

What Do Developers Use the Crowd For? A Study Using Stack Overflow

Rabe Abdalkareem, Emad Shihab, *Member, IEEE* and Juergen Rilling, *Member, IEEE*,
{rab_abdu,eshihab,juergen.rilling}@encs.concordia.ca

Abstract—Stack Overflow relies on the crowd to construct quality developer-related knowledge. What developers use the crowd constructed knowledge on Stack Overflow for is an open question. We answer this question, by analyzing 1,414 Stack Overflow related commits and observe that developers use this crowd based knowledge to support development tasks and to collect user feedback. We also studied the helpfulness and delay of Stack Overflow posts by identifying the type of questions that are more likely to be answered by the crowd. We find that development tools and programming language issues are areas where the crowd is most helpful and that web framework related issues are the most time consuming questions to receive an accepted answer for. Our findings can help developers better understand how to effectively use Stack Overflow as a development support tool, help Stack Overflow designers to improve their platform, and the research community to understand the strengths and weaknesses of Stack Overflow as a development tool.

Index Terms—Crowdsourcing, Stack Overflow, Analyzing Commits.

I. INTRODUCTION AND BACKGROUND

QUESTION and Answer sites (Q&A), such as Stack Overflow, are extremely popular amongst software developers. Such Q&A sites heavily rely on the contributions of crowds to provide accumulated, quality knowledge to the software development community. Typically, users post questions related to software development topics on these Q&A sites that are answered by one or more participants. In essence, the job of answering the questions is outsourced to the crowd [1].

Over the years, the role of Q&A sites has evolved to more than just answering questions. However, what the role of Q&A sites plays in today’s development lifecycle is still an open question. Therefore, the goal of our paper is to answer the questions “What reasons do developers resort to the crowd on Stack Overflow for?” and “What areas is the crowd most helpful in? and what areas take longer to obtain answers for?”. Answering these questions helps developers better understand what knowledge they can obtain from the crowd, what knowledge tends to be most helpful and what knowledge may take longer to attain.

Other studies have focused on a similar goal. For example, Treude *et al.* [2] qualitatively analyzed a sample of Stack Overflow questions, and found that developers use Stack Overflow to share knowledge, provide development support, learn new technologies, and search for solutions to both common and specific programming problems. However, a key

difference of our study is that we specifically examine cases where a developer making a code commit on GitHub has explicitly referenced a Stack Overflow post as a knowledge resource. The explicit mentioning of the Stack Overflow posts 1) gives us confidence that a Stack Overflow is indeed related to a particular code commit, 2) allows us to build a richer dataset (since we have a link to the commit and the associated Stack Overflow post), and 3) indicates cases where developers see a need for traceability/justification/documentation for their committed changes. Moreover, our study does not focus on a specific type of task (e.g., bug triage [3]), domain (e.g., mobile [4]), or programming language, rather we examine all type of commits that explicitly mention Stack Overflow. Doing so helps provide a more holistic view of the reasons that developers use Stack Overflow.

Our findings corroborate some of the earlier observations, i.e., that developers use Stack Overflow to gain knowledge, and provide additional insights in terms of the actual Stack Overflow knowledge type that is directly applied by programmers (e.g., we find that knowledge about specific programming language and API usage are the most common types of knowledge use from Stack Overflow). At the same time, some of our findings are novel and have not been reported in any of the earlier work. For example, we find that some developers use Stack Overflow to document known bugs and even implement features based on Stack Overflow posts. Moreover, we find that the crowd is most helpful in resolving questions related to development tools, programming languages, and implementation issues and that the most time consuming posts to answer are posts related to web frameworks and the documentation of bugs. The findings shed light on what developers use Stack Overflow for so that we can gain a better understanding of areas where crowd knowledge is most resorted to. Moreover, our study of the most helpful areas and the areas that take longest to acquire can be used to emphasize knowledge that requires more attention from practitioners and researchers that contribute to Stack Overflow.

