Protocol

Bruno

This document describes a protocol for evaluating if comments detected by an automatic procedure, currently a bag of words, are evidences of SATD - Self Admitting Technical Debt.

Technical Debt (TD) happens when a change in an artifact related to the development of a software is made in a way that is not definitive or in a way that is not the best possible given the current requirements.

A SATD happens when a developer states explicitly for the rest of the team that there is a Technical Debt. In this case, the Technical Debt Can be considered a conscious decision made by the whole team. In some cases it is rational to acquire a technical debt. For instance, the team could exchange a future burden of a late refactoring for a earlier release. The benefit in launching a release early could compensate the extra work.

One way to admit a SATD is writing a comment confessing the TD. The comment can be written by the developer that made the change in the code base that led to a TD or by a colleague of the same development team.

In this study, comments were extracted from 5 open source projects. With the help of a group of bag of words, some of them are potential evidences of SATDs. Three researches (Bruno, Prof. Earl and Prof. Marcio) must classify these potential SATD comments as one of the following categories:

- Certainly No
  - All the following sentences apply:
    - The researcher understands the context in which the comment is made.
    - The researcher was able to guess the requirements well enough.
    - The researcher is certain that the developer and the team did not postpone a definitive solution.
    - The solution referred by the comment complies with the current requirements.
    - The researcher thinks that the majority of the software developers will agree with him
  - The following sentence may apply:
    - The comment is not a not complaining about a TD or admitting one. It is an explanation about intentions behind the code
- Probably No
  - Some of the following sentences do not totally apply but most of them apply
    - The researcher understands the context in which the comment is made.
    - The researcher was able to guess the requirements well enough
    - The researcher is certain that the developer and the team did not postpone a definitive solution.
    - The solution referred by the comment complies with the current requirements.
    - The researcher thinks that the majority of the software developers will agree with him

  - The following sentence may apply too:
    - The comment is not a not complaining about a TD or admitting one. It is an explanation about intentions behind the code

- Unknown
  - Some of the following sentences totally apply:
    - The researcher do not understand the context in which the comment is made.
    - The researcher was not able to guess the requirements well enough
    - The researcher considers that if a group of developers was consulted there would be no clear consensus about the possible TD referred
- Probably Yes
  - Some of the following sentences do not totally apply but most of them apply
    - The researcher understands the context in which the comment is made.
    - The researcher was able to guess the requirements well enough
    - The researcher is certain that the developer and the team postponed a definitive solution.
    - The solution referred by the comment do not comply with the current requirements.
    - The researcher thinks that the majority of the software developers will agree with him
- Certainly Yes
  - All the following sentences apply:
    - The researcher understands the context in which the comment is made.
    - The researcher was able to guess the requirements well enough.
    - The researcher is certain that the developer and the team postponed a definitive solution.
    - The solution referred by the comment do not comply with the current requirements.
    - The researcher thinks that the majority of the software developers will agree with him