PeerJ Preprints NOT PEER-REVIEWED

# Crowdsourcing the Discovery of Software Repositories in an Educational Environment

Yuxing Ma
University of Tennessee
Knoxville, Tennessee
yma28@vols.utk.edu

Tapajit Dey
University of Tennessee
Knoxville, Tennessee
tdey2@vols.utk.edu

Jared M. Smith
University of Tennessee
Knoxville, Tennessee
jms@vols.utk.edu

Nathan Wilder
University of Tennessee
Knoxville, Tennessee
nwilder@vols.utk.edu

Audris Mockus
University of Tennessee
Knoxville, Tennessee
audris@utk.edu

## ABSTRACT

In software repository mining, it's important to have a broad representation of projects. In particular, it may be of interest to know what proportion of projects are public. Discovering public projects can be easily parallelized but not so easy to automate due to a variety of data sources. We evaluate the research and educational potential of crowdsourcing such research activity in an educational setting. Students were instructed on three ways of discovering the projects and assigned a task to discover the list of public projects from top 45 forges with each student assigned to one forge. Students had to discover as many of the projects as they could using the method of their choice and provide a market-research report for a fictional customer based on the attribute they selected. A subset of the results was sampled and verified for accuracy. We found that many of the public forges do not host public projects, that a substantial fraction of forges do not provide APIs and the APIs vary dramatically among the remaining forges. Some forges have been discontinued and others renamed, making the discovery task into an archaeological exercise. The students' findings raise a number of new research questions and demonstrate the teaching potential of the approach. The accuracy of the results obtained, however, was low, suggesting that crowdsourcing would require at least two or more likely a larger number of investigators per forge or a better way to gauge investigator skill. We expect that these lessons will be helpful in creating education-sourcing efforts in software data discovery.

## Keywords

Discovery; Educational Research; Crowdsourcing

## 1. INTRODUCTION

It is common to host software development tools as a service and a large number of repositories are being hosted on online forges. For example, the number and size of repositories hosted on GitHub roughly doubled over the last year. A census of public software projects is highly desirable for reasons similar to those of a population census [6]. Discovering the projects even on a known forge, may be a nontrivial task, however. To accomplish it, we ran an experiment to distribute the data gathering over a group of students in order to provide a practical example of data discovery. Such approach is promising because it has the potential for faster speed, lower cost, more flexibility, and better diversity, while at the same time providing important educational experience. This paper describes our efforts in this education-sourcing of the discovery activity. The outcomes will help to assess the accuracy of this approach from a research perspective and its educational value, which is expected to include benefits from team collaboration as well as learning and practicing novel skills in the discipline of data science.

## 2. RELATED WORK

Crowdsourcing is a concept coined by Jeff Howe in 2006[4]. It refers to outsourcing assignments to an undefined network of people instead of the traditional way of engaging a group of employees to solve a problem, which is considered a distributed problem-solving model. Crowdsourcing has drawn great attention since its conception. Users' motivations to participate have been studied [3] and crowdsourced data has been extensively analyzed for accuracy [2]. Many research works have been conducted in several fields using this model. For example, crowdsoucing has been used in relevance evaluation of information retrieval systems [8] and in facilitating the discovery of new medicines [7]. In general, crowdsourcing provides a quick and effective solution to aggregating efforts towards completion of a significant task across many individuals.

## 3. BACKGROUND

The basic goals for the discovery of public projects on a forge include, for example, a better understanding of how software forges are being used, how much information can be retrieve from various forges, and determining how many projects are public in each forge.

We are primarily concerned with the meta-questions: can the answers to the questions above be obtained via education-sourcing, are the answers accurate, and does education-sourcing

978-1-4503-2138-9.
10.1145/1235

provide good educational experience?

The setting for the experiment is detailed below and is geared to providing engaging and challenging group and individual activities.

## 3.1 Context

Students were told to imagine that they work for a cloud application provider, DevParadise, that needs to convince its investors that their business has a bright future. The students' task was to show that the number of customers of DevParadise is comparable to that of 45 competing forges.

Students were directed to use examples they gather to obtain an as complete as possible list of users and/or repositories hosted on their assigned, "competing" forges. While they were provided a template for the search strategy, they were free to use any reasonable means (including borrowing with attribution from the other teams) to get as many as they could within a very short but, unfortunately, very realistic time frame of two weeks.

Students were directed to keep in mind operational data pitfalls, such as any lack of context and missing or possibly incorrect data. Students were advised to notice that the cloud application provider may focus on specific types of customers (e.g. size, programming language, license, culture, enterprise versus public, etc.). Students were told to pick the desired characteristics of forge-targeted customers and estimate their percentage of the total number in the retrieved set. The students were also challenged to devise a way to estimate what fraction of each competitor's customers/repositories were in their respective retrieved set, and to try to assess if the distribution of different types of customers was the same in the retrieved and undiscovered sets of the site's data.