II. RELATED WORK

In addition to the study by Treude *et al.* [2] mentioned above, other works also studied how developers use Q&A sites. Barua *et al.* [5] proposed a semi-automatic approach to study general topics discussed on Stack Overflow and their trends, and found that web and mobile development are the most popular topics. Bajaj *et al.* [6] used Stack Overflow data to analyze common challenges and misconceptions among

Rabe Abdalkareem, Emad Shihab and Juergen Rilling are with the Department of Computer Science & Software Engineering, Concordia University, Montreal, Canada.

web developers. Rosen and Shihab [4] used Stack Overflow to determine what mobile developers on Stack Overflow ask about. Other researchers have performed studies that examine how Stack Overflow affects developers' activities during software development. For example, Vasilescu et al. [7] analyzed the effect of Stack Overflow activities on the software development process. They established associations between GitHub and Stack Overflow users, and found a correlation between participants' activities in the two platforms. Zagalsky *et al.* [8] also investigated the use of Stack Overflow and mailing lists as communication channels for the R project. They found that both resources provide active communication channels where participants are willing to help others. They also observed that Stack Overflow resorts to a crowd-based knowledge construction approach, where participants contribute knowledge independently, whereas for mailing list the focus is on improving specific answers.

In many ways, our work shares similar goals as these prior studies, i.e., to determine what developers use the crowd for during software development. However, our study differs in that we only consider explicit links between Stack Overflow posts and source code commits. Moreover, we use characteristics derived from these posts and commits to understand what knowledge is most helpful and what knowledge is most time consuming to attain.

III. STUDY DESIGN & APPROACH

The goal of our study is to determine the reasons that developers use the crowd for in their own projects, what areas they find the crowd to be the most helpful in and the areas that are most time consuming to attain answers for. In the following sections, we describe how we collect our *Stack Overflow related commits*, how we classify them, and how we measure their helpfulness and delay.

A. Selection of Studied Projects

To conduct our study, we first need to identify the software projects that contain Stack Overflow related commits. It is important to study a large sample of software projects in order to improve confidence in our analysis results. To select the projects that we want to study, we used the GHTorrent dataset [9] to obtain a list of non-forked projects (main-line) written in the most popular programming languages [10]: Ruby, Python, JavaScript, PHP, Java, Scala, C and C++. Based on our selection criteria, we are able to identify 4,163,814 projects. However, since it is a well known fact that GitHub contains a large number of software projects that are inactive or immature, we set a few other constraints to ensure that we only consider active and mature projects. Thus, we only considered projects that: (1) have at least 100 pull request, (2) have at least three developers, and (3) have more than 100 commits in the last year. Similar constraints were recommended in [11]. Applying this filtering further reduced the number of projects to 4,026, which we tried to clone for analysis. Since some projects were no longer available (e.g., they were deleted or made private), we were able to clone and study a total of 3,974 projects.

TABLE I: Statistics of the No. of Projects, Languages and Stack Overflow Commits Used in Our Study.

Language	# Projects	# Stack Overflow Related Commits
JavaScript	189	307
Python	179	348
Ruby	132	227
Java	123	193
PHP	99	154
C/C++	65	131
Scala	21	54
Total	808	1,414

B. Extracting Stack Overflow Related Commits

After selecting the list of software projects, the next step is to identify Stack Overflow related commits. To do so, we rely on string pattern matching techniques to detect these commits. For each project, we search all of their commit logs for the term 'stackoverflow' and its variants (i.e., capitalized first letter, all capitalized, with spaces). After applying the pattern matching technique, we obtained 1,780 *Stack Overflow related commits* that originated from 929 projects. As a final step, we performed a manual inspection of the commits and their associated projects to filter out duplicates and irrelevant commits (false positives). In the end, we were left with 1,414 commits from 808¹ projects to be analyzed. Of all the Stack Overflow related commits, ~97% of them contained a link to the Stack Overflow related post. Table I shows the descriptive statistics of our dataset. The projects in the dataset cover several programming languages and each programming language has a number of related projects and commits.

