# Code Snippet Replication in GitHub from StackOverflow and ViceVersa: Empirical Study

Venigalla Akhila Sri Manasa
Indian Institute Technology
Tirupati, Andhra Pradesh, India
cs18m017@iittp.ac.in

## ABSTRACT

There are a lot of questions answered on Stackoverflow website. Most of these questions also include code snippets in the answers. These questions are also answered by developers, who post their projects on github. Many a times, it may happen that these developers answer questions on StackOverflow by posting a part of their code snippet. It may also happen that the code snippets on StackOverflow are replicated in few projects on github. This may sometime result in inconsistencies and bug propagation.In this Empirical Study, the idea is to find the extent to which code snippets are reused and how suggestible is it to reuse the code snippets.

## KEYWORDS

GitHub, Stack Overflow, code snippets, replications, GitHub REPOSITORY LINK : https://github.com/AkhilaSriManasa/EmpiricalStudy

## 1 LITERATURE REVIEW

While reviewing existing literature, the following publications have been found relevant.

*(1)What do developers search for in web?*

*(2)Why and How developers fork, what from whom in GitHub*

*(3)Some From Here, Some from There : Cross Project Code Reuse in GitHub*

*(4)Stack Overflow In GitHub - Any Snippets There?*

*(5)Bug Propagation through Code Cloning:Empirical Study*

The Empirical Study could use the results of First paper to identify what part of code is most frequently replicated, then understand forking patterns from results of Second paper and retrieve necessary data accordingly. Paper 3 and 4 could be utilized to identify the methods of analyzing code and also methods of data scraping. Finally the fifth paper could help in finding methods to identify Bugs and their propagation in the replicated code snippets.

The paper, What do developers search for on the web, is cited in ESE dblp 2017. In this paper, queries from 60 developers, 235 software engineers, from more than 21 countries, across 5 continents have been considered. It has been observed that most of the searches are based on third party code reuse and reusable code snippets. Data Collection tool, ActivitySpace had been installed into participants' computer, to track and collect behavior for two weeks. 34 search tasks have been identified and investigated. They were grouped into 7 dimensions and practitioners were asked to rate frequency and difficulty of these search tasks. It was observed that developers frequently use stack overflow to find desired results(As many as 63% of searches ended up landing on StackOverflow page.)

The paper, Why and How developers fork, what from whom in GitHub dealt with dataset containing 236,344 developers and 1,841,324 forks. This dataset is collected from GitHub. Programming languages and owners of the forked repositories were analysed. It was observed that Developers fork repositories to submit pull requests, fix bugs, add new features and Keep copies. Repositories are forked from various sources like search engines, external sites, Open Source sites, etc. It was observed that repositories written in developer's preferred programming language were more likely to be forked. A study states, 14% of repositories on GitHub adopt fork and pull model. It has been observed that 46% of forks are made to submit pull requests, 36% by developers to fix bugs, 21.8% forks are done to add new features and 8.9% are done to keep copies, just in case, parent repositories gets deleted.

In the paper Some From Here, Some from There : Cross Project Code Reuse in GitHub, a clone finding tool, Deckard was used. It identifies copies of code fragments across projects. A clone is a code snippet present somewhere that is similar to code elsewhere. 5753 java projects were selected form GitHub and analysed for clones. It has been found that cross-project clones range between 10% - 30% of all code clones and 5% of code base. Most of these clones are utility clones, i.e., the entire files, directories or projects are copied with minimal changes. It is also observed that code cloning follows onion model.i.e., most of the clones are from same project, then from projects in same application domain, lastly from projects in different domains. In Deckard tool, the clones and their accuracy is specified by three parameters - *Stride* , *Similarity* and *Token size.* In the paper, Stride is set to infinity and Similarity is set to 1, to find exact clones. Three different token sizes have been used - 20, 30, 50. Deckard works only for single project. Hence, all projects' repositories being considered were placed in an umbrella directory having one folder for one project, to facilitate identification of cross project clones. A clone is called a cross-project clone if, for a given cloneID, atleast two instances of this are obtained from 2 different projects. A co-clone graph, with links between projects that share code clones, a co-developer graph, that will link projects with developers in common among them, are constructed to assess overlap between cross-project cloning and shared people between projects. In this paper, cross-project cloning in GitHub has been studied and it has been observed that a prominent cross-project cloning is present in OSS code.

