Software Engineers’ Questions and Answers on Stack Exchange

Matúš Sulír, Marcel Regeci

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
matus.sulir@tuke.sk, marcel.regeci@student.tuke.sk

Abstract—There exists a large number of research works analyzing questions and answers on the popular Stack Overflow website. However, other sub-sites of the Stack Exchange platform are studied rarely. In this paper, we analyze the questions and answers on the Software Engineering Stack Exchange site that encompasses a broader set of areas, such as testing or software processes. Topics and quantities of the questions, historical trends, and the authors’ sentiment were analyzed using downloaded datasets. We found that the asked questions are most frequently related to database systems, quality assurance, and agile software development. The most attractive topics were career and teamwork problems, and the least attractive ones were network programming and software modeling. Historically, the topic of domain-driven design recorded the highest rise, and jobs and career the most significant fall. The number of new questions dropped, while the portion of unanswered ones increased.

Index Terms—software engineering, Stack Exchange, question-and-answer website, analysis

I. INTRODUCTION

The Stack Exchange platform is a network of Q&A (question and answer) sites that enable people to collaborate effectively in finding solutions and answers to questions related to a wide variety of topics. It is used by millions of people every day to solve problems they encounter in their jobs or learn more about their hobbies. The original, oldest site – Stack Overflow1 – is focused on programming questions related to specific source code excerpts, such as algorithmic problems in a given program or technical questions about frameworks and libraries. Over the course of years, many sub-sites emerged on the Stack Exchange platform, including sites focused on mathematics, physics, or even cooking. One of the components of the platform is the Software Engineering Stack Exchange2 site designed for professionals, academics, and students working in the software systems development lifecycle.

Software Engineering Stack Exchange is the 16th oldest site of the Stack Exchange platform, with about 350,000 users and 60,000 asked questions3. The focus of this site is defined by a set of adequate question areas, which are namely: software development methods and procedures, requirements, architecture and design, quality assurance and testing, configuration management, assembly, release, and deployment4.

In the last decade, numerous papers analyzed Stack Overflow from various aspects. For example, Barua et al. [1] provided a general overview of topics and trends. Asaduzzaman et al. [2] focused on unanswered questions. Other papers analyzed specific topics of questions, such as chatbots [3] or concurrent programming [4].

However, we are not aware of any paper analyzing the Software Engineering Stack Exchange site, with the exception of a study by Verma et al. [5] that focused solely on the analysis of topics using LDA (Latent Dirichlet Allocation) during the years 2015–2017.

Therefore, in this paper, we would like to provide the results of the analysis of Software Engineering Stack Exchange from its inception in 2008 until April 2021, comprising the discussion of the most common and attractive topics, their historical development, the trends in the numbers of (un)answered questions asked over time, and the sentiment of the authors.

The analysis of Software Engineering Stack Exchange is interesting for two main reasons. First, due to the long-term operation of the platform and its high popularity, a large amount of data was collected, which constitutes a unique representation of the software engineers’ interests and attitudes. Second, the analysis of the data could help better understand the platform and thus optimize its operation and increase the effectiveness of collective cooperation of users by developing and applying new methods and policies.

II. RELATED WORK

A vast number of research works analyzed the main site of the Stack Exchange platform: Stack Overflow. Barua et al. [1] provided an overview of the main discussion topics and their coupling, along with the changes in developers’ interests and technologies used over time.

Asaduzzaman et al. [2] mined unanswered questions from the mentioned website and tried to reveal characteristics that make questions difficult to answer. Baltes et al. [6] studied the

1https://stackoverflow.com
2https://softwareengineering.stackexchange.com
3https://stackexchange.com/sites?view=list#technology-traffic
4https://softwareengineering.stackexchange.com/help/on-topic
We appended the text of the accepted answer to the body of the question since the author of the accepted answer is often an expert that uses proper domain terms, which should improve the quality of topic modeling. The resulting 28,077 questions were saved into a CSV (comma-separated values) file.

Then we pre-processed the text of the questions (including the appended accepted answers). Newline characters, punctuation, HTML tags, and code samples were removed. The text was transformed into a list of tokens – words in lowercase and without any special characters. We removed stopwords, such as articles, pronouns, or prepositions. The words were lemmatized, i.e., transformed to a base form (e.g., solving into solve). Frequently co-occurring words were merged into single tokens representing bigrams, for instance, “middle” and “ground” into “middle_ground”.

From the list of tokens, we created a dictionary by filtering out tokens that were present in either less than 30 documents (questions) or more than 20% of documents. This way, we excluded too specific or too general terms that would impair topic recognition. A corpus was then created, assigning the frequency of each token in the dictionary to every document.

The dictionary and the corpus were used as input of Mallet LDA [14]. The best model, based on the coherence score, was obtained for the number of topics set to 50. LDA assigns each document a set of topics along with a relevance score, which was used to determine the main (most relevant) topic for each question.

