NC STATE UNIVERSITY

The Effect of Code Dependencies on Software Project Quality

Abdulrahim Sheikhnureldin
abduls@gwu.edu

Goal

The aim of this research is to develop a better understanding of the effect of code dependencies, both social (commit counts) and technical (imports), on the overall quality of a software project. Can the number of dependencies alone predict the outcome of a commit (improvement, bug, etc.)? If not, what are the implications for software developers and business personnel?

Why is this Important?

- To software developers:
  - ○ Can steps be taken early on in the development lifecycle to prevent bugs?
- To businesses:
  - ○ Should resources be focused on projects that are more likely to yield bugs, or improvements?

Background and Methods

The data:

- 40 open source Github projects provided by IBM with data on:
  - Commits
  - Issues
  - Users
- 14 of these projects included:
  - Dependency data per class, for each commit ID
  - Related this back to issue data using commit ID

Quality: How were Issues Resolved?

Feature: issue_resolved_type:

- Improvement
- Bug
- New Feature
- Task
- Sub-Task

Other Features:

- issue_priority
- commit_count
- issue_status
- num_dependencies

Related Work:

- Holz, et. al [1] found that knowing which components (imports) will be in a Java software project can serve as good predictors for project complexity, size, and quality.
- de Souza et. al. [2] discuss the construction of social and technical dependencies
- Schroter [3] discusses the possibility of predicting software defects and changes using import relations

Analysis and Results

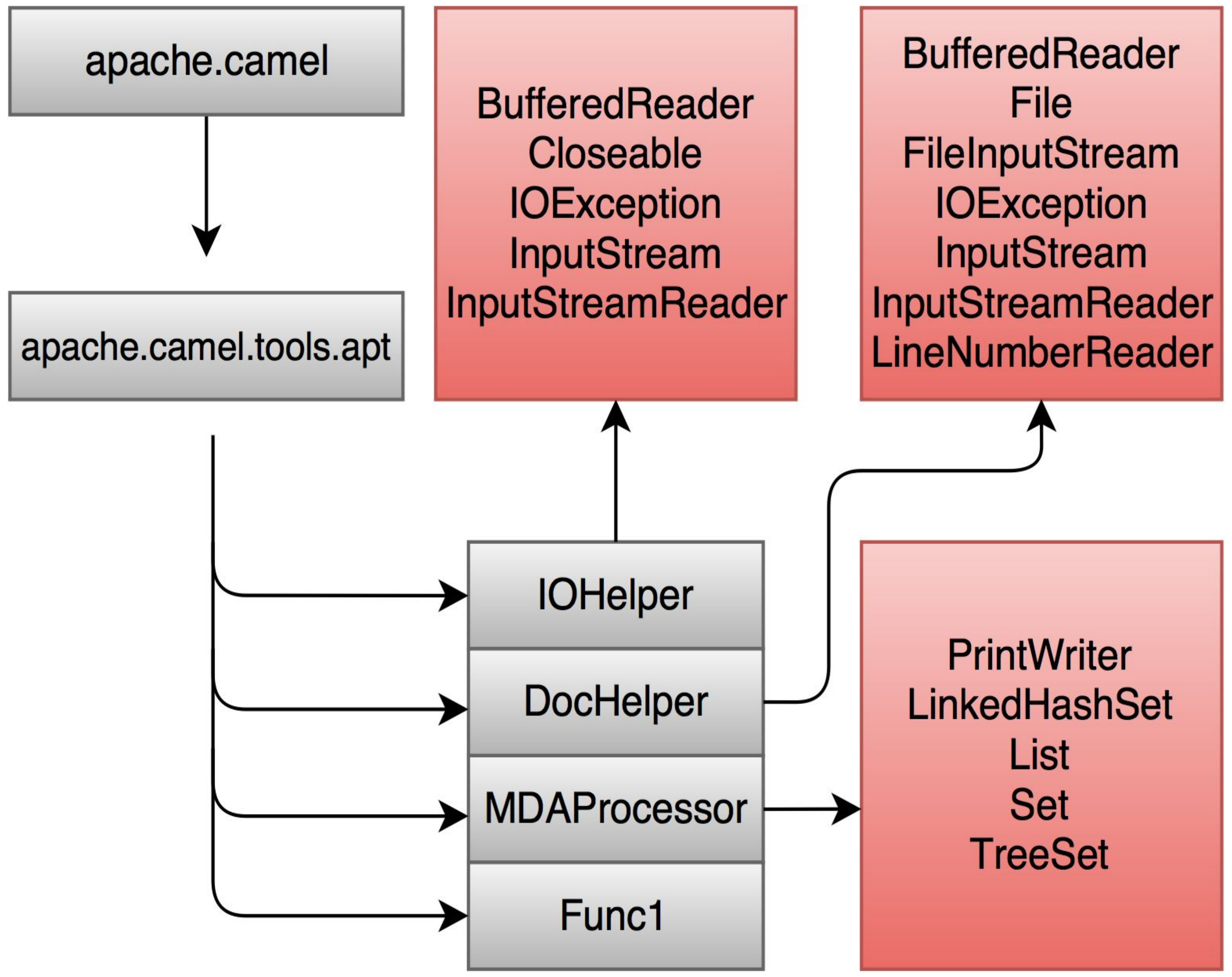

Figure 1: Java Project Call Graph - Packages, Classes, and Correlated Dependencies

Issue Resolved Type	Technical Dependencies F-Measure	Social Dependencies F-Measure
Improvement	0.425	0.787
Bug	0.407	0.746
Task	0.294	0.786
Sub-Task	0.265	0.747
Weighted Avg.	**0.368**	**0.767**

Figure 2: Random Forest Model Output - Technical Dependencies vs. Social Dependencies

Findings and Implications

No relationship between technical dependencies and software quality. However, social dependencies perform slightly better than technical dependencies. This may indicate that factors such as communication and issue prioritization between software developers, managers, and others are just as important as the technical aspects of the software project.