The publication Stack Overflow In GitHub - Any Snippets There?, studies how snippets of code in stack overflow are used in GitHub by programmers, in their projects.It studies whether stack overflow snippets could be found in realtime projects and if yes, does they suffer any adaptations, or if they are literally copied.It was observed that GitHub developers ask for help on StackOverflow to solve their own technical challenges. They also answer questions on StackOverflow to quench others' thirst for knowledge. GitHub developers may copy-paste StackOverflow code snippets in their project and may use their code in GitHub repositories to answer questions on Stack Overflow. Intra and Inter Code duplication analysis is performed on StackOverflow and GitHub. This paper has considered 909K python projects on GitHub that comprised of about 292M function definitions and 1.9M code snippets in StackOverflow. Blocks were extracted from GitHub and StackOverflow posts. From python files of GitHub projects, functions defined inside and outside classes have been extracted. Using markdown <code>...</code>, code snippets have been extracted from python posts. Tokenization is done on data extracted. Tokens in blocks were identified and their occurrences were counted. Also, facts about Block Hash, Token Hash, Lines Of Code , etc have been captured.Block Hash and Token Hash duplicates have calculated. Block Hash is the hash value of absolute block composition and Token Hash is hash value after elimination of spaces, tabs, comments, etc. Block hash of two blocks is equal implies that they are exact copies of each other. Token hash is equal implies they have idiosyncracies among them. In some snippets, there might be partial cloning, i.e., there might be additional variables or tracing and debugging statements in exact copies of blocks. These are detected by tool - SourcereCC. It was observed that Exact duplication among StackOverflow and GitHub exists, but is very rare(less than 1%). Token level duplication is more common(4M blocks in GitHub are similar to StackOverflow snippets).Near duplication shows (1.1% blocks are similar to StackOverflow and 2% of StackOverflow snippets are similar to those on GitHub). It was also observed that majority of duplication is only among blocks with minimal lines of code(2 lines of code)

The idea of the paper - Bug Propagation through code cloning-An Empirical Study is to study bug propagation intensity due to code cloning.Two patterns of clone evolution have been defined to indicate propagation of bugs due to code cloning, where, one pattern is observes percentage of clone fragments containing bugs propagated due to bug-fix changes and other pattern observes percentage of bug fixes in code clones that occur due to propagated bugs.Three types of clones have been considered, where Type1 refers to exact clones, Type2 and Type3 refer to near-miss clones.Exact clones are those code fragments that are exactly the same,omitting comments and indentations. Type2 clones are obtained from Type1 clones by renaming identifiers or modified data types. Type3 clones are obtained by addition or deletion of few lines in Type1 and Type2 clones.It has been observed that a particular code clone pair follows Similarity preserving co-change, where in the pair of clones preserve their similarity in spite of similarity changes in commit operation.It was observed that most bugs are propagated in Type2 and Type3 clones than in Type1.It has also been observed that 33% of code-clone containing propagated bugs due to bug-fix changes and 28.57% of the bug-fix changes in code clones occur to fix propagated bugs.

## 2 RESEARCH QUESTIONS ASKED

Research questions asked and answered in the above papers are as follows:

**RQ1:What do developers search for in web?**

It was observed that most of the searches are based on third party code reuse and reusable code snippets.

**RQ2:What are the webpages that are most frequently used by developers to solve their problems?**

It was observed that developers frequently use stack overflow to find desired results(As many as 63% of searches ended up landing on StackOverflow page.)

