

## Qualitative Analysis

*Round 1, June 21 – July 09, 2021*

### *Topics*

*Method of analysis*  
*Sample 1*  
*Concluding remarks*  
*Research questions*  
*Suggestions*  
*Notes*  
*References*

## Method of Analysis

We adopted a non-probability sampling, *purposive sampling*, to select refactoring-inducing and non-refactoring-inducing PRs, according to specific criteria, for an in-depth investigation of their review comments and discussion while cross-referencing their detected refactoring edits. We also validated these refactorings by checking for false positives and false negatives.

It is worth clarifying that we will follow that strategy until we reach a point of data saturation (when no new information emerges), through rounds of analysis. Accordingly, at each round, we will examine a purposive sample fitting a valuable scenario to the current purposes of the analysis. We chose that sampling strategy because it provides us gaining an in-depth understanding of data by exploring scenarios suitable at each step, in line with emergent patterns or ideas.

We experimentally considered 20 as the size for the first purposive sample: ten refactoring-inducing PRs and ten non-refactoring-inducing PRs. The median value of review comments in refactoring-inducing PRs is five, and the median value of reviewer is two in both refactoring-inducing and non-refactoring-inducing PRs. Thus, looking for a fair comparison between groups when addressing RQ<sub>1</sub> and RQ<sub>2</sub>, we randomly selected refactoring-inducing and non-refactoring-inducing PRs that contain five review comments and two reviewers for the first round of analysis from 20 refactoring-inducing PRs and 61 non-refactoring-inducing PRs.

Then, each analyst apart examined all sample's PRs. Next, one analyst checked all individual judgments in order to achieve a concluding judgment concerning the sample. As decision criteria, we considered the agreement of responses by at least two analysts. It is worth clarifying that, in such subjective decision-making, we considered the refactoring-inducement in settings where review comments either explicitly suggested refactoring edits or left any actionable recommendation that induced refactoring. We give examples as follows.

## Sample 1

- 13 refactoring-inducing PRs
- 7 non-refactoring-inducing PRs
- Number of review comments: 100
- Number of discussing comments: 68
- Number of subsequent commits: 40
- Number of refactoring edits: 68

## Concluding Remarks

- (1) 10/13 (76.9%) of refactoring-inducing PRs are due to code review.
- (2) 3/10 (30%) of “previously categorized as” non-refactoring-inducing PRs were false negatives (all refactoring edits due to code review): RefactoringMiner 1.0 should but did not detect 2 refactoring edits (*Extract Variable* and *Extract Method*); 2 instances of *Change Method Access Modifier* are not detectable by such version. Accordingly, TP = 66, FP = 0, and FN = 2 (precision 100%, recall 97.06%).
- (3) 1/13 (7.7%) of refactoring-inducing PRs present self-affirmed refactorings (two edits done by a contributor). Such edits are explicit in the commit message.
- (4) Both Refactoring-inducing and non-refactoring-inducing PRs comprise the three primary types of changes (adaptive, corrective, and perfective). Our classification follows the descriptions given by [Swanson, 1976] and the identification method provided by [Mockus and Votta, 2000]. To clarify, we explored PR descriptions and commit messages by searching for keywords that could denote the type of changes; for instance, keywords such as “*fix*” and “*correct*” indicate corrective changes. Specifically, in both refactoring-inducing PRs consisting of corrective changes, in which refactoring edits were induced by code review (Fluo #837, Flink #7970, Flink #9143) or led by PR's author (Flink #7971), we found only Rename and Change Type instances. In refactoring-inducing PRs comprising adaptive changes, we found diversified kinds of refactoring, such as Extract (Hadoop #942) and Move and Rename (Flink #7945) besides Rename and Change Type instances (Samza #1030, Flink #7945, Dubbo #2279).
- (5) 5 refactoring-inducing PRs present a code review bot. A repository's code review bot is easily detectable since it left a comment in the PR, including the bot commands performed. For instance, the *flinkbot* checks the PR description, whether a PR needs attention from a specific reviewer, the architecture, and the overall code quality.
- (6) It seems that there is no influence of PR description on both refactoring-inducing and non-refactoring-inducing PRs. Likewise, we found well-explained PR descriptions and none PR descriptions.
- (7) We found self-affirmed minor PR (in which, a title or a description self-declares a minor PR) in 1/13 refactoring-inducing PR (Kafka #5194) and in 2/7 non-refactoring-inducing PRs (Brooklyn-server