Since the topics produced by LDA are unnamed by themselves, we named each of the 50 topics by inspecting the list of the terms most relevant to it (as suggested by Mallet LDA) and manually reading a few associated documents if necessary.

B. Results

In Table I, we can see the list of the 20 most frequent topics of questions. The first column represents the portion of questions for which the given topic was identified as the main topic by LDA. The second column contains the name of the topic.

The asked questions were most frequently related to databases, quality assurance, agile software development, licensing, and asynchronous programming. However, the range of topics is quite diverse, and none of them prevails significantly.

The topic of database systems contained questions from a wide variety of sub-topics. Both traditional relational databases and NoSQL systems were discussed. The questions about relational databases included topics such as table joining, normalization and denormalization, indexes, views, or triggers. Many questions were related to performance concerns, particularly to the compromise between a clean database design and scalability. Object-relational mapping (ORM) was also frequently discussed. There were also questions looking at databases more broadly, e.g., about the design of an online database creator.

Questions about quality assurance were frequently related to test-driven development. Unit tests, integration tests, and
particular the distinction between them were also often discussed.

Agile software development questions discussed mainly terms related to Scrum (e.g., user stories, backlog, sprint planning, task estimation, story points), both from the theoretical point of view and specific practical problems that arose.

V. Most and Least Attractive Topics

Although questions regarding some topic may be frequently asked, it does not automatically mean this topic attracts attention from software engineers. For example, some questions are viewed by only a small group of visitors and remain unanswered for a long time, while others are frequently searched, viewed, and then heavily commented and answered. Therefore, in RQ2, we decided to study the attractiveness of individual topics.

A. Method

In RQ2, we used the data from the previous research question (RQ1), i.e., the complete Mallet LDA model. The names of all 50 topics and the assignment of each question to its main topic were retained.

For each question in the dataset, we calculated the Accumulated Post Score (AMS), which is a dimensionless number expressing the question’s importance based on its selected properties, such as the upvote and comment count. It was defined by Bajaj et al. [9] as:

\[ AMS = 3U - 25D + 10C + A + F \]

In the formula, \( U \) is the number of upvotes, \( D \) the number of downvotes, \( C \) is the comment count, \( A \) the answer count, and \( F \) the number of people who marked the question as favorite. In contrast to Bajaj et al. [9], the numbers of upvotes and downvotes were not counted separately, but as a total score, i.e., either positive (upvotes) or negative (downvotes).

Then we calculated the average AMS for every topic as a sum of the scores of each question for which this was the main topic divided by the number of questions.

B. Results

All topics were sorted by their average AMS. In Table II, there is a list of the most attractive topics according to this metric. The top three most attractive topics were jobs and career, teamwork problems, and code readability.

We can conclude that “jobs and career” is both a very attractive and frequently asked topic. The second item on the list, teamwork problems, also represents a socially oriented topic, which is often controversial and thus tends to produce more elaborate discussions. Code readability questions were frequently related to concerns such as style, comments, and indentation, which traditionally cause disputes between the proponents of individual options.

A list of the least attractive topics with their corresponding AMS is displayed in Table III. Questions related to network programming and web application backend development, software modeling, and access control attract too little attention from the platform users after they are asked.

One possible reason for the unattractiveness of network programming and web backend development questions on this website might probably be the existence of not only a well-known separate Stack Exchange site for programming (Stack Overflow) but also a site for networking (Networking Stack Exchange). Software modeling and particularly model-driven software development is often considered an academic approach by practitioners [15]. Finally, access control questions might face a similar problem as was already mentioned: There exists a sub-site of Stack Exchange dedicated to information security.

VI. Historical Trends

In RQ3, we will look at the development of the popularity of individual topics over the years, focusing on the most rising
TABLE III
THE 10 LEAST ATTRACTIVE TOPICS

<table>
<thead>
<tr>
<th>Score</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.8</td>
<td>network and backend</td>
</tr>
<tr>
<td>19.3</td>
<td>software modeling</td>
</tr>
<tr>
<td>19.4</td>
<td>access control</td>
</tr>
<tr>
<td>19.4</td>
<td>user data collection</td>
</tr>
<tr>
<td>20.6</td>
<td>asynchronous program execution</td>
</tr>
<tr>
<td>21.0</td>
<td>Model-View-Controller architecture</td>
</tr>
<tr>
<td>21.6</td>
<td>recursion</td>
</tr>
<tr>
<td>22.0</td>
<td>software development business contract</td>
</tr>
<tr>
<td>22.8</td>
<td>web services and microservices</td>
</tr>
<tr>
<td>22.9</td>
<td>sorting algorithms</td>
</tr>
</tbody>
</table>
B. Results

The plot of the number of all and unanswered questions asked over time is shown in Fig. 4. The overall asked question count increased almost four-fold from 2010 to 2011. After this year it started to decline steadily.

