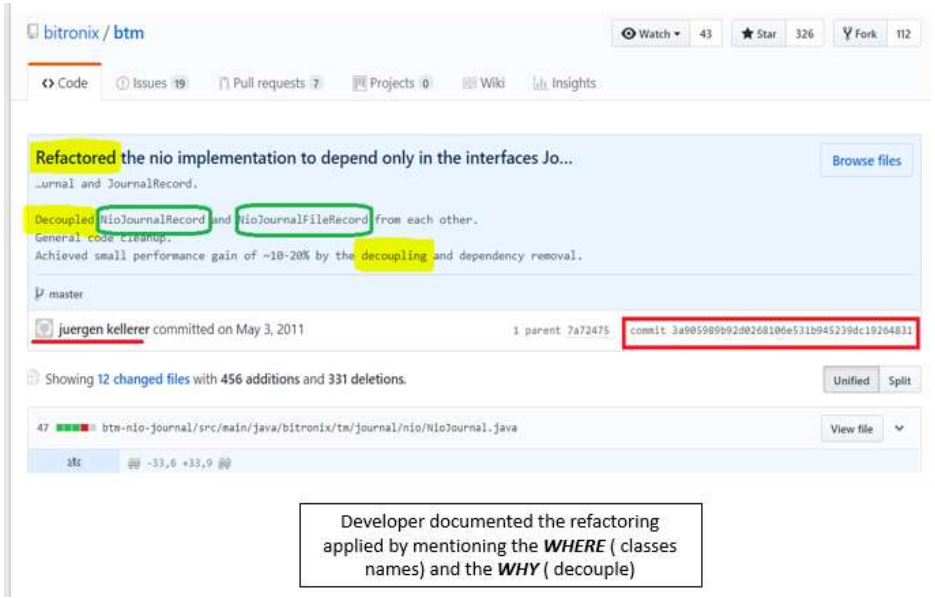
Welcome

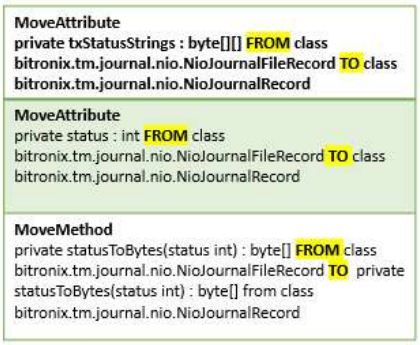
Welcome

Welcome! The objective of this survey is to understand the developers' perception of code refactoring documentation.
Filling out the survey will take about 20 minutes. Your time is much appreciated.
If you have any questions about the questionnaire or our research, please do not hesitate to contact us.

Best regards,
Soumaya Rebai
PhD candidate, Intelligent Software Engineering Research Lab, University of Michigan

Refactoring Documentation is the documentation of refactoring applied during a commit and/or pull request on GitHub.





List of applied Refactoring

Demographics

Demographics

What best describes your current position?

- ○ Developer
- ○ Tester
- ○ Manager
- ○ Team Leader
- ○ Dev Manager
- ○ _____ Other

Years of experience related to software development:

- ○ 1-5 years
- ○ 5-10 years
- ○ 10-15 years
- ○ >15 years

Level of expertise

	None	Very low	Low	Moderate	High	Very high
Continuous Integration	○	○	○	○	○	○
Refactoring	○	○	○	○	○	○
Software quality assurance	○	○	○	○	○	○
Software development (using any programming language)	○	○	○	○	○	○
Code Review	○	○	○	○	○	○

General Questions about Refactoring Documentation in practice

General Questions about Refactoring Documentation in Practice

How often do you **write** commit messages and pull request description to let your colleagues understand your code and changes?

- ○ Always
- ○ Most of the time
- ○ About half the time
- ○ Sometimes
- ○ Never

How often do you **use** commit messages and pull request description to understand the code and your colleagues' changes?

- ○ Always
- ○ Most of the time
- ○ About half the time
- ○ Sometimes
- ○ Never

How important do you think the refactoring documentation is?

- ○ Extremely important
- ○ Very important
- ○ Moderately important
- ○ Slightly important
- ○ Not at all important

How do you rate a developer's need for an automated documentation tool for refactoring?

A refactoring documentation tool could, for instance, identify the executed refactorings and include them in the commit messages.

