UNIVERSITY OF PASSAU
Chair of AI Engineering
https://www.fim.uni-passau.de/ai-engineering/

Mining Software Repositories Lab
WS 2022/2023
Prof. Dr. Steffen Herbold, Dr. Alexander Trautsch, Lukas Schulte

Exercise 1 · Due at 2023-03-03

General information

The task groups for specific days are a rough estimate. If you are finished with tasks for Monday you should start on the tasks for Tuesday. The faster you are working through the extraction and data analysis the faster you can explore more data and start on the presentation and maybe even prepare information for the final report.

Be aware that the tasks described for each weekday are only a minimal solution. You should explore more options. Some suggestions: are the hack patterns used here actually good? Are there other publications that use a different set? Are there differences between projects regarding SATD? Are there other projects not mentioned here that you can additionally mine? These are just some questions that you can explore and mention in the presentation of the results and in the final report.

Tasks

Research questions this week:

RQ1 Does SATD in code comments correlate with high code complexity?
Investigate code comments and corresponding code regarding SATD and complexity.

RQ2 Does SATD in commit messages correlate with high code complexity?
Investigate commit messages for SATD related keywords, inspect changes for complexity.

Monday

1. Checkout a repository[1] to a local folder as a test.
2. Create a Python script which lists the Java files of the repository.
3. Use Lizard[2] to extract cyclomatic complexity for the methods of each file.
4. Include our provided plugin for extracting SATD comments within methods[3]. It uses the patterns provided in previous research[4].
5. Extend the Python script so that it uses the plugin and you can extract cyclomatic complexity, name and SATD comments for each method from a file.

[1] https://github.com/apache/commons-net
[2] https://github.com/terryyin/lizard
[3] https://git.fim.uni-passau.de/trautsch/lizard-satd-plugin
[4] http://users.encs.concordia.ca/~eshihab/data/ICSME2014/satd.html

Tuesday

1. After you have your group, create a private Gitlab Repository[5] for your group for the week and invite the lecturer.

2. Extend the Python script so that you can run it over all commits of a repository using PyDriller. Include for each file before and after versions as done in the first week. Match the methods in the before and after versions of the file to get the before and after cyclomatic complexity as well as SATD comments for each method.
   Hint: If a match between a before and after method cannot be found, drop it from the results.
   Required: Only collect the methods which have changed in the commit.

3. Extend the Python script so that you match the patterns also on the commit message. You can include this as boolean feature, i.e., whether the commit message also contained SATD.
   Required: Include also the commit hash in the data so that you can distinguish between commits in the analysis later.

4. Your Python script should be able to take a repository URL, clone the repository and traverse over all commits. For each modified (not added, deleted, copied) Java file extract all methods, as well as before and after cyclomatic complexity and before and after SATD comments.

5. Make sure to extract to a format which you can later analyze.

6. Extract at least the changes of commons-net[6], commons-compress[7], commons-vfs[8] and commons-codec[9].

Wednesday

1. Create a new Python environment in a new folder notebooks. Install Jupyterlab, Pandas, Scipy, Matplotlib and Scikit-learn into the new environment.

2. Create a Jupyter notebook and load the existing data.

3. For RQ1 compare the complexity change in file changes where the number of SATD comments in the file increases with changes where that is not the case.

4. Conduct appropriate statistical tests to evaluate whether the change you see in the boxplot is statistically significant.
   Hint: Check sample distribution and apply an appropriate statistical test, you can refer to[10].

[5] https://git.fim.uni-passau.de
[6] https://github.com/apache/commons-net
[7] https://github.com/apache/commons-compress
[8] https://github.com/apache/commons-vfs
[9] https://github.com/apache/commons-codec
[10] https://sherbold.github.io/intro-to-data-science/11_Statistics.html

Thursday

1. For RQ2 compare compare the complexity change over the whole commit from commits with SATD comments with every other commit.
   Hint: Check sample distribution and apply an appropriate statistical test, you can refer to[11].
   Hint: Depending on the data, think about whether you have enough data to answer the question.

2. Finish up extraction and evaluation tasks and make sure you have updated the code on your Gitlab for this project with the current version.

3. Make sure that you can answer the research questions using your findings from this week.

4. Prepare a presentation with your results. Make sure you include a sound motivation, describe your methodology, data analysis and results as well as implications.

Friday

1. Finish up last changes.

2. Send your presentation to the lecturer prior to your presentation slot.

3. Present your results.

[11] https://sherbold.github.io/intro-to-data-science/11_Statistics.html