Research on Bug-fix Distance and Development Robustness

Group 04, CS212, LZU

1. Data Acquisition

We are interested in the relationship between bug-fix distance and development robustness. To find the distance, we need data of commits with fix tag and the commit history of the corresponding files and functions.

In terms of development robustness, our model will be based on the flowing hypothesis **(not the hypothesis to verify in the model)**: system development is more robust if its bug density is lower and new files and features are added more evenly over time.

We will not see much in Linux-stable as this is not where new features are actually developed. They are developed on the subsystems dedicated git tree and then during the merge window enter. Since we want to use the development speed of features, we will try and extract that from many subsystems at file level in Linux-next.

We have implemented four data extractor classes to extract the above indicators (The related files are in the repository https://github.com/neetneves/groupwork04).

(1) FixDistanceExtractor: get the bug-fix distance of the subsystem specified in the specified version range. Use regular expression to extract fix tags, commits, file names, and function names. Among them, the regex to find fix tags is "^\W+Fixes: [a-f0-9]{8,40} \(.*\)$", while others are not ambiguous.

(2) BugDensityExtractor: get the bug density of one file in the subsystem specified in the specified version range. The bug density is defined as the ratio of the number of commits with fix tag to the number of commits in total.

(3) FileAddExtractor: get the time difference of new files' creation in the subsystem specified. We get the history of files in the subsystem and then get the time of its first

commit. The get_timediff method returns the list of time-difference of creation of two next new files.

(4) FeatureAddExtractor: get the time difference of new features' addition in the subsystem specified in the specified version range. To get the commits adding features, we used key words "add feature" (maybe not so reasonable). And then we get the list of time-difference of addition of two next new features.

2. Data Preprocessing

3. EDA

Before modeling, we did EDA first. The result shows that data of the development speed of files and features is not useful at all. We randomly chose five subsystems to statistic on the development speed of their files and functions, and we find that many of them does not have new features or files added.

Files statistic: {'fs/afs/': nan, 'drivers/video/': nan, 'kernel/printk/': nan, 'net/nfc/': nan, 'tools/usb/': nan}

Features statistic: {'fs/afs/': nan, 'drivers/video/': 9000400.35483871, 'kernel/printk/': nan, 'net/nfc/': nan, 'tools/usb/': nan}

For features, this may be due to the wrong choice of our regular expression. But if we use "feature" instead of "add feature", we will get a lot of results, and the commit information shows that most of them are not adding new features. As for files, we look for the commit history of each file and the earliest commit. We found that many files were created at the same time very early. In the following time, few new files were created.

This may be because our knowledge is limited, but we have not found an effective solution. We believe that this is the case for system development itself, with very few

new features and new files added, and its uniformity is not statistically significant. Therefore, in this report, we only try to model on fix distance and bug density.

4. Assumption and Hypothesis

4.1 Assumption

If there are few changes happened between introduction of a bug and the fix of it, then it would mean that the developers/maintainers are good because they can write readable code and/or understand the code they are working on, thus there would be a good development.

4.2 Hypothesis

A system whose bugs are fixed in closer commit has a more robust development. The indicator of development robustness is bug density.

5. Model Development

6. Evaluation and Interpretation

7. Verification