**RQ3:Why developers fork repositories?**

Developers fork repositories to submit pull requests, fix bugs, add new features and keep copies etc.

**RQ4:How do developers find repositories to fork?**

Developers find repositories to fork using various search engines, or by reading content on external sites (like Twitter), by social relationships (i.e.,considering repositories forked by developers they follow on GitHub), by noting GitHub recommendation (i.e., trending repositories) on its explore page, etc.

**RQ5:Do developers care about repository owners?**

It was observed that 35.2% of people who responded care about repository owners when repositories are forked.

**RQ6:What kind of repositories do developers fork?**

Developers mostly fork repositories from creators

**RQ7:Do developers prefer repositories written in particular programming language?**

Developers generally fork repositories that are written in programming languages that they are well versed with or that they prefer more.

**RQ8:What is the prevalence of within and across project cloning, and what kind of code is cloned?**

A significant amount of code is observed to be reused across GitHub projects, and the corresponding clones followed fixed patterns.

**RQ9:Is there an asymmetry among the projects, i.e., are there projects that serve as clone sources more than their fair share? Moreover, are some projects reusing other projects' code at a greater rate?**

Cross-system cloning is a directed and nonuniform phenomenon and most projects "obtain" more clones than "provide" them. There are also projects that serve as hubs or "super-sources" of clones to other projects.

**RQ10: Can we find support for existing mechanisms which promote cloning, such as multi-project developers or specific application domains?**

Cross-project cloning is more prevalent inside a domain boundary. Authors of the codes being cloned don't play a much important role in such cases.

**RQ11:Are there any stack overflow code snippets in real time projects that are on GitHub?** Yes, GitHub projects contain code snippets from StackOverflow.

**RQ12:What is the extent to which projects on GitHub have the same code snippets in Stackoverflow?**

It was observed that 405K(1.1%) blocks among the dataset are similar to code snippets on StackOverflow.

**RQ13:What is the reason behind code snippet replication?**

It was observed that GitHub developers ask for help on StackOverflow to solve their own technical challenges. They also answer questions on StackOverflow to quench others' thirst for knowledge. GitHub developers may copy-paste StackOverflow code snippets in their project and may use their code in GitHub repositories to answer questions on Stack Overflow.

**RQ14:What percentage of code clones in different clone-types can be involved with bug propagation?**

From the experimental results and analysis it can be stated that a considerable proportion of the clone fragments having bug- ïňĄxes be involved with bug-propagation. According to the subject systems, up to 33% of the bug- ïňĄx clones comprised of propagated bugs. Type 1 clones exhibit the lowest possibility of being involved with bug propagation. Bug propagation is mainly observed in Type 2 and Type 3 clones with Type 3 clones showing the highest intensity of propagation.

**RQ15:What percentage of the bugs that are experienced by different clone-types can be propagated bugs?**

According to the experimental results and analysis, it was observed that, a considerable proportion of the bugs in code clones can be propagated bugs. This percentage can be up to 28.57% according to the subject systems. The overall percentage of propagated bugs is the highest in Type 2 case, and the lowest in Type 1 case. However, this percentage regarding Type 3 case is also very near to that of Type 2 case.

**RQ16:Which pattern of bug-propagation is more intense during evolution?**

As per the observation, the first bug propagation pattern is more likely to occur when two clone fragments in clone pair were created in same revision in comparison to the second pattern in which two clone fragments are in two different revisions.

## 3 METHODS USED TO PERFORM STUDY

*Data Collection tool, ActivitySpace is used to track developers' search patterns.

*Certain number of selected developers were sent mails consisting of few questions and their responses have been analyzed to find why and how do developers fork, what from GitHub.

*Deckard tool was used find the copies of code fragments across projects.

*Code snippets and blocks extracted are hashed and compared to find similarity of code snippets across repositories and also among GitHub and StackOverflow.

