Gavin Scott

March 11th, 2016

Will Code for Food

**Metric:** Abstractness vs. Instability

Note: Only non-UI, non-test packages were considered

**Data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | **Abstractness** | **Instability** | **Sum (A + I)** |
| controller.api\_interfaces | 0.0 | 0.625 | 0.625 |
| controller.authentication | 0.0 | 0.64 | 0.64 |
| controller.retrieval | 0.083 | 0.36 | 0.443 |
| model.common.common | 0.083 | 0.24 | 0.323 |
| model.common.common.sorting | 0.0 | 0.6 | 0.6 |
| model.common.common.users | 0.0 | 0.58 | 0.58 |
| model.common.form | 0.17 | 0.18 | 0.35 |
| model.common.messaging | 0.17 | 0.59 | 0.76 |
| model.getInvolved | 0.0 | 0.36 | 0.36 |
| model.resources | 0.0 | 0.30 | 0.30 |
| model.ridesharing | 0.0 | 0.61 | 0.61 |
| tasks | 0.5 | 0.53 | 1.03 |
|  |  |  |  |
| **Average** | **0.084** | **0.48** | **0.55** |

**Analysis:**

An in-depth explanation of abstractness, instability, and how the relationship between the two can be used as a code-quality measure can be found in my previous metrics posting.

Looking only at the averages of the three parameters (abstractness, instability, and the sum of the two), it does not look like much progress has been made since the midway point through the quarter. Indeed, after the first metrics posting we talked as a group about making a concerted effort to improve these results, but we were never able to get around to it because more important or pressing tasks always took priority. However, midway through the quarter the standard deviation was slightly higher; we had more modules with a sum of 1, or close to 1, but also more that were on the lower end. However, this change was very slight and should not be overemphasized, but it does indicate a slight improvement in our code quality in the last month or so. As the size of the modules increases a slight decrease in the balance of abstractness and instability can be hard to prevent, and our code shows that with the overall decrease in the number of modules with a sum close to the ideal (1). That being said, it is better to have consistent code quality, rather than modules that vary widely from very high quality to very low quality, and the small change in the standard deviation of these metrics indicates that we are moving away from this, if very slowly.

Next quarter I would like to make a conscious team effort to improve this metric, because I believe that it is one of the more powerful indicators of code maintainability. I expect that we will be able to find some time because the bulk of development has been finished, but I also know that we will be working hard to fix bugs before and during the beta release and improving other metrics such as code coverage.