# Large-scale, AST-based API-usage analysis of open-source Java projects

## Summary

This paper performed analysis on SourceForge java project to see the usage of API. The paper introduces the footprint of API to look at both the number of API calls and the number of distinct API methods used by a project. The result indicates that while the number of API's (seems to be libraries by my viewpoint) does not necessarily increase with project size, the number of distinct API methods do increase with project size. Other analysis of the results shows that only a small amount of API methods are covered by most projects. The 2 major threats worth mentioning are use of single AST-based method and using SourceForge as only source.

## Things I would like to see discussed

- Construction of the corpus. The paper only mentioned getting Java like projects from SourceForge, however, no detailed information on what is gathered and how it was gathered. It just seems to me like random projects from SourceForge.

- The result seems to show that the number of distinct API methods increase as the scale of the project increase. I wonder if that's always the case, since if this follows when the scale is really large theoretically it would included every known API and would not increase afterwards.

- Why does build and unbuild really matter? The only reason given was buildable software is supposed to be well written mentioned in the reference group.