****WEEK 2****

## 4 DATA SETS USED

The paper, "What do developers search for in web?", uses queries from 60 developers, 235 soft-ware engineers, from more than 21 countries, across 5 continents.

The paper, "Why and How developers fork, what from whom in GitHub" uses dataset containing 236,344 developers and 1,841,324 forks extracted from GitHub

The paper, "Some From Here, Some from There : Cross Project Code Reuse in GitHub", uses 5753 java projects extracted from GitHub for analysis.

The paper, "Stack Overflow In GitHub - Any Snippets There?", uses 909K python projects from GitHub, that comprised of 292M function definitions. It also extracted 1.9M code snippets from StackOverflow.

The paper, "Bug Propagation through Code Cloning:Empirical Study", considers four systems for experimentation and analysis. They are - "Carol", "Freecol", "jEdit", "Jabref".

## 5 UNSOLVED PROBLEMS IN THIS AREA

*Is it suggestible to replicate code snippets?

*What is the impact of code snippet replication on GitHub and StackOverflow websites?

*What are the issues faced in code snippet replication?

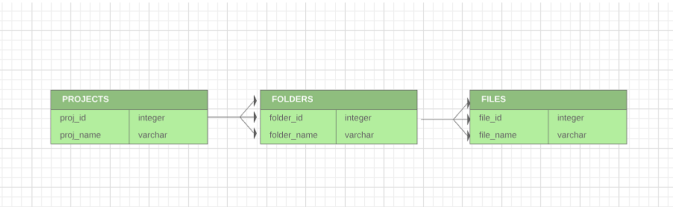
*Are bugs propagated in code snippet replication?

*How can bug propagation be detected in initial stages of project creation?

## 6 DATA SET DETERMINATION AND SCHEMA

Data set has been collected by examining 8 projects based on python on github. The source code of these projects have been collected and the same are reflected in the github repository.Data Scraping is done by considering the root folder and navigating down the root to find the python scripts. The database schema is given as in figure1.

**Figure 1: Database Schema**



## 7 METHOD USED TO CREATE DATA SET

First Eight projects based out of python in github are extracted and stored in a table on database.Each project's name is stored as a tuple. Each project is then drilled to next level to find the folders present in it. These are stored in separate tables, with table names pertaining to parent's name. They are further drilled down to get python scripts. File names that end with ".py" are considered and are stored in the database in the form of tables. Each ".py" file is opened and raw data is extracted and stored as ".py" files in their respective folders. These can be found on the github repo.

## 8 FUTURE TASKS TO BE DONE

Should compare the scripts downloaded, check for repeated code snippets

Should validate Database schema and organize database accordingly

Should identify code errors and check for propagated bugs.

## REFERENCES

[1] Manishankar Mondal, Chanchal K. Roy, Kevin A. Schneider. *Bug Propagation through Code Cloning: An Empirical Study.* ICSME 2017 dblp, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=a.

[2] Jing Jiang, David Lo, Jiahuan He1ÂůXin Xia, Pavneet Singh Kochhar, Li Zhang. *Why and how developers fork what from whom in GitHub.* ESE 2017 dblp, from https://link.springer.com/content/pdf/10.1007%2Fs10664-016-9436-6.pdf

[3] Xin Xia,Lingfeng Bao, David Lo, Pavneet Singh Kochhar, Ahmed E.Hassan, Zhenchang Xing *What do developers search for on the web?*. ESE 2017 dblp, from https://link.springer.com/article/10.1007%2Fs10664-017-9514-4

[4] Mohammad Gharehyazie, Baishakhi Ray, Vladimir Filkov *Some From Here, Some From There:Cross-Project Code Reuse in GitHub*. MSR 2017 dblp, from http://web.cs.ucdavis.edu/ filkov/papers/clones.pdf

[5] Di Yang, Pedro Martins, Vaibhav Saini, Cristina Lopes *Stack Overflow in Github: Any Snippets There?*. MSR 2017 dblp, from https://arxiv.org/pdf/1705.01198.pdf