### *3.1.1 Techniques for Discovering Projects*

To retrieve data from 45 assigned forges, students primarily used one of three techniques that they were instructed to use and that are described below.

#### *Using the Search API.*

Although not all forges provide an API to retrieve data, some do. For example, GitHub and BitBucket have very similar and powerful REST APIs. Other forges have APIs that vary in their provided detail, usefulness, and accessibility. The APIs come with differing operational caveats. All or most APIs institute a rate limit on requests per hour. And public APIs will likely not include any information on private repositories. SourceForge APIs do not provide a way to list all hosted projects.

#### *Using Search Engines.*

If students don't want or are unable to use a site's search API (assuming it exists), they can use a search engine like Google, Bing, Baidu, or any others. For example, if students want to search repositories on BitBucket, they can send requests like "https://www.google.com/search?q=site%3Abitbucket.org" as repositories on BitBucket have the similar address format: http://*bitbucket.org* (* here represents any string, including an empty string). Students then go through all the pages returned to select links that appear to point to a unique repository. As with search APIs, the search engines also may have a rate limit on request frequency. Excessive requests from a single IP address will be forbidden if their frequency goes beyond that limit.

#### *Using Keyword Search.*

Some sites provide a keyword based search of public repositories. If no API is provided and results returned from search engines are poor, a keyword search may be able to be used. For example, to search projects in BitBucket that contain keyword 'test', they can send a request like "https://bitbucket.org/repo/all?name=test". However, this method seems to have a low guarantee of completeness.

# 4. EXPERIMENTAL CONTEXT

We will now describe the context in which this experiment in educational crowdsourcing was performed.

## 4.1 Course Description

The discovery project was a part of a class entitled the Fundamentals of Digital Archaeology. It included a mix of upperclassmen and graduate students. Some knowledge of programming was expected from students taking the course. For this project, students were divided into teams and given two weeks to finish their discovery tasks.

## 4.2 Students' Background

The course had 38 students in total. 28 of them majored in computer science, 9 students came from the business analytics program, 1 student was from nuclear engineering. Among computer science students, 7 students were graduates, 16 students were seniors and 5 were juniors. All business students were enrolled in the masters program in business analytics.

## 4.3 Hosts Explored

We retrieved a list of the source code repository hosts provided on a popular blog post [1]. The list contains open source hosts including GitHub, Bitbucket, Launchpad, Sourceforge, Google Code, and Eclipse. Any of the forges may use different version control systems (VCS), host specific types of repositories, and vary in other ways.

## 4.4 Division of Tasks

The students were divided into eight groups, and forges were evenly (and randomly) dispersed among groups. The students were free to decide how to divide work within a group and were able to collaborate on each forge or assign each forge to a specific team member. However, performance was judged upon the results and reports provided by the team as a whole.

## 4.5 Task Duration and Recording

We created eight private repositories on GitHub, and each team used its own repository to accumulate their analysis and results. Team members were able to use the repository to commit code, share ideas, and schedule assignments among members. They were also able to use other tools to communicate and cooperate together. The use of Python and R for analysis was encouraged, as both languages are well suited to the data discovery and analysis domains.

We provided students with individual docker containers that had necessary data science and web retrieval tools. These tools included IPython notebooks for doing interactive analysis. We also gave the students access to an un-

**Table 1: Forges with No Public Repository**

| Forge Name | Forge URL | API |
|---|---|---|
| ProjectLocker | projectlocker.com | REST API |
| Freepository | freepository.com | None |
| Codebase | codebasehq.com | None |
| Unfuddle | unfuddle.com | Domain specific [1] |
| Beanstalk | beanstalkapp.com | REST API |
| SourceRepo | sourcerepo.com | None |
| Review Board | reviewboard.org | REST API |
| Deveo | deveo.com | REST API |
| SSH Control | sshcontrol.com | None |
| versionshelf | versionshelf.com | None |
| Pulp | pulpproject.org | None [2] |

restricted Amazon Web Services account that would allow them to spin up new servers with new IP addresses in order to partially work around heavy rate limiting. We used a class-accessible MongoDB database running locally at the university to store retrieved data. Each team was able to create separate collections in MongoDB to store their retrieved data for later work with Python or R. We planned to create a separate database for each of the 45 collections. MongoDB was chosen because it provides APIs in several languages that are simple for a novice programmer to use, the data type was most likely to be text, and there was no need to do complex relational operations among the data.

