Comparing Emperical Methods

ECE1785: Assignment 1 - Comparison of Methods

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Wang Yilong
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—

I. INTRODUCTION

II. HOW TO BREAK AN API

The first paper we will consult is "How to Break an API" [1], a 2016 multiple case study of three ecosystems philosophies toward change.

A. Method

A multiple case study, interviewing 28 developers in each of three ecosystems. Developers were recruited with recent, relevant experiences in similar roles contributing to packages with multiple upstream and downstream dependencies. The three ecosystems studied are Eclipse, R/CRAN, and Node.js/npm. Semi-structured interviews focused on personal practices and experiences negotiating upstream and downstream dependencies. Interviews were recorded, transcribed, then coded. Findings were validated with Dagenais and Robillard's [2] methodology to check fit and applicability as defined by Corbin and Strauss [3].

B. Research Questions

- How do developers make decisions about whether and when to perform breaking changes?
- How do they mitigate or delay costs for other developers?
- How do developers react and manage change in their dependencies?
- How do policies, tooling, and community values influence decision-making?

C. Evidence used to Reach Conclusion

III. NEED FOR TWEET

The second paper we studied is "Need for tweet: How open source developers talk about their github work on twitter" [4].

A. Method

In the second paper, the researchers proposed a heuristic computational method to link software developers' Github account to their twitter accounts. Then they

B. Research Question

- How to link developers cross-platform?
- How is the tweet content correlate with the role of the user on the project

C. Evidence used to reach conclusion

IV. DIFFERENCES IN METHODS

V. DIFFERENCES BETWEEN QUALITATIVE AND QUANTITATIVE

VI. WHEN EACH METHOD IS APPROPRIATE

A. Qualitative

B. Quantitative

VII. WEAKNESSES OF EACH METHOD

A. Qualitative

B. Quantitative

VIII. CONCLUSION

REFERENCES

[1] C. Bogart, C. Kästner, J. Herbsleb, and F. Thung, "How to break an api: cost negotiation and community values in three software ecosystems," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016, pp. 109–120.

[2] B. Dagenais and M. P. Robillard, "Creating and evolving developer documentation: understanding the decisions of open source contributors," in *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*, 2010, pp. 127–136.

[3] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage publications, 2014.

[4] H. Fang, D. Klug, H. Lamba, J. Herbsleb, and B. Vasilescu, "Need for tweet: How open source developers talk about their github work on twitter," in *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, pp. 322–326.