January 16, 2016

1. PROBLEM STATEMENT

Problem 1: Proactively Recommending Code Examples to Reduce Integration Conflicts

Developers frequently use source code examples as the basis for interacting with an application programming interface (API) to obtain the needed functionality [3]. In this case, the source code example serves as the explicit origin for a reuse task which transforms a sequence of APIs with corresponding control structure to the target context [1]. Existing code search engines always return multiple examples that are found based on the relevance to the user query rather than the ability to fit for the target context. For any selected example with a number of structural dependencies, developers have to rely on their intuition to determine which related elements should be transformed to the target context .

we propose an approach to recommend code examples that implement a queried feature based on their integration cost to the target context. Instead of using the entire code example, we extract salient program elements that might provide the main queried functionality, integrate these salient elements to the user-defined location, and proactively add related elements into the context with user interaction to resolve integration conflicts.

Identify Duplicated Feature Requests

There are a number of tracking system that helps the development process, such as Bugzilla, JIRA, and Pull Request mechanism in Github. Our preliminary study shows that only X% of all issues of Z Apache projects reported in JIRA are unique, and Y% of all pull requests are unique within 22 projects that actively use pull requests reported by [2]. Our tool can also be used to recommend a potential fix based on reported bug report or description in natural language.

We first extract four types of information from these issue descriptions and corresponding comments: issue description in natural language, execution trace/bug report, code snippet, and diff patch.

We evaluate our approach using the dataset from JIRA issue tracking system, and pull request in Github.

2. REFERENCES

[1] W. B. Frakes and K. Kang. Software reuse research: Status and future. *IEEE Trans. Software Eng.*, 31(7):529–536, 2005.

[2] V. Hellendoorn, P. T. Devanbu, and A. Bacchelli. Will they like this? evaluating code contributions with language models. In *12th IEEE/ACM Working Conference on Mining Software Repositories, MSR 2015, Florence, Italy, May 16-17, 2015*, pages 157–167, 2015.

[3] C. Sadowski, K. T. Stolee, and S. G. Elbaum. How developers search for code: a case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, Bergamo, Italy, August 30 - September 4, 2015*, pages 191–201, 2015.