C. Classifying Stack Overflow Related Commits

Once we determined the number of *Stack Overflow related commits* to examine, we performed an iterative coding process to identify and categorize the different reasons that developers use Stack Overflow [12]. We first inspected every commit message, the source code associated with the commit, and the Stack Overflow post referenced in the commit. We read the main issues discussed in the commit message and the Stack Overflow post, which helps us determine how to classify the commit. The aforementioned process was performed iteratively in that every time a new category is added, we re-examine all the previously classified commits to determine if the categorization changed. As a result of this classification process, we ended up with 14 different reasons why developers mention Stack Overflow in source code commits.

Like any human activity, our classification is prone to human bias. To examine the validity of the classification, we got another PhD student (in addition to the first authors) to independently classify a statistically significant sample of commits to reach a 95% confidence level using a 5% confidence interval. The statistically significant sample of 302 Stack Overflow related commits was classified into different areas and Cohen's Kappa coefficient was used to evaluate

¹http://das.encs.concordia.ca/wp-content/uploads/2016/12/list_of_studied_projects.csv

the level of agreement between the two annotators. Cohen's Kappa coefficient is a well-known statistical method that is used to evaluate the inter-rater agreement level for categorical scales. The resulting coefficient is scaled to range between -1.0 and +1.0, where a negative value means poorer than chance agreement, zero indicates exactly chance agreement, and a positive value indicates better than chance agreement. In our work, we found the level of agreement between the annotators to be +0.78, which is considered to be excellent agreement [13].

D. Measuring the Helpfulness and Delay of Stack Overflow Posts

The second goal of our study is to better understand which areas developers find the crowd to be most helpful in and which areas take longer to attain an answer to, i.e., which posts take longer to receive an accepted answer for.

As a proxy for helpfulness, we use the number of votes (sum of upvotes - downvotes) that a Stack Overflow question receives. Our intuition here is that if a question is helpful to a developer, then they will give it an upvote, which indicates that this question/post is helpful. The more votes a question has, the more helpful it is considered to be. Once we measure the votes for the individual posts, we group them into their respective areas. To provide a **helpfulness** measure for a specific area, we present the median of the votes for all of its posts.

We further argue that if a post takes longer time to obtain an accepted answer, then the developer will be delayed more, which is negatively perceived. Therefore, we use the time to obtain an accepted answer for posts in a specific area as a proxy for **delay**. For each area, we measure the time difference between the initial post (question) and its first accepted answer. Then, we aggregate all of the values and present the median time per area.

IV. RESULTS

To understand what developers use the crowd constructed knowledge on Stack Overflow for, we break down our study into two main parts and associate a research question with each part:

- RQ1 What are the main reasons developers resort to Stack Overflow?
- RQ2 What areas is the crowd most helpful to developers in? What areas takes longest to attain answers for from the crowd?

Answering RQ1 helps us to determine what developers use Stack Overflow for so that we can gain a better understanding of areas where crowd knowledge is most resorted to. We can use our findings to further facilitate the integration of crowdsourcing into software development. Answering RQ2 helps us to better understand what types of knowledge from Stack Overflow is considered most helpful by the developers and takes longest to acquire. Identifying these types of knowledge can be used to emphasize knowledge that requires more attention from practitioners and researchers that contribute to Stack Overflow.

RQ1. What are the main reasons developers resort to Stack Overflow?

As mentioned earlier, we manually examined each commit message, the code changes associated with a commit, and the Stack Overflow post mentioned in the commit to determine the reason the commit mentions the Stack Overflow post. Thus, we use our classification to identify the reasons why developers use Stack Overflow.

Table II shows the 14 different reasons developers use Stack Overflow, that are grouped into five high-level categories, namely 'Using Knowledge', 'Documenting Bugs', 'Promoting Stack Overflow', 'Feature/System Improvements' and 'Code Reuse'; another category, 'Other', was added to categorize commits that rarely appeared and/or did not fit into any of the major categories. For each reason, we provide a description, an example and the frequency they occurred (as a percentage of commits). As shown in Table II, we found that developers resort to the crowd on Stack Overflow mainly to gain knowledge. The most frequent knowledge is related to programming languages (in 22.07% of the commits), to ask about API use (in 21% of the commits), configuration management (in 7.21%), gain knowledge about web frameworks (in 6.51% of the commits), and web browsers (in 4.31% of the commits). From our analysis of the commits and posts related to the aforementioned categories, we observed that the developers mainly take advantage of the technical knowledge provided by the crowd on Stack Overflow. Our findings show the key role of the crowd is to support and complement traditional documentation.