Very low Very high

0○ 1○ 2○ 3○ 4○ 5○ 6○ 7○ 8○ 9○ 10○

List some challenges associated with refactoring documentation

- ☐ Time-consuming
- ☐ Deadlines pressure
- ☐ lack of tools to at least semi-automate the documentation
- ☐ Lost with other functional changes
- ☐ _____ Others:

To what extent do you agree that documenting refactoring and quality improvements should follow specific guidelines/best practices?

- ○ Strongly agree
- ○ Agree
- ○ Neither agree nor disagree
- ○ Disagree
- ○ Strongly disagree

Based on your experience, do developers accurately document refactoring and quality improvements ?

- ◯ Definitely yes
- ◯ Probably yes
- ◯ Maybe
- ◯ Probably not
- ◯ Definitely not

Based on your experience, are you aware of any tools to help developers document non-functional changes?

- ◯ Yes
- ◯ No

Could you please provide the name of these tools?

C#1: HOW

CI Refactoring Documentation Model

Our initial Continuous Integration (CI) Model contains 5 components.

Component #1: HOW

How developers document refactoring in practice. The answer to this question will help to understand whether developers mix functional and non-functional requirements changes when it comes to documenting refactoring.
Non-functional Requirements consist of quality changes. The definition of the **QM*OOD*** quality attributes can be found in https://ieeexplore.ieee.org/abstract/document/979986

	Always	Most of the time	About half the time	Sometimes	Never
How often do you document refactoring with functional requirements	○	○	○	○	○
How often do you usually find developers mix functional and non-functional in their refactoring documentation	○	○	○	○	○

Developer's Experience with finding/documenting the ***HOW*** component in CI environment

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult
How difficult is to document functional and non-functional requirements together	○	○	○	○	○

Importance of the ***HOW*** component

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
To what degree do you agree that mixing functional changes with non-functional changes in refactoring documentation is a good practice?	○	○	○	○	○

C#2: WHY

Component #2: the Rationale : WHY

Why these refactorings were applied i.e reasons/issues behind the changes.
For instance, changes can be explained by the quality attributes changes and its severity, and/or fixing code smells

Code Smells

Do you want to document and/or find fixed and/or introduced code smells in refactoring documentation?

- ○ Yes
- ○ Maybe
- ○ No

The definition of code smells and their different types can be found in http://www.tusharma.in/smells.

Please list the names of code smells that you think it is important to document them.

Quality Attributes Changes

QMOOD quality attributes include extendibility, reusability, functionality, flexibility, understandability, and effectiveness. The detailed description of the attributes can be found in this article (https://ieeexplore.ieee.org/abstract/document/979986).

The following example is an instance of undocumented changes in quality attributes.

As **Figure 2** shows, reusability and functionality quality attributes were significantly improved by the pull request captured in **Figure 1**. Despite the importance of the changes in the quality of the system as a whole, the developer did not include this information in his documentation.