An interesting observation is that the number of questions without an accepted answer remained almost constant during the years 2011–2020. This means the portion of unanswered questions increased dramatically during these nine years, specifically from 32% to 56%.

VIII. AUTHORS’ SENTIMENT

Ideally, the authors on the Stack Exchange platform should remain objective and avoid too emotional responses. In reality, they sometimes express also emotions in their posts. In RQ5 we would like to determine if there is a relationship between the sentiment (subjectivity and polarity) of the answers and the reputation of their authors. The reputation on Stack Exchange is a numerical score that the users earn for asking, answering, and editing questions. Basically, a higher reputation means higher experience with the given site.

A. Method

Using Stack Exchange Data Explorer, we downloaded the texts of answers having at least 100 characters since shorter text could be unsuitable for sentiment analysis. To prevent the assessment of users based on merely one or two posts, only users having at least 3 such answers on the website were included in our query. Because of the limitations of the tool, for users having more than fifteen answers we selected a random sample of size 15. The resulting dataset contained 49,284 answers.

Each answer was pre-processed by removing newlines, HTML tags, and code snippets. Polarity (positivity vs. negativity) and subjectivity (the degree to which the given text is subjective) were computed using the TextBlob library [16].

For every user, we calculated the average subjectivity and polarity. A measure of correlation capturing also nonlinear relationships, Spearman’s rho, was computed first between the reputation and subjectivity and then between the reputation and polarity of users.

B. Results

Fig. 5 displays a hexbin plot, where the reputation of users is on the x-axis and the average subjectivity on the y-axis. The logarithmic color scale represents the number of users in each hexagonal bin. Extreme outliers, namely 8 users with a reputation higher than 100,000, were removed from the plot. While there exist some predominantly subjective or objective users, a vast majority is located in the middle. The plot shows no signs of correlation, which is confirmed by Spearman’s rho equal to 0.05 (where 0 means no correlation and 1 perfect positive correlation).

Similarly, Fig. 6 contains a hexbin plot of the users’ average polarity based on their reputation. The majority of users tend
IX. Threats to Validity

Now we will describe threats to the validity of our study.

A. Construct Validity

For the individual research questions, we analyzed only subsets of all data available on Software Engineering Stack Exchange. In RQ1–RQ3, we used only questions having a positive score and an accepted answer. A limited year range was analyzed in RQ3 and RQ4, and a random subset of answers was queried for users with too many posts in RQ5. However, we always provided a rationale for the given selection criteria.

B. Internal Validity

Rapid changes in the historical development of topics might be caused by new or changed rules coming into effect. For example, the decline of the topic “jobs and career” was probably caused not by the lack of software engineers’ interest in their careers but due to many questions about jobs being marked as not relevant by the community.

C. External Validity

The study was based on a single data source, namely the Software Engineering Stack Exchange site, so it might not be representative of all software engineers’ questions about the software systems development lifecycle and related areas. Although the analyzed website contains more than 200,000 posts, extending the study to use multiple data sources would improve its external validity.

D. Reliability

The naming of individual LDA topics was performed by one of the authors. However, the other author checked all assignments based on the most relevant terms and also a subset of names by skimming the text of selected posts assigned to a given topic.

Jupyter notebooks and data dumps used to produce the results described in this study are publicly available in a permanent repository.6

X. Conclusion and Future Work

We analyzed the questions and answers on the Software Engineering Stack Exchange website from various aspects. According to our findings, the topics of most frequently asked questions were database systems, quality assurance, and agile software development.

The most attractive topics, taking into account metrics such as upvote, answer, and comment counts, were jobs and career, teamwork problems, and code readability. On the other hand, questions related to web application backend, software modeling, and access control were among the least attractive ones.

The new question count on the analyzed website was rising until 2011 but started to decline gradually after this year. Since the number of unanswered questions remained almost the same, their relative proportion significantly increased.

During the years 2010–2020, domain-driven design, asynchronous programming, and inheritance were the most rising trends. Jobs and career, education and research, and software

6https://doi.org/10.17605/OSF.IO/M8STH
licensing question counts dropped the most. However, this does not mean software engineers are no longer interested in these questions – instead, they started to be considered off-topic on the given website. This creates a fragmentation problem, where the developer must carefully consider at which Stack Exchange sub-site to ask a given question, if ever.

Regarding the authors’ sentiment, we found no correlation between the reputation of users and the average subjectivity or polarity of their answers.

In the future, we could answer more research questions, particularly related to comments and bounties. We would also analyze more data sources, including the main Stack Overflow website and third-party question-answer sites, and compare the results. Finally, other topical sub-sites of the Stack Exchange platform also provide a wealth of information suitable for analysis.

ACKNOWLEDGMENT

This work was supported by VEGA Grant No. 1/0630/22 Lowering Programmers’ Cognitive Load Using Context-Dependent Dialogs.

REFERENCES