#411, Kafka #5111). Moreover, we found self-affirmed minor review comments (in which, a reviewer declares a review comment as minor) in 1/13 refactoring-inducing PR (Flink #7945) and in 2/7 non-refactoring-inducing PRs (Brooklyn-server #4111, Flink #91).

- (8) We note that reviewers present experience in code review as of 2016, except 1 reviewer (Flink PR #91) whose activity is not included in his contributions profile. We believe this is due to the inclusion of more contributing activities on GitHub in 2016. As such, there may be more experienced reviewers than noted.
- (9) In this sample, non-refactoring-inducing PRs were not reviewed on Wednesdays.
- (10) In refactoring-inducing PRs, in which code review induced refactorings, *Change Type* (26/52) and *Rename* (13/52) instances are the most frequent refactoring kinds, whereas *variable* (14/52) and *method* (11/52) are the most frequent refactoring targets in sample 2. In refactoring-inducing PRs, in which authors led refactorings, *Change Type* (9/17) and *Rename* (8/17) instances occurred, being *parameter* (5/17) and *method* (4/17) the most frequent refactoring targets.
- (11) When considering the number of contributions of the PRs' participants, it seems that refactoring-inducing PRs authors are less experienced than non-refactoring-inducing PRs authors (Table 1). Also, it seems that reviewers of refactoring-inducing PRs are slightly more experienced than reviewers of non-refactoring-inducing PRs (Table 1, Table 2). Accordingly, we conjecture that refactoring-inducing PRs' authors, because they are more experienced, develop less problem-prone code and, therefore, less refactoring-prone code.

<i>Table 1 – Stats of PRs' participants by category</i>			
Category	Stats	Author	Reviewer
<i>Refactoring-inducing PRs</i>	Average	668.9	2,801.6
	SD	853.3	3,417.6
	Median	424	1,414
	IQR	794	3,698
	Outliers	3,127	14,442
<i>Non-refactoring-inducing PRs</i>	Average	5,375.6	2,907.6
	SD	10,982.5	4,358.2
	Median	511	1,373
	IQR	12,862.5	1,228
	Outliers	none	12,678

<i>Table 2 – Apache roles of PRs' participants by category</i>		
Category	Author	Reviewer
<i>Refactoring-inducing PRs</i>	2 committers	5 PMC

	1 commiter/PMC	2 committers 5 committers/PMC 1 PMC 'incubating'
<i>Non-refactoring-inducing PRs</i>	1 committer/PMC	2 PMC 4 committer/PMC

## Research Questions

- **RQ<sub>1</sub> How are refactoring-inducing and non-refactoring-inducing PR's review comments characterized?**

### *Refactoring-inducing PRs*

1 refactoring-inducing PR presents refactoring edits submitted by a committer and PMC, in which review comments deal with code style.

Reviews in 7 refactoring-inducing PRs, where refactoring edits were induced by code review, address code logic (6 PRs), and present reasons (1 PR).

Reviews in 2 refactoring-inducing PRs, where refactoring edits were submitted by the authors (fixing a typo in 1 PR), directly address code logic.

Reviews in 3 refactoring-inducing PRs, where refactoring edits were induced by both code review and authors, suggest improvements and respective reasons.