Figure #1: https://github.com/atomix/atomix/pull/905

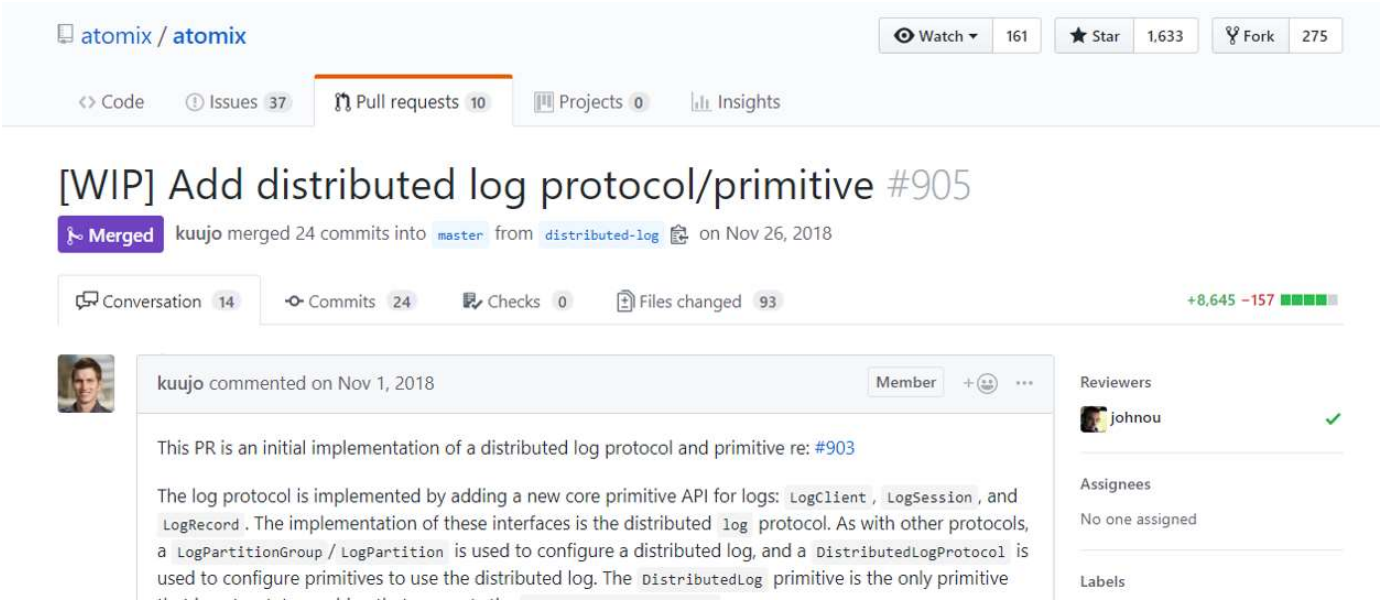
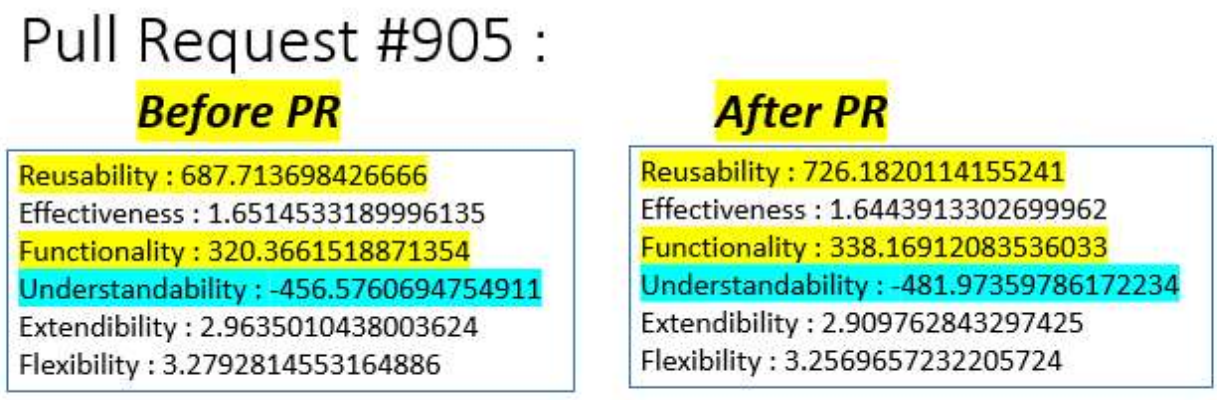


Figure #2:



Do you want to document quality attributes changes in refactoring documentation ?

- ○ yes
- ○ Maybe
- ○ No

Please choose all the QMOOD quality attributes that you think should be documented when changed

- ☐ Extendibility
- ☐ Functionality
- ☐ Reusability
- ☐ Effectiveness
- ☐ Flexibility
- ☐ Understandability

	Always	Most of the time	About half the time	Sometimes	Never
How often do you document the rationale for your applied refactoring?	○	○	○	○	○
How often do you find the rationale documented for the executed refactorings?	○	○	○	○	○

Developer's Experience with finding/documenting the ***WHY*** component in CI environment

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult
How difficult is it to find the rationale of the carried out refactoring in commit pull request messages?	○	○	○	○	○
How difficult is to document the rationale for refactoring?	○	○	○	○	○

Importance of the rationale of refactoring (the **WHY**) in refactoring documentation

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
To what degree do you agree that including the rationale (WHY) for refactoring in refactoring documentation?	○	○	○	○	○

Please share with us other reasons for refactoring that you want to document in general.

C#3: WHERE

Component #3: Refactoring Location: WHERE

Where the refactorings were introduced i.e at the package, class or method level.
The following is an example of missing refactoring location in the documentation. **Figure 1** shows an example of a commit extracted from an open-source project. The actual refactoring

applied by the developers is summarized in **Figure 2.** The commit message shown in **Figure 1** did not mention the class where the refactoring was applied.

