# Bug fix labeling protocol

This labeling protocol is based on protocol from "[Which Refactoring Reduces Bug Rate?](https://www.cse.huji.ac.il/~feit/papers/Refactor19PROMISE.pdf)" ([GitHub repository](https://github.com/evidencebp/Which-Refactoring-Reduces-Bug-Rate)).

In the full classification of each commit, we mark whether it is corrective, adaptive, perfective, refactor and being in English.

A commit might be some of these.

However, the following protocol refers only to the classification for corrective.

**Task**

Our goal is to label git commit message and decide whether the commit contains a bug fix.

1. For each commit we should fill two fields:
   1. Is\_bug\_fix, taking the values, True, False and ? (as intermediate value)
   2. Justification which is the reason based on the text for the classification.

The labeling is based on the commit message for each commit. Ideally this will be enough for the classification. In case it is not enough, annotator can use the commit diff, the issue details or any other mean to determine the ground truth.

**Instructions**

1. Unless we find an indication to a fix, the default should be False.
2. In case of uncertainty, the commit should be labeled by the option you consider most likely. Start the justification with the word “Uncertain” and then describe the source of the uncertainty and the reasons to your decision.
3. In case that it is hard to decide based on the commit message alone, one should check the commit content in [git](https://github.com/search?q=&type=Commits). The concept that we label is “Is the commit a bug fix?” and not “Does the commit message indicates a fix”. Hence, if a fix indication is found in git, the commit should be labeled as True.
4. The “Justification” field should contain the reason to the labeling.
   1. In case of a specific common non fix action (e.g., refactor, performance improvement, adding a feature), the action should be named.
   2. In case of an other action, the justification should be “other action”
   3. In case of a fix, the justification should be support from the text (e.g., fixed, correct).
   4. In case of messages not in English, the comment should be “Not in English”.
   5. In case of English text but no a specific term indicating a bug, the comment should be “No syntactic evidence”. This cases are rare. Consider text like “The previous implementation of the multiplication function returned 5 for mult(2,2). Now it returns 4.” The commit should be labeled as True since it is a fix, though some methods cannot identify it.
5. Fixes that are not bug fixes (e.g., documentation, style) should be labeled as False.
6. Specifically, typos and spelling errors should be labeled as False.
7. Warning fix (e.g., pylint warning) are not considered to a bug fix and should be labeled as False.
8. Unless the change is aimed to solve a specific know problem, it is not a fix. For example, in “improving reliability”, one cannot point at the problem being fixed. These cases should be labeled as False.
9. In cases that the commit contains some modifications, if at least one of them is considered a bug fix, the commit should be labeled as True.
10. Fix to build/compile, are considered a bug fix and should be labeled as True.
11. Tests are considered to be a part of the system and its requirements. Therefore, a bug fix in the tests is a bug fix and should be labeled as True.
12. Merge is a technical instrument. Its usage alone is not an indication regarding the content or being a bug fix. A merge commit should be judged by its content.