Also, we found that developers use Stack Overflow to document bugs (in 13.08% of the commits) and even for feature/system improvements that they implement (in 1.77%). These findings show that the role of the crowd on Stack Overflow is more than just providing knowledge or finding relevant code. The crowd on Stack Overflow also provides insight on known issues and features that users would like to see. These results suggest that developers pay attention to such issues raised by the crowd and that Stack Overflow serves as medium for identifying and tracking feature requests and issues.

The other interesting categories were related to the promotion of Stack Overflow in a project (in 3.18% of the commits), where developers would introduce a tag on Stack Overflow to facilitate documentation. Moreover, we found that the direct reuse of code from Stack Overflow is very minimal in terms of the number of explicit mentioning of the Stack Overflow posts in commit messages (in 1.70% of the commits). However, we believe that developers reuse more code than they admit due to various reasons such as potential copyright violations or plagiarism.

RQ2. What areas is the crowd most helpful to developers in? What areas takes longest to attain answers for from the crowd?

Now that we have identified the different reasons developers use Stack Overflow, we want to better understand what knowledge the crowd is most helpful for and what knowledge takes longest to acquire.

TABLE II: Identified reasons for using Stack Overflow with a description, example and the percentage of commits in each area.

Reasons for Using Stack Overflow	Description	Example	% Commits	
Using Knowledge	Programming Languages	Any knowledge related to programming languages and their features. For instance, using format in Python, using sequential for loop in JavaScript, how to do casting, regular expression, or a programming language limitation.	"Changed all boolean casts that were using Boolean() function to use double negation(!!), which is faster: [StackOverflow Link] "	22.07%
	API Usage	Knowledge related to how to use an API including argument, deprecation, specify method to perform a task in an API.	"mailutils: send_email() with attachments* Extends mailutils. send_email() API to support attachments, following the recipe: [StackOverflow Link] "	21%
	Configuration Management	Knowledge related to configuration management. For example, knowledge on how to configure Maven tool in the development environment.	"Fix maven assembly warning about using root dir. It's a bad practice in Maven to define '/' as the output dir. It's better to leave it empty. See also [StackOverflow Link] ."	7.21%
	Web Frameworks	Stack Overflow posts related to the usage of web frameworks and their configurations.	"Fix client names with dot do not work this is Spring Framework MVC behavior as described in [StackOverflow Link] "	6.51%
	Web Browsers	A developer uses knowledge regarding web browsers. For example, the presence or absence of features in specific browsers.	"Fix grid context menu position for Firefox. Firefox does not have offsetX. pageX is absolute and lets the menu jump all over the place. Solution based on [StackOverflow Link] "	4.31%
	Development Tools	Knowledge related to configuring development tools (e.g., IDE, Git, SVN) versions, settings, etc.	"Update Git to delete a remote branch with '-delete' more memorable syntax. Use Git1.7 syntax based on this answer: [StackOverflow Link] "	4.17%
	Implementation Issues	Developers used suggestions or tutorials from Stack Overflow posts to implement an algorithm or a feature in their projects without copying and pasting source code.	"Introduce the non-daemon process pool as an alternative to the original multiprocessing pool. This adds support for hierarchical multiprocessing (child classes can use multiprocessing again). The code is based on the following StackOverflow answer: [StackOverflow Link] ."	3.89%
	Database Technologies	Knowledge related to database and their supporting technologies (e.g., database configurations, maintenance, modeling, queries, etc.).	"Postgres column renaming. Switched "name" column name to "shoreline_name" so we don't collide with possible Postgreskeywords/types [StackOverflow Link] ."	2.83%
	Operating Systems	Knowledge related to operating systems features or issues.	"Explicitly set empty extension name for backup files on Mac, this parameter is needed, otherwise an error is shown. See this SO post for more information: [StackOverflow Link] "	2.40%
Documenting Bugs	The developer fixed a bug in the project and provided the link to the Stack Overflow post where the bug has been described.	"Fix AttributeError when IssueEvent has assignee. This was discovered by a user on StackOverflow [StackOverflow Link] and fixed as soon as I realized it was a bug."	13.08%	
Promoting Stack Overflow	A developer introduced a tag on Stack Overflow related to his/her project to facilitate its documentation or to promote the usage of their tag on Stack Overflow.	"Promote stackoverflow for questions", "Drop google groups in favor of stackoverflow tag.", "Link to [StackOverflow Link] for Q and A Thanks to Vincent Scheib for arranging and Paul Kinlan for donating his Stack Overflow karma to create the tag".	3.18%	
Feature/System Improvements	A developer implements a new feature or improves the project based on Stack Overflow users request.	"Extend key bindings for prompt commands to support predefined searches This adds support for binding keys to ':'/' and ':?'.for example: bind stage 2: Based on this request by Joelpet on stackoverflow: [StackOverflow Link] "	1.77%	
Code Reuse	A developer copies and pasts a source code snippet from a Stack Overflow post.	"Close tip popup on click outside the tip box. Credit: [StackOverflow Link] "	1.70%	
Other	Developers use knowledge from Stack Overflow, but we cannot identify the type of usage exactly or some rare cases that it is not worth of having a separate category for them.		5.87%	