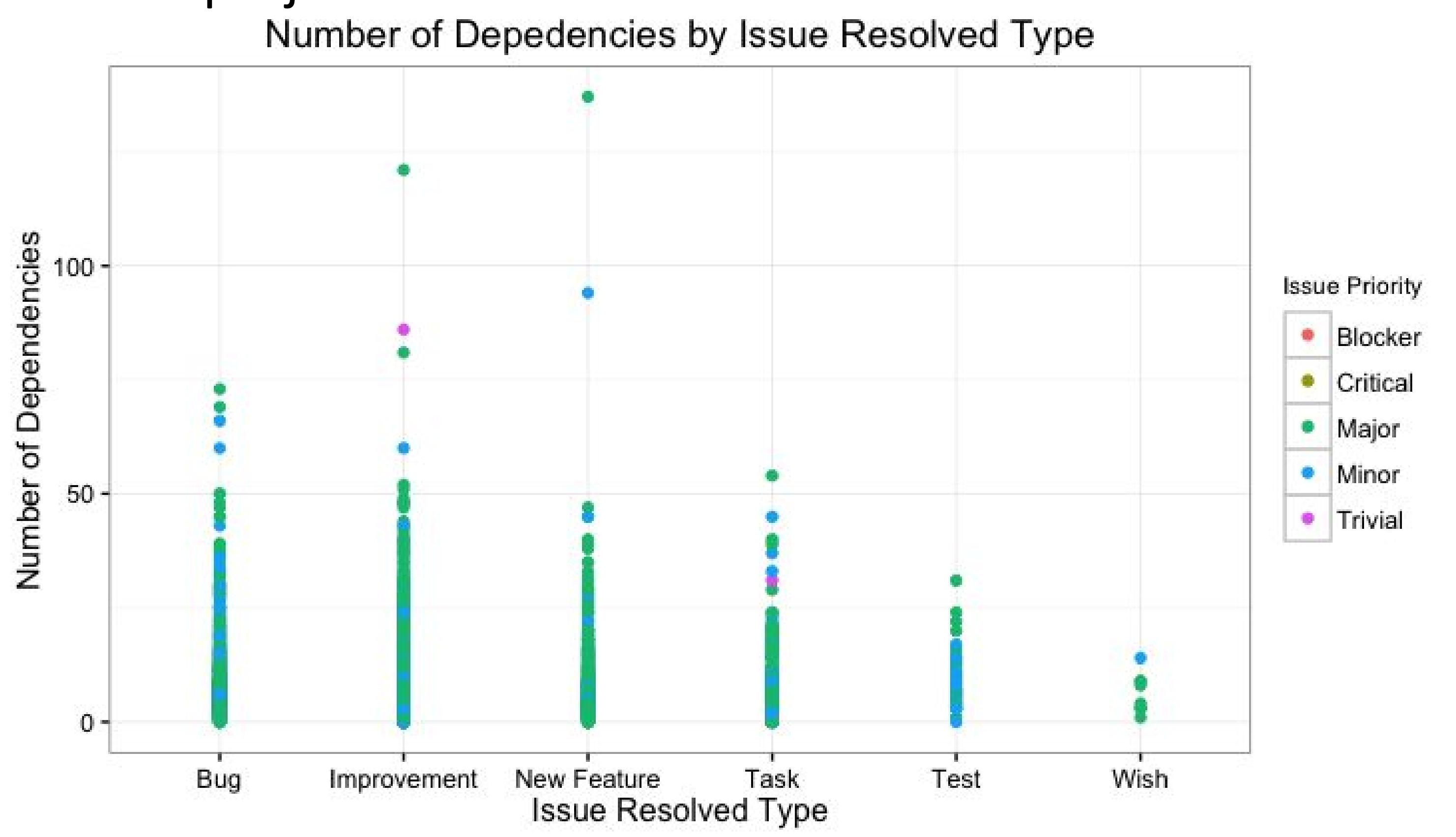

Figure 3: Sample Project - Dependency Frequency by Issue Resolved Type

Threads to Validity

- Small sample size of only 14 projects with dependency data
- Raw number of dependencies may overlook the importance of certain dependencies over others

Future Work

- Identify more features to improve predictions
- Predict project quality by specific dependency rather than raw number of dependencies
- Establish a relationship (if any) between technical dependencies and social dependencies

Notes

References

[1] Holz, Wolfgang, Rahul Premraj, Thomas Zimmermann, and Andreas Zeller. "Predicting Software Metrics at Design Time." *Product-Focused Software Process Improvement Lecture Notes in Computer Science* (2008): 34-44.

[2] Trainer, Erik, Stephen Quirk, Cleidson De Souza, and David Redmiles. "Bridging the Gap between Technical and Social Dependencies with Ariadne." *Proceedings of the 2005 OOPSLA Workshop on Eclipse Technology EXchange - Eclipse '05* (2005)

[3] Schroter, Adrian. "Predicting Defects and Changes with Import Relations." *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)* (2007):

Acknowledgements

[1] Research mentors: Dr. Tim Menzies and Dr. Christopher Parnin

[2] This work was supported in part by funding from the NSF, and the DevOps Analytics team in the Cloud Division of IBM, Research Triangle Park, NC.