### *Non-refactoring-inducing PRs*

Reviews in non-refactoring-inducing PRs directly suggest the insertion of tests and address simple code logic (e.g., correct use of a method, correct range of variable's values, and correct constant), typos, code aesthetics, and documentation. Three PRs present no review comments in Java files.

- **RQ<sub>2</sub> What are the differences between refactoring-inducing PRs and non-refactoring-inducing PRs, in terms of review comments?**

It seems that the quality of code reviews (when suggesting improvements and reasons) denotes some influence on refactoring edits.

It is not rare for a PR to only have refactoring suggestions. Apparently, reviewers tend to suggest refactorings if they find some other suggestion related to actual behavioral changes in the code. It means that, usually, reviews in refactoring-inducing PRs address both refactoring and behavioral changing suggestions, whereas, in non-refactoring-inducing PRs, reviews address behavioral or no suggestion at all.

- **RQ<sub>3</sub> How do reviewers suggest refactorings in review comments in refactoring-inducing PRs?**

Reviewers directly address code logic or give suggestions for improvement and reasons. Also, Rename instances are directed suggested by reviewers.

It seems that reviewers submit comments only when refactoring is needed. They always provide direct comments in the code where a change should be done.

In most cases, reviewers use words like 'Can we' or 'Maybe we should' followed by the suggestion regarding the code and a brief explanation of the reason for the change, possibly to let the author know that such adjustment is not necessary for the code operation, but to bring more readability or maintain a coding standard among the other developers of the repository.

In sample 1, review comments that induce refactoring present questions on code logic (dubbo #3299, flink #7945, commons-text #39), using terms such as "should we...", "can we...", "maybe...", "any reason to...?"; warnings on good practices (fluo #837, flink #7970); and direct suggestions as a method name (dubbo #2279, hadoop #942).

In non-refactoring-inducing PRs, there is a discussion regarding the reason for refactoring. It can happen that the PR's author wins such discussion, not performing the refactoring edits.

#### ● **RQ<sub>4</sub> Do refactoring suggestions justify the reasons?**

In 4/13 (30,8%) of the PRs, the reviewers present reasons that induced refactoring edits. Reasons are contextualized sentences concerning problematic situations, sometimes accompanied by explanations using examples (such as code's structure under analysis, Flink #7970 and Kafka #5194). In other cases, as in Flink #7945, reasons begin with any question on logic ("Can we...?") assisted by examples; in other, reviewers provide a direct explanation as in Dubbo #2279. It seems that when reviewers provide any reason for refactoring, they give contextualized arguments through examples.

#### ● **RQ<sub>5</sub> What is the relationship between refactoring recommendations and actual refactorings in refactoring-inducing PRs?**

When the reviewers get straight to the point, refactoring edits are always done. For instance, in Kafka #5194, the reviewer says "The interface belongs in the package of the providers. Should be org.apache.kafka.common.config.provider.ConfigProvider. See Transformation for a similar example.", inducing a Move Class edit. Other examples comprise explanations using examples in the code under analysis (Fluo #837, Flink #7970); sentences on code conventions (Kafka #5194); direct suggestions such as "you can replace to..." (Hadoop #942).

In a few cases, when there is space for discussion, refactorings may not be done. It seems that review comments which propose changes by depicting the source code are not welcome. We speculate that such scenarios may affect code ownership. For instance, in Dubbo #4870, a reviewer suggests a method change ("should we...") by indicating such code change in the comment, nonetheless, PR's author presents an argument and a counterexample, so winning the discussion. Also, an extensive source code change indicated by a reviewer in Kafka #5111 was ignored.

Refactoring complexity is another point that deserves attention. To exemplify, in Flink #7970, a reviewer argue concerning the advantages of avoiding casts, so inducing a *Rename Class* that, in turn, promotes *Change Type* instances in that code.

Also, an *Extract Superclass* usually results from a longer explanation and discussion on why it would help.