We also provided a single Google Sheet for all of the teams to enter a summary of their analysis and results. It had the columns listed in Table 2 and also the URL of the API if one was provided, an indication whether any public projects were hosted and the relevancy of those projects to DevParadise, and additional important information including rate limits, page length, techniques used to collect data, etc. The final column was to indicate whether the data was complete or not and the existence of any guarantees of the completeness of the data (e.g. the existence of an API may say something about the completeness and consistency of the data available).

# 5. FINDINGS

Per our research goals we describe the direct discoveries of the education-sourcing effort, the experiences encountered as students were engaged in these discovery tasks, the accuracy evaluation, and other indirect results.

## 5.1 Basic information on the discovered projects

The 45 forges tested can broadly be grouped into forges that only have private repositories and forges that have public (and, in most cases, private) repositories. Many of the forges with no public projects have highly functional APIs for their customers. In some cases a limited number of repositories were provided for free, so some students created their own test repository and were able to test the API, thus providing educational value.

Table 1 lists the forges that we found to host only private repositories.

Since API use was not a task requirement, in some cases students used Google or Bing search to collect data even if an API was available. This, in conjunction with a strict time limit and limited experience in discovery tasks, often resulted in only a fraction of the entire set of projects being retrieved.

Some of the listed forges have ceased operation: Google Code, Gitorious, BetterCodes, and RhodeCode. Students were able to find archived data from Google Code (on the original site) and Gitorious (in a web archive). The two largest forges GitHub and BitBucket were not assigned in discovery tasks because they were used as examples to instruct students on how to conduct the discovery tasks and on what results could be obtained.

Some of the smaller forges that hosted exclusively public repositories encouraged only repositories related to a larger open-source project. One example was JoomlaCode, which hosted tools and extensions to the Joomla! web framework. Another example was Eclipse; which hosted extensions to the Eclipse IDE, complementary programs, or other projects of interest to the larger Eclipse community.

While some forges, such as Unfuddle, seemed to explicitly disallow public repositories altogether; several with no free hosting option apparently effectively discouraged public repositories by using a cost model that attracts closed-source software. Setting the default repository access to private may have also contributed. CloudForge requires that the public-private access setting apply to one's entire account. This would necessitate purchasing two accounts in order to use it for both proprietary and public projects.

Out of 45 forges, only 14 provided an API to retrieve data, but not all APIs supported public search functions. Some APIs were for interacting with one's own repositories only. REST APIs were the most common type of API.

## 5.2 Findings on the crowd-sourcing activity

For some forges, data cleaning is needed before trying to use an API. For example, a google search for "stash/repos?visibility=public" results in many sites that are not Stash servers. It is necessary to determine which of these sites are stash servers, and only then the Stash API can be used to identify repositories.

For Transifex and CodePlex, the search for repositories on the site was more effective if done using the forges' built-in search. Specifically for Transifex, the students wrote a program to try random search terms and pull links to repositories that were found. In total, on Transifex roughly 2,600 repositories were found by using the built in website search method, and the remaining 2,800 were found using Google search.

Students also gathered information on the characteristics of individual repositories in the forge. From this analysis, we were able to gather information including license usage, categories of projects, numbers of repository members, and downloads. For CodePlex, the license distribution and downloads per license reveals that although the Apache license was the most used on the forge (with over 60,000 uses out of the total 107,000 repositories found), repositories with the Apache license did not have nearly as many downloads (roughly 75,000 downloads) as repositories with the GPLv2 license (roughly 375,000 downloads across only 7,500 repositories out of the total 107,000 repositories). Conversely, on

[1] http://{subdomain.}unfuddle.com/api/v1/...

[2] This was found to be wrong by a spot check later. "Pulp has a well-documented REST API and command line interface for management."- https://pulp.readthedocs.org/en/latest/user-guide/introduction.html

