

## Summary of the Paper

### Goal:

Compile and analyze the **distinct revisions** of **core module**, reaching the **maximum compilation** and evaluating the **evolution** of the module.

### Data sources:

37838 distinct revisions of the core module of 68 open-source software system across Apache, Google and Netflix.

### Definitions:

**Impactful**<sup>[3.1]</sup>, **Broken**, **Breaker**, **Carrier**, **Fixer**, **Neutral** commits.<sup>[3.2]</sup>

### Approaches and Tools:

HistoryCompiler: work to reach maximum compilation.

Analysis Wrapper: interface.

Orchestrator: cloud management operations.

NoSQL Datastore: data storage.

ReporterParser: retrieve

GitAnalyzer, GitHubAnalyzer: retrieve meta data.

DataAnalyzer, Plotter: analyze data.

SonarQube & FindBugs.

Games-Howell Statistical Significance test.

Build tools like Ant, Maven and Gradle.

Naïve Bayes and Logistic Regression.

Jaccard Distance.

### Results:

1. High compilation rate reached.
2. Most *breaks* are caused by single author and fixed within a short time, although exceptions exist.
3. Uncompilability could be predicted based on commits meta data.
4. Efficiently evaluate the dataset of two groups of authors indicating significant differences between them, minimizing missing data.

## My ideas

1. Dive more into details of the figures such as in figure 7. For example, the meaning of two climaxes in figure 7 and the breaker line of Google in figure 5.
2. It seems it takes a long time to analyze the project in AWS. Better approaches, better algorithms or optimizations to shorten the analysis time for future use may be worthy to work on.
3. A further research into the pattern of commit messages, not only limited to predict compile error but also perhaps for bug detection as well as other SQ metrics. Information about other aspects other than SQ may also be revealed.
4. The time interval between two commits may vary between different authors even if the amount of changes is the same. For example, some authors may like to test more times locally before committing which may make the interval long while others may just not be on the project during the interval. This issue could be more serious with OSS which means the interval could be unstable itself and may need reconsideration.
5. Further grouping of authors may lead to more conclusions. Now as stated in the paper, the authors are divided into two groups depending on their email addresses, but more pattern of grouping may be inducted from analysis of their committing styles such as the intervals, the messages, etc..