Refactor labeling protocol

We follow the taxonomy of “[Characteristics of Application Software Maintenance](https://pdfs.semanticscholar.org/2821/ee262ee92a8c5c5bb90cda7d0702e9f61af5.pdf)”

We divide commits into

1. Corrective - fixes. For bug fix protocol see the CCP labeling protocol.
2. Adaptive - implementation of new features
3. Perfective - improvement of current system
   1. Refactor - a perfective action. **Code refactoring** is the process of restructuring existing computer code—changing the [*factoring*](https://en.wikipedia.org/wiki/Decomposition_(computer_science))—without changing its external behavior ([wikipedia](https://en.wikipedia.org/wiki/Code_refactoring))
4. Other - whatever not fall into the above. Should be a small category

Specific cases

1. Perfective and refactor
   1. Clean up is perfective and refactor.
   2. Moving code to a different file is perfective and refactor.
   3. Optimization/Performance improvement is perfective. If the same algorithm is kept, it is also a refactor.
   4. Unification is perfective and refactor.
   5. Re implementation is perfective and refactor
   6. Simplifying is perfective and refactor
   7. Renaming entities is perfective and refactor.
   8. Removing unused code is perfective and refactor.
   9. Reimplementation is perfective and refactor.
2. Perfective but not refactor
   1. Improving style is perfective yet not a refactor.
   2. Clean up not in code (e.g., documentation) is perfective but not a refactor
   3. Fix documentation, adding documentation - perfective but not refactor.
   4. Indentation, code style changes are perfective but not refactor
3. Corrective
   1. Fixing a typo is corrective but not a bug fix
4. Adaptive
   1. Debug message/ feature aimed not to the end user - adaptive. We consider the features for programmers as part of the features.
   2. Changing an interface is adaptive.
   3. Updating a library is adaptive
   4. Porting code is adaptive
5. Merge are judged by content. Lacking information, default is adaptive since the merge action itself adapt the software.
6. In case of a commit the has some categories (e.g., adaptive and perfective), we won’t consider the perfective since we aim for a model of high precision.