PeerJ Preprints NOT PEER-REVIEWED

Table 2: Active Forges with Public Repositories

| Forge Name | Forge URL | API | Search Method | Repositories Retrieved |
|---|---|---|---|---|
| CloudForge | cloudforge.com | Private API | Google search | 42 |
| SourceForge | sourceforge.net | REST API | Google search | 48,000 - 50,000 |
| launchpad | launchpad.net | API | API | 36,860 |
| Assembla | assembla.com/home | No | Google search | about 70,000 |
| CodePlex | codeplex.com | REST API | Google search | 100,000 |
| Savannah | savannah.gnu.org | No | Google search | 3613 |
| CCPForge | ccpforge.cse.rl.ac.uk/gf | No | Google search | 126 |
| Jenkins | ci.jenkins-ci.org | REST API | API | 106,336 |
| Repository Hosting | Respositoryhosting.com | No | Google search | <88 |
| KForge | pythonhosted.org/kforge | API | python.org search | 81,000 |
| Phabricator | phabricator.org | Conduit API | API | about 10,000 |
| Fedorahosted | fedorahosted.org/web | No | Google search | 914 |
| JavaForge | javaforge.com | No | Google search | 7672 |
| Kiln | fogcreek.com/kiln | No | Google search | 43 |
| SVNRepository | SVNRepository.com | No | Google search | 15 |
| Pikacode | pikacode.com | No | Google search | 2 |
| Planio | plan.io | No | Google search | 26 |
| GNA! | gna.org | No | Google search | 1326 |
| JoomlaCode | joomlacode.org/gf | REST API | Google search | 971 |
| tuxfamily | tuxfamily.org | No | Google search | 209 |
| pastebin | pastebin.com | No | Google search | about 1800 |
| GitLab | gitlab.com | No | Google search | about 57,000 |
| Eclipse | eclipse.org/home/index.php | No | Google search | 214 |
| Turnkey GNU | turnkeylinux.org/all | No | Google search | 100 |
| JavaNet | home.java.net/projects/alpha | No | Google search | 1583 |
| Stash | atlassian.com/software/bitbucket/server | REST API | API | 5400 |
| Transifex | transifex.com | No | Google search | 5400 |
| Tigris | tigris.org | No | Google search | 678 |

CodePlex, the least downloaded and the least used license was the EPL license.

Another characteristic of repositories investigated were the categories and number of members per repository, including those members with commit access. For JavaForge, of 7,672 projects gathered, there was an average of 11.79 members per project. The most popular category by far on JavaForge was "Communications" with around 35% of all total category counts.

These finding clearly demonstrate the instructional value of such an education-sourcing approach with students discovering innovative ways to conduct the task and to answer a large number of interesting and relevant questions.

## 5.3 Accuracy of the Results

We sampled five projects to verify accuracy of the findings. In particular, it was possible to find 452,046 projects via Google Code's search functionality, yet the educational sourcing produced only 206,000[5]. Furthermore, SourceForge educational sourcing produced only 48,000 projects, while even in 2009 the number of projects on SourceForge exceeded 100K[5]. The students found 3,600 projects on Savannah while in Jan 2016 the number of projects reached 3707 [3]. The number of projects on Launchpad were 37,834 in Jan 2016[4], and the educational sourcing produced 36,860 projects in October 2015.

[3]http://savannah.gnu.org/
[4]https://launchpad.net/

# 6. CONCLUSIONS

The reported results of software project discovery activity using education-sourcing are likely to be useful for future mining, analysis, and development endeavors, especially those involving public and private software repository hosting services or other data sources that require heterogeneous discovery methods. For example, we discovered that many public forges don't host public projects, a substantial fraction of forges don't provide APIs, and API types vary greatly among remaining forges. Additionally, some forges are defunct and others renamed. When verifying the education-sourced data, we also found that the accuracy may not always be high. Some students failed to find the APIs provided by forges, and some were unable to use these APIs, which demonstrates that there may be a need to assign more independent investigators to each forge in order to assess individual investigator skill. The educational value, however, was significant with many interesting qualitative and quantitative observations that were produced and that had relevance to the target objective.

# 7. ACKNOWLEDGMENTS

We would like to thank all the students who participated in the course and contributed their results to this work. We would also like to thank the University of Tennessee, Knoxville for facilitating courses where educational research can be conducted.

PeerJ Preprints NOT PEER-REVIEWED

## 8. REFERENCES


[1] Top source code repository hosts. https://blog.profitbricks.com/top-source-code-repository-hosts.

[2] G. Barbier, R. Zafarani, H. Gao, G. Fung, and H. Liu. Maximizing benefits from crowdsourced data. *Computational and Mathematical Organization Theory*, 18(3):257–279, 2012.

[3] M. Hossain. Crowdsourcing: Activities, incentives and users' motivations to participate. In *Internatonal Conference on Innovation, Management and Technology Research*, pages 501–506, May 2012.

[4] J. Howe. The rise of crowdsourcing. *Wired*, 14:1–5, June 2006.

[5] J. Howison, M. Conklin, and K. Crowston. Flossmole: A collaborative repository for floss research data and analyses. *International Journal of Information Technology and Web Engineering*, 1(3):17–26.

[6] A. Mockus. Amassing and indexing a large sample of version control systems: towards the census of public source code history. In *6th IEEE Working Conference on Mining Software Repositories*, May 16–17 2009.

[7] T. C. Norman, C. Bountra, A. M. Edwards, K. R. Yamamoto, and S. H. Friend. Leveraging crowdsourcing to facilitate the discovery of new medicines. *Science Translational Medicine*, 3(88):88mr1–88mr1, 2011.

[8] B. S. Omar Alonso, Daniel E. Rose. Crowdsourcing for relevance evaluation. *ACM SIGIR Forum*, 42(2):9–15, December 2008.