Paper 1

1) **Title:** Test Routes: A Manually Curated Method Level Dataset for Test-to-Code Traceability

Authors:

1. András Kicsi
2. Laszol Vidacs
3. Tibor Gyimothy

Conference: MSR ’20, October 5–6, 2020, Seoul, Republic of Korea

2) **Introduction:** Guaranteeing programming quality is a fundamental subject with a tremendous measure of logical and modern foundation. One of its principle perspectives is programming trying which means to reveal the shortcomings that exist in the product. Broad testing is viewed as acceptable practice all through the world. On account of a bigger programming framework, faltering measures of tests can be made as a side-effect of the turn of events. It isn't uncommon for a framework to consolidate a huge number of tests.

Having countless tests, in any event, telling which part of code a test plans to assess can mean an extensive trouble. This issue, following tests to their units under test, is called test-to-code detectability. As a rule, it implies something other than considering the strategies an experiment calls, which can be plentiful. It needs to think about the goal of the creator of the test, to point at explicit pieces of the code that the creator intended to test. While great coding rehearses like after a naming shows can enable this procedure, to even these cannot take care of each issue. Thinking about a previously existing framework, reflectively implementing such shows can be verging on the unimaginable. In this way, programmed extraction strategies are important to reveal the recognizably joins.

Motivation:

```
publicvoidtestIsAfterNow(){as
    sertEquals(false,
        newDateTime(TEST_TIME_NOW − 1).isAfterNow());
        sertEquals(false,
        newDateTime(TEST_TIME_NOW).isAfterNow());ass
    ertEquals(true,

}
```

The method aims to provide a boolean value signifying whether the instance it belongs to resembles a date and time in milliseconds after our current date and time. To evaluate this, the test calls the method three times on instances set to be less, equal and greater than our current

date and time. This naturally increases through- out development, thus the real present moment is not appropriate for testing this method.

3) Research Methodology:

- Dataset

Structure

Our dataset distinguishes four different method level roles in the testing process. The test cases are the methods aiming to assess a specific part of the software. They often contain one or more assertion statements and make method calls to conduct the evaluation. The test cases often enlist the aid of helper methods for setup or modularity purposes. These methods are customarily well separated from the production part of the source code

Data

The presented dataset contains information on four systems. The number of methods in the dataset can be seen in Table 2 for each system, grouped by category. Some interesting observations can be made from this, for example, Joda-Time seems to use a large number of test helpers while Commons Lang uses very few and also appears to often test more than one method with a test case. The number of production context methods also differ seriously.

System	Test Methods	Focal Methods	Prod. Context Methods	Test Helpers
Commons Lang	50	89	91	11
Joda-Time	50	54	312	101
JFreeChart	70	79	430	58
Gson	50	55	102	30

Possible uses

The knowledge of real traceability links facilitates test-driven de- velopment and improves software evolution. It can enable seamless integration between continuous code changes and unit tests, and serve as an important source of system documentation

4) **Result:** Test-to-code discernibility gives the objective of experiments in the creation code. Cutting edge calculations are normally assessed utilizing a generally modest number of physically gathered connections. Likewise, without a doubt, not many of these connections are accessible, which hampers the near assessment of novel strategies. In this paper, we depicted a physically curated dataset, which contains follow capacity joins for 220 experiments at technique level granularity from 4 open-source Java programs. The dataset incorporates the recognizability interfaces as well as features the setting of the test and creation techniques. The entire dataset comprises of in excess of 2000 orders of strategies that were physically analyzed.