LOG6306-Patrons pour la compréhension de programme
TP: Replication study

Biruk Asmare, Muse
biruk-asmare.muse@polymtl.ca

September 21, 2021

1 Introduction

In this TP, you will conduct a Literal and operational replication study of the paper: **On the Distribution of Test Smells in Open Source Android Applications: An Exploratory Study** by Anthony Peruma et al [3]. This paper reports an exploratory study on the prevalence and persistence of unit test smells in android open source applications. We have production code and test code in software. The production code holds the main logic of the application while the test code is a special program that tests a production code. Unit test smells are code smells that occur in unit test code. They are different in type and nature from traditional code smells (code smells that are not related to test codes).

The purpose of a replication study can be to verify previously founded results or to improve the generalizability of the result to other contexts. There are many approaches to classify replication studies in software engineering and other disciplines [1]. However for this TP, we consider literal replication study and operational replication study.

1.1 Literal replication

In this type of replication study, the independent and dependent variables are duplicated as close as possible to the original study. We keep the context of the study as close as possible. The objective of this type of replication study is to verify the previously found results and conclusions.

1.2 Operational Replication

In this type of replication, we can vary research questions and change the way data is collected and analyzed. We could change the subject systems, some aspects of the methodology, or extend the study with new experiments.

2 Instructions

- This TP is group work. We will have four different replication studies. A maximum of 3 and minimum of 2 students per group are allowed.
- The four variations of the replication study are as follows

G1: *Literal replication* This group conducts the literal replication of the paper to validate the original results. Two students will work on this.

G2: *Operational replication changing the subject systems to python.*

This group conducts an operational replication by changing the subject systems from java (android) projects to python projects. Two students will work in this group.

G3: *Operational replication with extension*:

This group conducts operational replication by doing literal replication like G1 and adding extra research question to investigate the co-occurrence between test smells and traditional smells in java subject systems. Three students will work in this group.

G4: *Operational replication with extension using python subject systems*

This group conducts operational replication by doing replication like G2 and adding extra research question to investigate the co-occurrence between test smells and traditional smells in python subject systems. Three students will work in this group.

- All groups will not do the survey and manual analysis mentioned in section 2.

3 Smell detection tools

All the tools necessary for this replication study will be provided. However, You will also need to implement your own scripts for data-collection, integration of the smell detection tools and analysis of results. The details of the tools and some data-collection practices will be described in the lab.

- Test smell extraction and detection tools
  1. **Ts-detect** [1]: smell detection tool for java applications
  2. **Test smell detection for python** [2]
  3. **Test file detector**[3]. This tool detects test files from java projects. The tool can be adapted to work with python.
  4. **Test file mapping** [4]: This script locates production files given test files. This script can be adapted to work with python.
- Traditional smell detection tools
  - **DECOR** [2] traditional code smell detection for java. The jar file will be provided.
  - **Pysmell**[5]. Traditional code smell detector for python.
- **PyDriller** [6] Python library that provides an API to extract information from local or remote GitHub repositories.

[1] https://github.com/TestSmells/TestSmellDetector
[2] https://github.com/TestSmells/PythonTestSmellDetector
[3] https://github.com/TestSmells/TestFileDetector
[4] https://github.com/TestSmells/TestFileMapping
[5] https://github.com/chenzhifei731/Pysmell
[6] https://pydriller.readthedocs.io/en/latest/

4 Deliverable

- An article in IEEE format detailing your report of the replication study. The contents of the article will be discussed in the lab
- Replication package: A GitHub repository containing all the materials necessary to replicate your study. Materials such as data collection and analysis scripts and raw datasets.

5 Before coming to lab

- We are going to use python as a primary language for data collection as well as for analysis. Make sure you have a working python environment that supports Jupyter notebooks to follow the lab. I recommend installing Anaconda [7] to get the python environment as well as many data analysis libraries including Pandas.
- You need to have Java run-time environment to run the java smell detection tools in your machine
- Install **git** on your system
- Read the original paper carefully
- Please from a group as described in the instructions. Two students per group for G1 and G2 and three students for G3 and G4.

References

[1] Omar S. Gómez, Natalia Juristo, and Sira Vegas. "Replications Types in Experimental Disciplines". In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM '10. Bolzano-Bozen, Italy: Association for Computing Machinery, 2010. ISBN: 9781450300391. DOI: 10.1145/1852786.1852790. URL: https://doi.org/10.1145/1852786.1852790.

[2] Naouel Moha et al. "DECOR: A Method for the Specification and Detection of Code and Design Smells". In: *IEEE Transactions on Software Engineering* 36.1 (2010), pp. 20–36. DOI: 10.1109/TSE.2009.50.

[3] Anthony Peruma et al. "On the Distribution of Test Smells in Open Source Android Applications: An Exploratory Study". In: *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*. CASCON '19. Toronto, Ontario, Canada: IBM Corp., 2019, pp. 193–202.

[7] https://www.anaconda.com/