**Protocol of Survey for Developers**

**on Incomplete Composites**

This protocol describes how to execute our survey from pull requests and emails for developers to evaluate incomplete composites.

1. **Apply composite refactoring to remove a code smell**

1.1 Search possible incomplete composites (i.e., two or more interrelated refactorings that did not remove a certain code smell instance). This search may follow the steps of Protocol for Incomplete Composites Selection.

1.2 Understand the relation of the incomplete composite with the code smell instance that was not removed.

1.3 Select refactorings to compose the incomplete composites to remove the code smell instance according to Fowler’s recommendations [1] and Bibiano’s recommendations [2] to remove the type of this code smell instance. This selection also may be based on the understanding provided in step 1.2.

1.4 Apply the recommended refactorings on source code (method or class) that has

the code smell instance (step 1.1).

1.5 Execute Organic tool [1] on refactored source code to evaluate whether the code

smell instance (step 1.1) was removed after the application of recommended refactorings (step 1.4).

1. **Send pull requests for developers**

2.1 Submit a commit of refactored source code (step 1.4).

2.2 Open a pull request for the submitted commit (step 2.1).

2.3 Describe the pull request on the recommended composite refactoring according to the following template:

***Title: “****[ProjectName]: Refactoring to remove [CodeSmellName]”*

***Description:***

*(Greet developers)*

*“Hi guys, how are you?”*

*(Introduce the goal of the pull request) “I found a refactoring opportunity on [ProjectName] and would like to share it with you.”*

*(Introduce the code smell) “The completeCompoundConfigs () method is almost duplicated in the ReferenceConfigBase and ServiceConfigBase classes.”*

*(Detail the code smell and the recommended refactorings) “As most of the attributes used by this method belong to the AbstractInterfaceConfig class, my suggestion is to pull up the completeCompoundConfigs () method to the AbstractInterfaceConfig class.”*

*(Describe the benefits to apply the recommended refactorings) “The benefit is the removal of the feature envy (this method has many calls to the superclass). What do you think?”*

2.4Make double-check on the text of the pull request description by another co-author to avoid a misunderstanding (this text may be according to the following guidelines [3][4]).

2.5 Submit the pull request (step 2.2)

2.6 Observe whether the pull request was accepted at least one time per day

2.7 If the pull request was accepted, then to interact with developers to understand better on the application of the incomplete composite refactoring

2.7.1 Analyze the developers’ answers of the accepted pull request

2.7.2 Ask for developers on incomplete composite refactoring

*- “Hi, I'm a Ph.D. student and I'm researching refactoring. I noticed that on commit x you extracted the Y class from package x to package y. Did you apply this refactoring for the purpose of purely improving the structure (remove the code smell x) or did you have another goal? Thanks for your attention. “*

2.8 If the pull request was not accepted, then send an email for the developer that was the author of the incomplete composite (step 1.1) to understand better their application.

2.8.1 Send email for the author of composite refactoring and the review of pull request according to the following template

*(Greet the developer)*

*“Hi [DeveloperName], how are you?”*

*(Introduce you)*

*My name is AuthorName, I'm a researcher from (...). I'm researching about refactoring. I would like to share a refactoring opportunity with you on her/his source code.*

*(Introduce the composite refactoring and code smell)*

*I noticed that on commit x you applied the refactorings X on the Y class. Also, I have found that this class had some improvements in the code structure. But, I observed that method C has a code smell called Feature Envy because this method can be more interested in class Z. This method has many calls to class Z.*

*(Introduce the pull request that was rejected)*

*I submitted a pull request suggesting to extract and move this method to the class Z. But this pull request was rejected (please see the pull request* [*link*](http://link)*) I would like to know why the pull request was rejected and if the refactorings that were applied to remove this code smell, the Feature Envy?*

*Thanks for your attention.*

*Best regards, AuthorName.*

2.9 Observe whether the email was answered at least one time per day

**References**

[1] Bibiano *et* al. "A Quantitative Study on Characteristics and Effect of Batch Refactoring on Code Smells." ESEM'19.

[2] Fowler, 1999, "Refactoring: Improving the Design of Existing Code".

[3] Silva *et* al. "Why We Refactor? Confessions of GitHub Contributors", FSE'16.

[4] Soares *et* al. "Rejection Factors of Pull Requests Filed by Core Team Developers in Software Projects with High Acceptance Rates", ICMLA'15.