Figure#1:
https://github.com/bitronix/btm/commit/01089cf232441613c8767adedfca888dcd924ce4

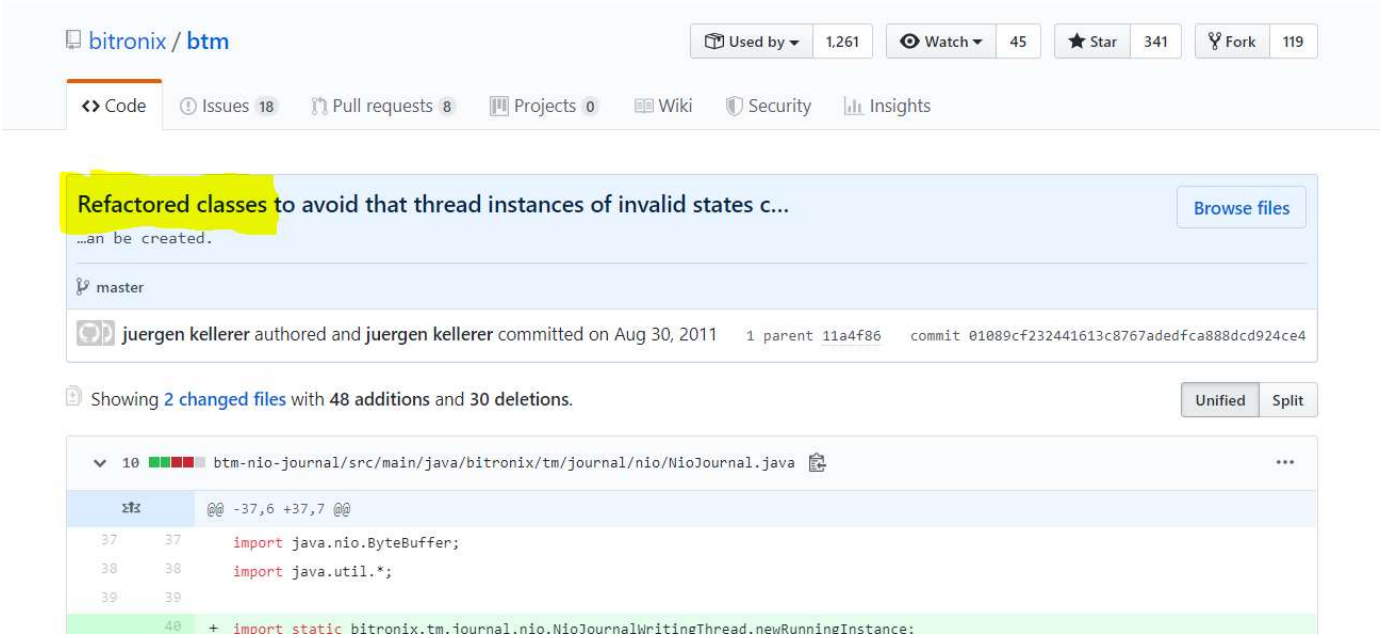
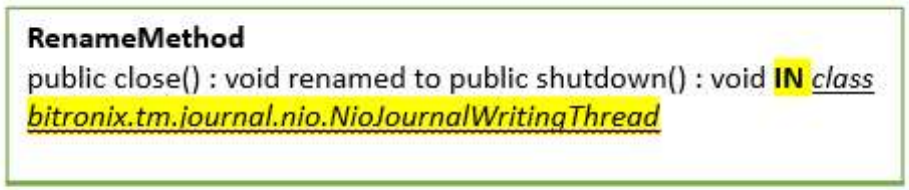


Figure #2:



	Always	Most of the time	About half the time	Sometimes	Never
How often do you document the location of your refactoring?	○	○	○	○	○

	Always	Most of the time	About half the time	Sometimes	Never
How often do you find the location of refactoring documented ?	○	○	○	○	○

Developer's Experience with finding/documenting the ***WHERE*** component in CI environment

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult
How difficult is it to find the location of refactorings (WHERE) ?	○	○	○	○	○
How difficult is to document the location of refactorings ?	○	○	○	○	○

Importance of the ***WHERE*** component

	Strongly agree	Agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree	Disagree	Strong disagre
To what degree do you agree that including the location of refactorings (WHERE) is important in refactoring documentation?	○	○	○	○	○	○	○

C#4: Refactoring Types

Component #4: Refactoring Type: WHAT

What are the types of applied refactorings i.e rename class, rename method, extract class, etc...

