

UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Stanislav Mõškovski

Building a tool for detecting code smells in Android application code

Master's Thesis (30 ECTS)

Supervisor: Kristiina Rahkema, MSc

Supervisor: Dietmar Pfahl, PhD

Tartu 2020

Building a tool for detecting code smells in Android application code

Abstract:

Write abstract text here

Keywords:

List of keywords

CERCS:

CERCS code and name: <https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e>

Building a tool for detecting code smells in Android application code

Abstract:

Write abstract text here

Keywords:

List of keywords

CERCS:

CERCS code and name: <https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e>

Contents

1	Introduction	4
1.1	Research context	4
1.2	Research motivation	4
1.3	Thesis outline	4
2	Background	5
2.1	Code smells	5
2.2	Definitions of investigated code smells	5
2.3	Related work	5
2.4	SonarQube	5
3	Method	6
3.1	Analysis tools	6
3.1.1	SonarQube plugin	6
3.1.2	SonarQube project bulk analyzer	6
3.2	Selected datasets	7
3.3	Methodology	7
4	Results	8
4.1	Developed tools	8
4.2	Analysis results	8
5	Conclusion	9
	References	9
	Appendix	10
	I. Glossary	10
	II. Licence	11

1 Introduction

1.1 Research context

Describe what code smells are. Describe how code smells are different from bugs. Shortly about previous research and how we plan to be different.

1.2 Research motivation

Describe why solution proposed in this thesis is useful. Goals of the thesis:

- Develop a tool, describe why it would be useful from different perspectives (developers, project managers, data scientists)
- Extend the body of knowledge about the occurrence of code smells in Android applications (extend the number of code smells, provide analysis results, compare the results with already published results, additional results for code smells not yet published in the literature)

1.3 Thesis outline

Shortly describe structure of the thesis. What does each chapter tell the reader?

2 Background

2.1 Code smells

Describe code smells in general, what are they, how were they found at first. Describe how to fix code smells. Describe why would you want to fix them.

2.2 Definitions of investigated code smells

Here we need to describe definitions of code smells that we investigated. Problem: code smells were defined by Kristiina, but no paper has been published yet?

2.3 Related work

Describe existing tools. Discuss their results and implementations. Here we can describe the same 3 tools that were used during the seminar: paprika, infusion and anti patterns code smells plugin for SonarQube.

2.4 SonarQube

Describe what is SonarQube. Describe why was SonarQube chosen as implementation platform. Describe how can SonarQube be extended. Describe what does it mean to write a plugin for SonarQube: extension points (sensor/rule), what are the possibilities for the user (enabling/disabling rules), possibility to run both server side and inside an IDE (SonarLint).

3 Method

3.1 Analysis tools

Describe tools to be used in analysis.

3.1.1 SonarQube plugin

Describe plugin for SonarQube (what does it do, how does it work, what can be configured, show end user can use the plugin, architecture of the plugin). What do consider: how can the user select what rules to enable, context of a single rule (severity etc)

3.1.2 SonarQube project bulk analyzer

Describe the tool that was developed for bulk project analysis. Describe its functionality:

- Parse the input corpus file
- Clone the git projects
- Build the projects (using gradle/maven), here we can also mention that we use wrapper if it is provided with the project and if the wrapper is not provided we use default version installed on the system.
- In case of Gradle, we have to modify the build script to include the SonarQube analysis plugin.
- Create projects in the SonarQube server (with configured profile)
- Run the analysis on the projects (data is uploaded by the SonarQube plugin itself)
- To implement: fetching results from the SonarQube once all of the projects are analyzed.

3.2 Selected datasets

Describe how we selected the dataset. We might be using the dataset that was used by the authors of another paper, so that we can compare our results to those that they have already provided. Also mention here that some of the projects in our corpus were not using the build system, so we were not able to analyze them. Here we can say that we excluded them because we could build them, but SonarQube itself cannot analyze projects that do not use build systems.

3.3 Methodology

Describe that we want to perform analysis of projects in order to:

- see how code smells that we implemented are distributed inside analyzed applications
- see how our code smells definitions compare to already published results
- see how code smells not yet published by the literature are distributed inside analyzed applications

Here we also need to describe the method that we will use to analyze the applications. For example for a each application:

1. Build the project
2. Analyze the project
3. Extract what code smells were found / how many were found (statistics)

Also for some code smells we need to determine some parameters statistically (using box plot technique). This section would need to describe how we would find those parameters statistically.

4 Results

4.1 Developed tools

Here describe plugin for SonarQube. In introduction we mentioned groups that we think the tool might be used for. So here we provide our ideas how each group can be helped with our tool, provide screenshots or other artifacts that might help our points.

Also describe the results of developing the bulk analyzer, provide simple instructions on how to run this tool, provide output from help command, which will show input parameters and basic instruction on how to run.

4.2 Analysis results

Describe the results that we got from project analysis. Say how those compare to already existing results. Distribution of code smells that are not yet published. How many projects in corpus versus how many were actually successfully analyzed. Statistics on code smell distributions inside analyzed applications.

5 Conclusion

Say that we have created a tool and it works on the projects that we checked, but we dont know if it actually helps, since we did not perform any empirical study. Say that there might be some limitations with the dataset that we have selected. Discuss future work.

Appendix

I. Glossary

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Stanislav Mõškovski**,
(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Building a tool for detecting code smells in Android application code,
(title of thesis)

supervised by Kristiina Rahkema and Dietmar Pfahl.
(supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Stanislav Mõškovski
dd/mm/yyyy