TABLE III: The Helpfulness & Delay classified by the different reasons of using knowledge from Stack Overflow.

Reason	Helpfulness			Delay		
	# Questions	Median of Votes [§]	Rank	# Accepted Answers	Median of Time (Hours) [†]	Rank
Development Tools	61	39	1	52	6.7	9
Programming Languages	310	29	2	279	0.3	1
Implementation Issues	55	26	3	48	2.6	4
Configuration Management	102	19	4	87	5.6	7
Database Technologies	41	17	5	37	1.1	3
Web Browsers	61	16	6	50	0.5	2
Web Frameworks	93	15	7	77	13	11
Operating Systems	37	13	8	33	0.5	2
API Usage	301	10	9	254	3.1	5
Code Reuse	25	7	10	19	6.1	8
Feature/System Improvements	27	3	11	18	3.6	6
Documenting Bugs	178	2	12	139	9.4	10
Promoting Stack Overflow	0	0	NA	0	0	NA
Other	76	15	-	59	2.6	-

[§]The median of number of votes a question receive on Stack Overflow.

[†]The median time taken for a question to receive an accepted answer.

Table III shows the number of questions, the median number of votes, the number of accepted answers, and the median time (hours) to obtain an accepted answer for each reason. The ascending ranking for helpfulness is based on the median votes for the different reasons of knowledge reuse from Stack Overflow and the descending ranking for delay is based on the median hours it takes to obtain an accepted answer for a given category.

From Table III, we observe that posts related to development tools, programming languages, implementation issues, and configuration management are areas where the crowd can be considered to be the most helpful. On the other hand, the posts related to web frameworks, documenting bugs, and development tools are areas that take the longest to answer.

However, the most interesting analysis comes from combining these two views, i.e., helpfulness and delay, to determine areas that developers can expect to receive helpful answers in a timely fashion and vice versa. We use a bubble plot, shown in Figure 1, that plots the ranks of helpfulness vs. delay for each area of using Stack Overflow. The size of the bubble represents the number of commits for a particular area. From Figure 1, we observe that areas such as 'implementation issues', 'programming languages', 'database technologies', and 'web browsers' provide the highest utility for developers, i.e., developers receive very helpful and quick answers. On the other hand, areas such as 'API usage' and 'operating systems' tend to be answered quickly, but the answers given are not perceived to be very helpful. Similarly, answers to questions in areas such as 'web frameworks' and 'documenting bugs' are perceived to be less helpful and take even longer to answer. In such cases, the crowd on Stack Overflow may not be the right resource for developers who are looking for answers.