Figure 1 shows an example of a commit extracted from an open-source project. The actual refactorings applied by the developers are summarized in **Figure 2**. The commit message shown in figure 1 did not mention the applied refactorings.

Figure#1:
https://github.com/bitronix/btm/commit/3a905989b92d0268106e531b945239dc1926483

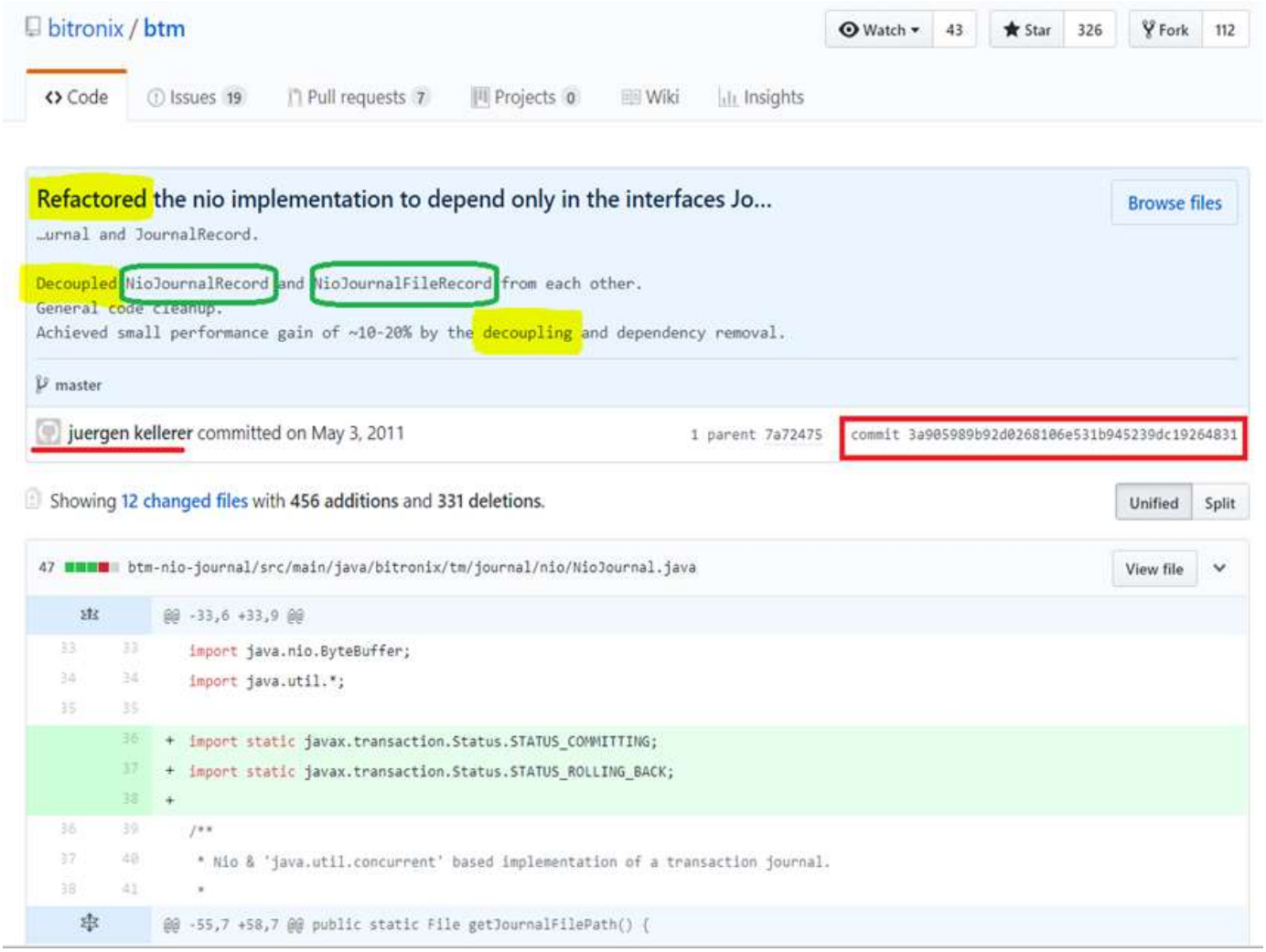
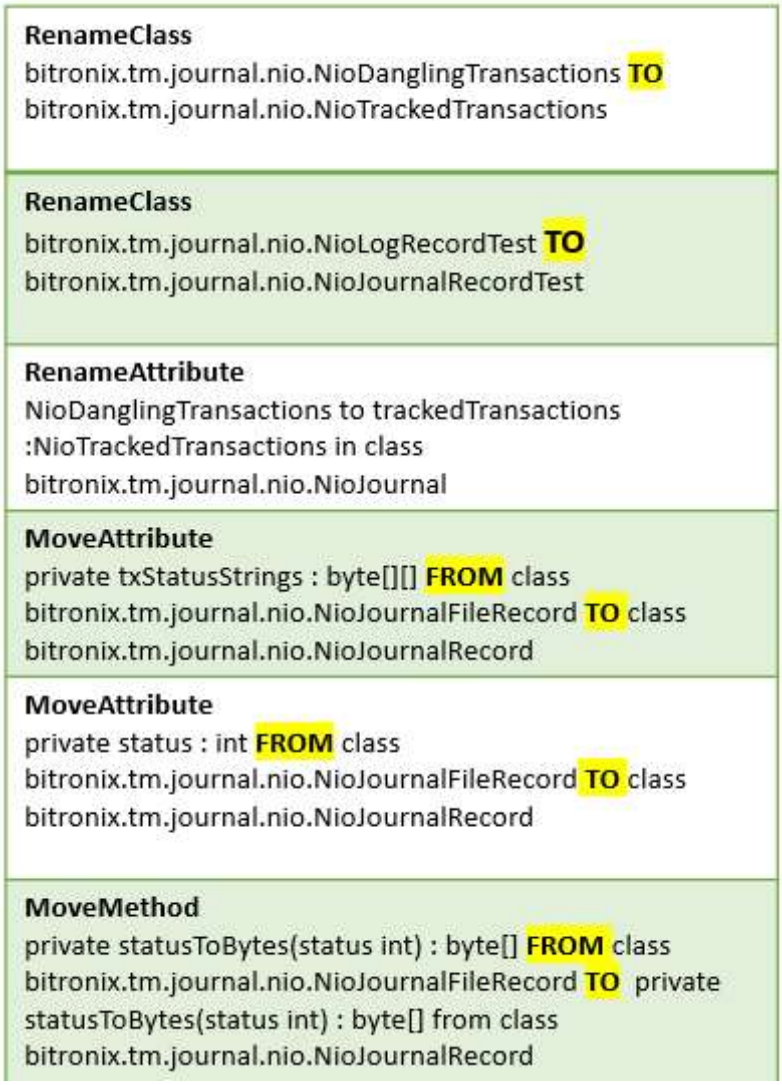