## Suggestions

- Add new fields to the spreadsheet: *direct\_review\_comment\_flag* and *discussion\_flag*.
- Second sample: 10 refactoring-inducing PRs and 10 non-refactoring-inducing PRs that contain only one subsequent commit. Why? Intending to explore PRs consisting of only one refactoring edit (*Ramon*), we need to consider such a simple setting, which gives us an opportunity of comparing refactoring-inducing and non-refactoring-inducing PRs.

## Notes

- **Apache roles<sup>1</sup>:**
  - **Contributor** is a developer who contributes to a project in the form of code or documentation.
  - **Committer** is a contributor who has write access to the code repository.
  - **PMC member** is a committer who has write access to the code repository and can agree and approve/disapprove the changes.
- **Repositories timeline (contribution activities):**
  - hadoop – May 2009
  - flink – Dec 2010
  - brooklyn-server – Jun 2011
  - dubbo – Jun 2012
  - kafka – Dec 2012
  - fluo – Jun 2013
  - samza – Aug 2013
  - commons-text – Nov 2014
  - beam – Dec 2014
  - incubator-iotdb – May 2017
  - incubator-iceberg – Dec 2017

## Refactoring-inducing PRs:

<sup>1</sup> <https://apache.org/foundation/how-it-works.html#roles>

**flink 7971**: 2019 | commit merge | flinkbot | corrective changes | self-affirmed refactoring | **refactorings led by PR's contributor** | non-valuable review comments (aesthetics) | 2 refactoring edits

- 2° subsequent commit: 1 *Rename Parameter*
- 3° subsequent commit: 1 *Rename Method*
- 3 subsequent commits, 4 file changes, 9 added lines, 9 deleted lines, 22 days to merge
- Author, contributions since 2012 (109)
- An Apache Flink committer and PMC (contributions since 2013 (5,334), reviews since 2016) added 2 commits containing refactoring edits
- Reviewer (Apache Flink PMC), contributions since 2016 (877), reviews since 2017
- PR opened on Wednesday, reviewed on Thursday (long delay, waiting for more reviews)

**dubbo 3299**: 2019 | commit merge | 2 mentions, 1 external reference after PR merge | no bot | perfective changes (optimization) | **refactorings inspired by code review** | direct review comments (presenting a few concise questions concerning logic) | 6 refactoring edits

- 1° subsequent commit: 4 *Replace Variable with Attribute*
- 2° subsequent commit: 1 *Replace Variable with Attribute* | 1 *Change Variable Type*
- 2 subcommits, 5 file changes, 38 added lines, 47 deleted lines, 1 day to merge
- Author, contributions since 2015 (1,302)
- Contributor, contributions since 2015 (429)
- Reviewer, contributions since 2016 (828), reviews since 2017
- Reviewer (Apache Dubbo committer), contributions since 2015 (138), reviews since 2018
- PR opened on Tuesday, reviewed (inducing refactoring) on Tuesday (same day)

**fluo 837**: 2017 | squash and merge | no bot | corrective changes | **refactorings inspired by code review** | review comments (presenting a question concerning logic) | 9 refactoring edits

- 1° subsequent commit: 7 *Rename Method* | 2 *Change Parameter Type*
- 1 subcommit, 1 file change, 35 added lines, 30 deleted lines, 5 days to merge
- Author, contributions since 2011 (3,127)
- Reviewer, contributions since 2008 (3,040), reviews since 2016
- Reviewer (Apache Fluo PMC), contributions since 2014 (2,192), reviews since 2016
- PR opened on Thursday, reviewed (inducing refactoring) on Friday (1 day)

**beam 4460**: 2018 | commit merge | 1 external reference after PR merge | initial refactoring edits | no bot, PR presents a fulfilled checklist for contribution | perfective changes (simplification) | **refactorings led by the author** | direct review comments (presenting a few concise questions concerning logic) | 2 refactoring edits

- 1° subsequent commit: 2 *Change Variable Type*
- 5 subcommits, 6 file changes, 10 added lines, 10 deleted lines, 0 days to merge

