

tool-recommender-bot

Chris Brown and Emerson Murphy-Hill

Department of Computer Science

North Carolina State University

Raleigh, NC

Email: dcbrow10@ncsu.edu, emerson@csc.ncsu.edu

Abstract—Recommendation systems were developed to improve the adoption of useful software tools and features designed to save time and effort in completing tasks that are often ignored by users. Previous research suggests that peer-to-peer recommendations are the most effective mode of tool discovery and that the receptiveness of recommendees is the most important characteristic in determining the outcome of tool recommendations. To help increase awareness of useful tools, we developed and evaluated a new recommendation system tool-recommender-bot designed to integrate aspects of peer interactions and user receptivity into automated tool suggestions for software developers of real-world applications. Our findings suggest that tool-recommender-bot is awesome, cool, and very effective in improving tool discovery.

Index Terms—Software Engineering; Tool Recommendation; Tool Discovery; Open Source

I. INTRODUCTION

Tool discovery is a problem...

Automated recommendation systems can help solve this problem...

But existing recommendations systems are ineffective...

Peer interactions and receptiveness are effective [1]...

We created tool-recommender-bot to solve this...

RQ1: How often can we expect tool-recommender-bot to make recommendations?

RQ2: How useful are recommendations from tool-recommender-bot to developers?

To answer these questions, we conducted a study analyzing tool-recommender-bot on five? popular open source Java projects to observe how many tool suggestions would be made based on past changes to the code base and how software developers reacted to receiving recommendations. This research makes the following contributions:

- We introduce the design and implementation of a novel automated recommendation system tool-recommender-bot
- We provide implications for future

II. RELATED WORK

Improving tool discovery...

Existing automated tool recommendation systems...

III. TOOL

tool-recommender-bot is awesome. Here's how...

A. Implementation

tool-recommender-bot technical details...

1) *Jenkins*:

2) *Maven*:

3) *Error Prone*:

B. Receptiveness

tool-recommender-bot was designed to integrate characteristics of peer interactions into automated recommendations. To better understand what makes peer-to-peer recommendations an effective mode of tool discovery, we observed how colleagues recommend tools to each other while completing tasks in a previous study. Our results found that the receptiveness of users was the only significant indicator of determining whether a tool recommendation was effective or not. Fogg defines receptiveness with to criteria, *demonstrating desire* and *familiarity* with the technology [2].

a) *Desire*:

b) *Familiarity*:

IV. METHODOLOGY

A. Projects

Trending open source java projects on Github that build with maven used for evaluation...

B. Study Design

We divided our study into two segments to address each research question:

1) *RQ1*: Last 100 pull requests on repositories...

2) *RQ2*: Followed up with pull request authors to gather data on recommendation...

V. RESULTS

A. How often can we expect tool-recommender-bot to make recommendations?

Tons of recommendations...

No false positives...

B. How useful are recommendations from tool-recommender-bot to developers?

Excellent responses from recommendees...

Statistically significant data...

VI. DISCUSSION

A. Implications

Here's what our results say about ways to improve tool recommendation systems...

B. Threats to Validity

Internal:

External: Java, open vs. closed source, Error Prone,

C. Future Work

More tools to recommend (static analysis, security, etc.)

More programming languages instead of just java...

More build systems (ant, gradle, TravisCI, bazel)...

VII. CONCLUSION

REFERENCES

- [1] C. Brown, J. Middleton, E. Sharma, and E. Murphy-Hill. How software users recommend tools to each other. In *Visual Languages and Human-Centric Computing*, 2017.
- [2] B. Fogg. Creating persuasive technologies: An eight-step design process. In *Proceedings of the 4th International Conference on Persuasive Technology*, Persuasive '09, pages 44:1–44:6, New York, NY, USA, 2009. ACM.