V. LOOKING AHEAD

There are two key observations that we believe will impact the future development of Q&A sites and how developers use the crowd. The first observation can be helpful for Stack Overflow designers to enhance current Stack Overflow features to meet the increasing demand from developers. The second

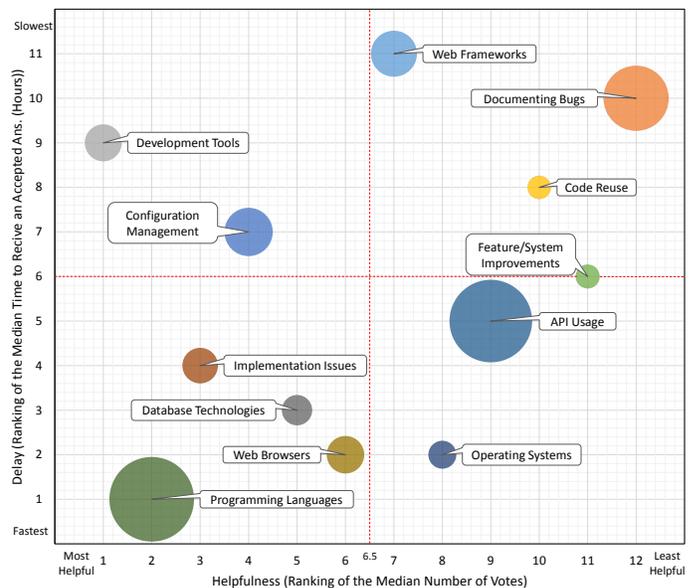


Fig. 1: The x-axis of the plot shows the average ranking of the median of number of votes for each area (1 is the most helpful). The y-axis shows the median time (hour) for a question to receive an accepted answer for each area (1 as the fastest), while the size of the bubble present the number of commits.

is an observation that can help developers improve traceability and documentation of their changes.

Observation 1: Using the Crowd for More Than Just Knowledge

Clearly, our results show that the majority of developers use Stack Overflow to gain knowledge. However, we also observed that developers use the crowd on Stack Overflow for more than acquiring knowledge. In fact, developers seem to be using these crowd based resources to document bugs and determine features that they want to implement. Hence, we believe that future versions of Stack Overflow need to

incorporate a mechanism where developers can obtain direct feedback from the crowd. Presently, we see that role of the crowd to mainly focus on reporting bugs and requesting requirements. However, we envision that in the future the crowd on Stack Overflow will play an increasing role, as a source for refining requirements, providing testing, and even helping refine software design. Another interesting finding is that developers reuse code snippet from Stack Overflow. How to ensure the quality or integrity of these shared code snippets is an area where crowd based platforms can do better (now all someone can do is give an upvote). Also, providing some sort of score that indicates the ‘adaptability’ or ‘ease of integration’ of a code snippet would be beneficial.

One important suggestion based on this specific finding for Stack Overflow designers is to provide techniques to assess the quality of source code snippets posted on Stack Overflow. In order to investigate and find ways one can automatically generate test cases for source code snippets posted on Stack Overflow, for example.

Observation 2: Linking Changes to Crowd Discussion

Our dataset is based on the fact that certain developers explicitly mentioned the Stack Overflow posts in their commit messages. With the increasing use of the crowd in software development, we believe that developers should link to discussions that they used to help reach their final coded solutions. These discussions can help document and provide rationale for certain decisions, hence developers should provide links to them in their commits. Much like how commits contain bug IDs, we believe that in the future, every commit should also provide a link to any crowd-based discussions that are related. One unique feature of these crowd-based discussions is that they continue to evolve, and so even if a bug is found in posted code, others may help provide an update or a fix in the future. If links to these discussions are provided, this evolved code can help address future issues with the code.