- Author, contributions since 2014 (1,372)
- Reviewer (Apache Beam PMC), contributions since 2011 (5,277), reviews since 2016
- Reviewer, contributions since 2010 (4,173), reviews since 2016
- PR opened on Monday, reviewed on Monday (same day)

**flink 7970:** 2019 | commit merge | 3 external references after PR merge | initial refactoring edits | flink-bot | corrective changes | **refactorings led by both the author and code review** | valuable review comments (presenting suggestions of improvement and respective reasons) | *Rename (1)* directly suggested by a reviewer | 26 refactoring edits

- 1° subsequent commit: *1 Rename Class (by code review)* | *5 Change Attribute Type (by code review)* | *9 Change Return Type (by code review)* | *1 Rename Attribute (by the author)* | *1 Change Attribute Type (by the author)* | *8 Change Variable Type (by code review)* | *1 Change Parameter Type (by code review)*
- 1 subcommit, 18 file changes, 252 added lines, 146 deleted lines, 1 day to merge
- Author (Apache Flink committer), contributions since 2012 (173)
- Reviewer (Apache Flink PMC), contributions since 2015 (364), reviews since 2016
- PR opened on Wednesday, reviewed (inducing refactoring) on Wednesday (same day)

**samza 1030:** 2019 | commit merge | no bot | adaptive changes (adding a new feature) | **refactorings led by the author (typo)** | direct review comments (presenting a few questions concerning logic) | 1 refactoring edit

- 1° subsequent commit: *1 Rename Variable*
- 1 subcommit, 1 file changes, 3 added lines, 3 deleted lines, 1 day to merge
- Author, contributions since 2010 (424)
- Reviewer, contributions since 2010 (1,388), reviews since 2017
- PR opened on Monday, reviewed on Wednesday (2 days)

**kafka 5194:** 2018 | commit merge | 2 external references after PR merge | initial refactoring edits | no bot | perfective changes (optimization) | self-affirmed minor PR | **refactorings inspired by code review** | direct review comment (presenting reasons) | 2 refactoring edits

- 1° subsequent commit: *2 Move Class*
- 3 subcommits, 21 file changes, 27 added lines, 16 deleted lines, 0 days to merge
- Author, contributions since 2012 (496)
- Reviewer (Apache Kafka committer), contributions since 2011 (4,155), reviews since 2016
- Reviewer, contributions since 2015 (2,521), reviews since 2016
- Reviewer, contributions since 2007 (14,422), reviews since 2016
- Contributor (Apache Kafka committer and PMC), contributions since 2015 (1,440), reviews since 2016
- PR opened on Monday, reviewed (inducing refactoring) on Monday (same day)



**flink 7945**: 2019 | commit merge | 1 external reference after PR merge | initial refactoring edits | flinkbot | adaptive changes (new feature) | **refactorings led by both the author and code review** | valuable review comments (presenting suggestions of improvement and respective reasons) | *Rename (1)* directly suggested by a reviewer | self-affirmed minor review comments | 11 refactoring edits

- 1° subsequent commit: *1 Change Return Type (by the author)*
- 2° subsequent commit: *1 Rename Class (by the author)* | *1 Change Attribute Type* | *4 Change Parameter Type* | *1 Rename Method (by the author)*
- 3° subsequent commit: *2 Move and Rename Class (by code review)* | *1 Rename Class (by code review)*
- 3 subcommits, 27 file changes, 589 added lines, 178 deleted lines, 1 day to merge
- Author (Apache Flink committer and Apache Beam committer), contributions since 2014 (155)
- Reviewer, contributions since 2011 (233), reviews since 2016
- PR opened on Saturday, reviewed (inducing refactoring) on Monday (2 days)

**flink 9143**: 2019 | commit merge | flinkbot | corrective changes | **refactorings led by code review** | direct review comments (presenting a subtle question concerning logic) | *Rename (3)* directly suggested by a reviewer | 3 refactoring edits

