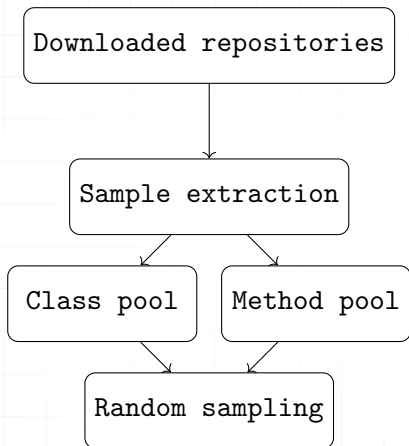
[1] Lewowski T., Madeyski L. (2020) Creating Evolving Project Data Sets in Software Engineering, https://doi.org/10.1007/978-3-030-26574-8_1



Obtaining samples I



Obtaining samples II



Recruiting staff for grading - experience

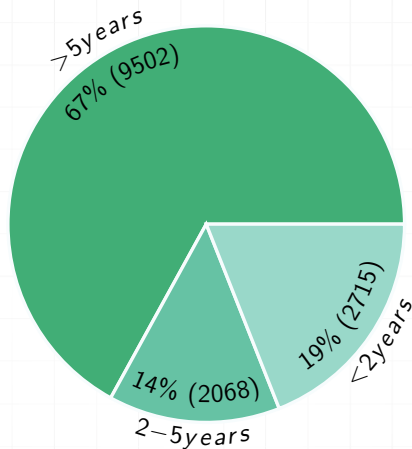


Figure: Experience distribution between members of grading staff



Categorizing grading staff



Grading tool

```
1 public static synchronized IServiceFactory getJava6FactoryFromPlugin( String factoryName )
2 {
3     IServiceFactory isf = getJava6PluginFactoryMap().get( factoryName );
4     if ( isf == null )
5     {
6         String s = "could not find annotation processor " + //$NON-NLS-1$
7             factoryName + " from available factories defined by plugins"; //$NON-NLS-1$
8         AptPlugin.log(new Status(IStatus.WARNING, AptPlugin.PLUGIN_ID, AptPlugin.STATUS_NOTOOLSJAR, s, null));
9     }
10    return isf;
11 }
```

Source code sample

Source on Github

Full source file link

FactoryPluginManager.java:103-113

Your review

You've submitted 3810 reviews so far

long method none minor major critical

feature envy none minor major critical

Reviewing area

Next >

Smells

[More info](#)

long method

A method containing too many lines of code, usually also of high complexity and low cohesion.

feature envy

A method using other classes (fields and methods) more than its own class data. Often occurs along with the data classes.

Severity

Instructions

- none*
code is not affected by the smell
- minor*
code is only partially affected by the smell
- major*
the smell characteristics are all present in code
- critical*
the smell is present, and has particularly high values of size, complexity or coupling



Grading procedure

Selecting sample

Grading sample

Verification

Arbitrage



Dataset metrics

- ▶ Reviews: 14739
- ▶ Samples: 4770
- ▶ Projects: 523
- ▶ Reviewers: 26
- ▶ Smells: 4



Industry-relevance

- ▶ research was done as an industrial project,
- ▶ we used industry-relevant repositories,
- ▶ samples were graded by experienced developers