Figure #2:



	Always	Most of the time	About half the time	Sometimes	Never
How often do you document the type of your refactoring?	○	○	○	○	○
How often do you find the type of refactoring documented ?	○	○	○	○	○

Developer's Experience with finding/documenting the ***WHAT*** component in CI environment

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult
How difficult is it to find the type of refactorings in commit and pull request messages?	○	○	○	○	○
How difficult is to document the type of refactorings in commit and pull request messages?	○	○	○	○	○

Importance of the ***WHAT*** component

	Strongly agree	Agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree	Disagree	Strong disagre
To what degree do you agree that including the type of refactorings (WHAT) is important to address in refactoring documentation?	○	○	○	○	○	○	○

C#5: WHEN

Component #5: WHEN

When in the development process do developers document their refactorings.

When you tend to document refactoring

	Extremely likely	Moderately likely	Neither likely nor unlikely	Moderately unlikely	Extremely unlikely
After fixing a bug	○	○	○	○	○
Before releasing the code	○	○	○	○	○
During code review	○	○	○	○	○
After introducing major functional changes	○	○	○	○	○

How difficult is to document refactoring

	Extremely easy	Moderately easy	Neither easy nor difficult	Moderately difficult	Extremely difficult
After fixing a bug	○	○	○	○	○
Before releasing the code	○	○	○	○	○
During code review	○	○	○	○	○
After introducing major functional changes	○	○	○	○	○

Model Modification

Model Modification

Please say whether you would keep or remove the following components from the initial model

	Keep	Neutral	Remove

	Keep	Neutral	Remove
HOW: Non-Functional+ Functional requirements	○	○	○
WHY: The rationale	○	○	○
WHERE: The location of refactorings	○	○	○
WHAT: The types of refactorings	○	○	○
WHEN	○	○	○

Please list other components that you think they need to be included in the model

If you are interested in the outcome of this research study, you can leave your email here!