- 1° subsequent commit: *2 Rename Variable*
- 2° subsequent commit: *1 Rename Method*
- 2 subcommits, 2 file changes, 12 added lines, 10 deleted lines, 0 days to merge
- Author (Apache Flink committer and PMC), contributions since 2012 (495)
- Reviewer (Apache Flink committer and PMC), contributions since 2013 (6,039), reviews since 2016
- PR opened on Wednesday, reviewed (inducing refactoring) on Wednesday (same day)

**dubbo 2279**: 2018 | commit merge | no bot | adaptive and corrective changes | **refactorings led by both the author and code review** | direct review comment (presenting reasons) | *Rename (3)* led by the author | 4 refactoring edits

- 1° subsequent commit: *1 Change Variable Type (by code review)* | *1 Rename Variable (by code review)*
- 2° subsequent commit: *2 Rename Method (by the author)*
- 2 subcommits, 6 file changes, 64 added lines, 56 deleted lines, 3 days to merge
- Author, contributions since 2016 (550)
- Reviewer, contributions since 2015 (519), reviews since 2017
- Reviewer (PMC ‘incubating’), contributions since 2011 (120), reviews since 2018
- PR opened on Tuesday, reviewed (inducing refactoring) on Tuesday (same day)

**Non-refactoring-inducing PRs:**

**incubator-iotdb 342**: 2019 | commit merge | no bot | adaptive changes (adding a new feature) | direct review comments (presenting a few questions concerning logic in non-Java files) | None review comments in Java code

- 1 subcommit, 2 file changes, 0 added lines, 14 deleted lines, 9 days to merge
- Author, contributions since 2011 (511)
- Reviewer, contributions since 2014 (1,046), reviews since 2017
- PR opened on Saturday, reviewed on Monday (2 days)

**beam 5772**: 2018 | commit merge | 1 external reference after PR merge | no bot, PR presents a checklist for contribution | perfective changes (update) | A well-explained PR description | direct review comments concerning logic in non-Java files | None review comments in Java code (PR has no Java file)

- 1 subcommit, 1 file change, 6 added lines, 0 deleted lines, 0 days to merge
- Author, contributions since 2011 (1,248)
- Reviewer, contributions since 2011 (5,707), reviews since 2016
- PR opened on Tuesday, reviewed on Tuesday (same day)

**commons-text 39**: 2017 | commit merge | initial refactoring edits | no bot | perfective changes (update) |

No PR description | Review comments concerning a variable and a visibility modifier | **refactorings led by code review** | 2 refactoring edits

- 1<sup>o</sup> subsequent commit: 1 Extract Variable
- 2<sup>o</sup> subsequent commit: 1 Change Method Access Modifier
- 2 subcommits, 2 file changes, 4 added lines, 4 deleted lines, 3 days to merge
- Author, contributions since 2012 (391)
- Reviewer, contributions since 2010 (8,132), reviews since 2016
- Reviewer (Apache Commons PMC), contributions since 2014 (475), reviews since 2016
- PR opened on Friday, reviewed (inducing refactoring) on Monday (3 days)

**brooklyn-server 411**: 2016 | commit merge | initial refactoring edits | no bot | corrective changes (fixing faults) | self-affirmed minor PR | Review comments regarding code logic (a package-specific name) and documentation | self-affirmed minor review comments

- 1 subcommit, 1 file changes, 23 added lines, 0 deleted lines, 0 days to merge
- Author, contributions since 2010 (25,011)
- Reviewer, contributions since 2011 (12,678), reviews since 2016
- PR opened on Monday, reviewed on Monday (same day)

**kafka 5111**: 2018 | commit merge | 1 external reference after PR merge | no bot | perfective changes (update) | self-affirmed minor PR | direct review comments concerning logic (including a code inline by suggesting an option to a class, in turn, ignored by the author)