VI. LIMITATIONS

There are a number of limitations to our study. First, the commits were manually classified by the first author. Like any human activity, this process is susceptible to human error. To ensure the validity of the classification, we got another PhD student to classify a statistically significant sample of 308 commits and found their agreement to be excellent (Cohen’s Kappa value of +0.78). Also, our findings are based on 1,414 commits, where developers explicitly mention Stack Overflow. There may be other cases where developers use Stack Overflow, but do not mention it in the commit message. Lastly, our study is based on open source projects that are hosted on GitHub, therefore, our study may not generalize to other open source or commercial projects.

VII. CONCLUSION

We investigate the reasons developers use Stack Overflow for and what areas the crowd is most helpful and what areas are most time consuming to attain answers for. We find that the crowd mostly provide technical knowledge to developers,

however, the role of crowd-based sites, such as Stack Overflow is evolving. Our results revealed that using crowd knowledge through Q&A platforms, such as Stack Overflow, can be used for various purposes of software development process including collecting users’ feedback and code reuse. We draw from our findings to suggest that crowd-based sites such as Stack Overflow provide tools to support feedback from the crowd to developers and provide mechanisms to evaluate the quality of code posted on such sites. For developers, the ability to provide direct links to crowd-based resources will become an essential since such links can serve as living documentation of their code and/or design decisions. In addition to its direct findings, our study highlights areas where crowd based knowledge may not be the best fit, e.g., for questions related to web frameworks.

In the future, we plan to examine the reasons that the crowd is perceived to be more helpful in certain topics and the reasons for delay in answers other areas. In particular, we plan to study the impact of technological and socio-technical factors on the ability of different areas to receive helpful and timely answers.

REFERENCES

- [1] T. D. LaToza and A. van der Hoek, “Crowdsourcing in software engineering: Models, motivations, and challenges,” *IEEE Software*, vol. 33, no. 1, pp. 74–80, 2016.
- [2] C. Treude, O. Barzilay, and M.-A. Storey, “How do programmers ask and answer questions on the web? (nier track),” in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE’11. ACM, 2011, pp. 804–807.
- [3] A. S. Badashian, A. Hindle, and E. Stroulia, “Crowdsourced bug triaging,” in *Proceedings of the IEEE International Conference on Software Maintenance and Evolution*, ser. ICSME’15. IEEE, 2015, pp. 506–510.
- [4] C. Rosen and E. Shihab, “What are mobile developers asking about? a large scale study using stack overflow,” *Empirical Software Engineering (EMSE)*, vol. 21, no. 3, pp. 1192–1223, 2016.
- [5] A. Barua, S. W. Thomas, and A. E. Hassan, “What are developers talking about? an analysis of topics and trends in stack overflow,” *Empirical Software Engineering (EMSE)*, vol. 19, no. 3, pp. 619–654, 2014.
- [6] K. Bajaj, K. Pattabiraman, and A. Mesbah, “Mining questions asked by web developers,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR, 2014, pp. 112–121.
- [7] B. Vasilescu, V. Filkov, and A. Serebrenik, “Stackoverflow and github: Associations between software development and crowdsourced knowledge,” in *Proceedings of International Conference on Social Computing*, ser. SocialCom’13. IEEE, 2013, pp. 188–195.
- [8] A. Zagalsky, C. G. Teshima, D. M. German, M.-A. Storey, and G. Poo-Caamaño, “How the r community creates and curates knowledge: a comparative study of stack overflow and mailing lists,” in *Proceedings of the 13th International Workshop on Mining Software Repositories*, ser. MSR’16. ACM, 2016, pp. 441–451.
- [9] G. Gousios, “The ghtorrent dataset and tool suite,” in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR’13. IEEE, 2013, pp. 233–236.
- [10] B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, and V. Filkov, “Quality and productivity outcomes relating to continuous integration in github,” in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE’15. ACM, 2015, pp. 805–816.
- [11] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, “The promises and perils of mining github,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR’14, 2014, pp. 92–101.
- [12] C. B. Seaman, “Qualitative methods in empirical studies of software engineering,” *IEEE Transactions on Software Engineering (IST)*, vol. 25, no. 4, pp. 557–572, 1999.
- [13] J. Fleiss, “The measurement of interrater agreement,” *Statistics methods for rates and proportions*, pp. 212–236, 1981.