Why-We-Refactor Dataset

Danilo Silva
Universidade Federal de Minas Gerais, Brazil
danilofs@dcc.ufmg.br

Nikolaos Tsantalis
Concordia University
Montreal, Canada
tsantalis@cse.concordia.ca

Marco Tulio Valente
Universidade Federal de Minas Gerais, Brazil
mtov@dcc.ufmg.br

1. LICENSE

2. URL

http://aserg-ufmg.github.io/why-we-refactor

3. SCORECARD

3.1 Insighful

There is a limited number of studies investigating the real motivations driving the refactoring practice based on interviews and feedback from actual developers. Kim et al. [2] explicitly asked developers "in which situations do you perform refactorings?" and recorded 10 code symptoms that motivate developers to initiate refactoring. Wang [3] interviewed professional software developers about the major factors that motivate their refactoring activities and recorded human and social factors affecting the refactoring practice. Both studies were based on general-purpose surveys or interviews that were not focusing the discussion on specific refactoring operations applied by the developers, but rather on general opinions about the practice of refactoring.

This is the first dataset investigating *the motivations behind refactoring based on the actual explanations of developers on specific refactorings they have recently applied*.

3.2 Useful

This dataset is important for two main reasons:

It can serve as a *benchmark* for evaluating refactoring reconstruction tools (i.e., tools that recover refactoring operations from the change history of projects), since it includes a list of 539 commits from 185 GitHub-hosted Java projects with 12 different types of refactorings detected by the RefactoringMiner tool and confirmed with manual inspection. Moreover, in 222 of these commits the detected refactorings have been confirmed by the developers themselves, who additionally provided explanations about the reasons behind the application of these refactorings. To the best of our knowledge, this is the most extensive and reliable refactoring dataset available in the literature. Previous datasets with documented refactorings [1] have been created based on refactorings applied by the paper authors in open-source systems. The detection of such refactorings is less challenging for refactoring reconstruction tools, because the changes corresponding to the refactoring operations are isolated and not tangled with changes corresponding to other maintenance activities (i.e., bug fixes, feature additions).

The dataset provides insight on the actual motivations behind the application of refactorings as explained by the developers themselves, breaking several *misconceptions* around the refactoring practice. For example, most refactoring recommenders have been designed based on the concept that developers extract methods either to eliminate code duplication, or decompose long methods. In the dataset, we report 11 different reasons behind the application of Extract Method refactorings. Each motivation requires a different strategy in order to detect suitable refactoring opportunities. Having a list of motivations driving the application of refactorings can help researchers and practitioners to develop refactoring recommenders tailored to the actual needs of the developers, and thus promote more effectively the refactoring practice to the developer community.

3.3 Usable

The dataset is publicly available at GitHub, as a navigable web site, where researchers and practitioners can view and explore each commit including a refactoring with a motivation inferred in our study. We also provide the collected information in JSON format, to facilitate its import and use by other tools.

4. REFERENCES

[1] O. Chaparro, G. Bavota, A. Marcus, and M. Di Penta. On the impact of refactoring operations on code quality metrics. In *Proceedings of the 2014 IEEE International Conference on Software Maintenance and Evolution*, pages 456–460, 2014.

[2] M. Kim, T. Zimmermann, and N. Nagappan. An empirical study of refactoring challenges and benefits at Microsoft. *IEEE Trans. Softw. Eng.*, 40(7), July 2014.

[3] Y. Wang. What motivate software engineers to refactor source code? evidences from professional developers. In *IEEE International Conference on Software Maintenance*, pages 413–416, Sept 2009.

Why-We-Refactor Dataset

Danilo Silva
Universidade Federal de Minas Gerais, Brazil
danilofs@dcc.ufmg.br

Nikolaos Tsantalis
Concordia University
Montreal, Canada
tsantalis@cse.concordia.ca

Marco Tulio Valente
Universidade Federal de Minas Gerais, Brazil
mtov@dcc.ufmg.br

1. URL

http://aserg-ufmg.github.io/why-we-refactor

2. LICENSE

Danilo Silva, Nikolaos Tsantalis, Marco Tulio Valente. Why We Refactor? Confessions of GitHub Contributors. In 24th International Symposium on the Foundations of Software Engineering (FSE), 2016.

3. DESCRIPTION

First, the dataset includes a list of 539 commits from 185 GitHub-hosted Java projects with refactorings identified by the RefactoringMiner tool and confirmed with manual inspection. The detected refactorings covers 12 different types:

1. Extract Method (EM)
2. Move Class (MC)
3. Move Attribute (MA)
4. Rename Package (RP)
5. Move Method (MM)
6. Inline Method (IM)
7. Pull Up Method (UM)
8. Pull Up Attribute (UA)
9. Extract Superclass (ES)
10. Push Down Method (DM)
11. Push Down Attribute (DA)
12. Extract Interface (EI)

Second, the dataset includes a list of 222 commits with refactorings and their motivations. This is the subset of the commits with refactorings where the developers answered to our mails, describing the reasons for performing the detected refactorings. The motivation theme for each refactoring was proposed after analysing all developers' answers using thematic-analysis. The dataset also includes information about the IDE used to perform the refactorings (if reported by the developer). In case the refactoring is performed manually, we also include a possible reason for not using the IDE refactoring tool support.

Figure 1 illustrates one of these commits. In this example, performed in the NEO4J project, there are two refactorings. The first one is a EXTRACT SUPERCLASS, in which the superclass `AbstractLuceneIndexAccessorReaderTest` was extracted from two other classes. This refactoring was performed to *Extract common state/behavior*. The second refactoring, in which the method `query` was extracted from two other methods of class `LuceneIndexAccessorReader`, was performed to *Remove duplication*. Additionally, in this example the refactoring was performed manually (*Manual* tag) due to a *Lack of trust* in refactoring tools. Finally, the IDE is *Unknown* because it was not reported by the developer. The dataset also includes meta-data about the commits (SHA1 hash, author, date, and time).

The dataset is publicly available at GitHub, as a navigable web site, where researchers and practitioners can view and explore each commit including a refactoring with a motivation inferred in our study. We also provide the collected information in JSON format, to facilitate its import and use by other tools.

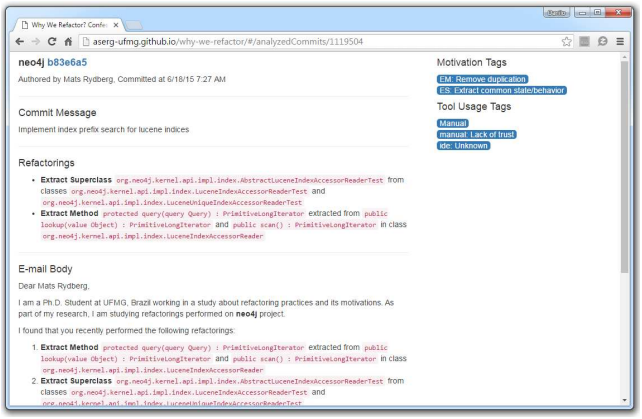

Figure 1: An example of a commit with refactorings and their motivations.