- 2 subcommits, 2 file changes, 47 added lines, 44 deleted lines, 1 day to merge

- Author, contributions since 2010 (1,035)
- Reviewer (Apache committer and PMC), contributions since 2014 (1,257), reviews since 2016
- Reviewer, contributions since 2011 (1,635), reviews since 2016
- Contributor (Apache Kafka committer and PMC), contributions since 2015 (2,274), reviews since 2016
- PR opened on Thursday, reviewed on Thursday, reviewed on Friday (2 days)

**flink 91:** 2014 | commit merge | 1 external reference after PR merge | initial refactoring edits | no bot | corrective changes (fixing faults) | direct review comments concerning code logic and typo | self-affirmed minor review comments

- 1 subcommit, 3 file changes, 5 added lines, 3 deleted lines, 0 days to merge
- Author, contributions since 2012 (241)
- Reviewer, contributions since 2012 (90), reviews since 2014
- PR opened on Thursday, reviewed on Friday (1 day)

**flink 4055:** 2017 | squash and merge | 1 mention | no bot, PR presents a checklist for contribution | corrective changes (fixing faults) | No PR description | direct review comments concerning logic in non-Java files | None review comments in Java code (PR has no Java code)

- 2 subcommits, 6 file changes, 46 added lines, 47 deleted lines, 9 days to merge
- Author (Apache Flink committer and PMC), contributions since 2013 (649)
- Reviewer (Apache Flink PMC, Apache IoTDB PMC, Apache Beam committer), contributions since 2016 (221), reviews since 2016
- Contributor (Apache Flink PMC), contributions since 2013 (1,275), reviews since 2016
- Contributor (Apache Flink committer and PMC), contributions since 2012 (1,567), reviews since 2016
- PR opened on Friday, reviewed on Friday, reviewed on Sunday (2 days)

**hadoop 942:** 2019 | commit merge | 3 external references after PR merge | hadoopbot | adaptive changes | No PR description | direct review comments concerning better coding | **refactoring led by code review** | 1 refactoring edit

- 1<sup>o</sup> subsequent commit: *1 Extract Method*
- 4 subcommits, 5 file changes, 74 added lines, 11 deleted lines, 2 days to merge
- Author, contributions since 2013 (69)
- Reviewer (Apache Hadoop committer and PMC), contributions since 2014 (674), reviews since 2014
- PR opened on Monday, reviewed on Tuesday (1 day)

**incubator-iceberg 254:** 2019 | commit merge | 2 mentions | initial refactoring edits | no bot | perfective changes (optimization) | A direct-and-clear PR description | direct review comments concerning logic and a visibility modifier | **refactoring led by code review** | 1 refactoring edit

- 2<sup>o</sup> subsequent commit: *1 Change Method Access Modifier*
- 2 subcommits, 6 file changes, 11 added lines, 21 deleted lines, 1 day to merge

- Author, contributions since 2013 (33)
- Reviewer, contributions since 2009 (4,199), reviews since 2016
- PR opened on Thursday, reviewed on Friday (1 day)

**dubbo 4870:** 2019 | commit merge | 1 review comment after PR merge | no bot | corrective changes (fixing faults) | direct review comments regarding code logic

- 1 subcommit, 2 file changes, 2 added lines, 4 deleted lines, 0 days to merge
- Author, contributions since 2013 (80)
- Reviewer, contributions since 2016 (1,373), reviews since 2017
- PR opened on Monday, reviewed on Monday (same day)

## References

Swanson, E. B. “The Dimensions of Maintenance”. In *Proceedings of the 2nd International Conference on Software Engineering*, 492–497. Washington, USA: IEEE Computer Society, 1976.

Mockus, A. and Votta, L. G. “Identifying Reasons for Software Changes Using Historic Databases”. In *Proceedings of the International Conference on Software Maintenance*. Washington, USA: IEEE Computer Society, 2000.