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 Overflow[1] – 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 Exchange[2] 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 questions[3]. 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 deployment[4].

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

[1] https://stackoverflow.com

[2] https://softwareengineering.stackexchange.com

[3] https://stackexchange.com/sites?view=list#technology-traffic

[4] https://softwareengineering.stackexchange.com/help/on-topic

This is the accepted version of: M. Sulír, M. Regeci. Software Engineers' Questions and Answers on Stack Exchange. 2022 IEEE 16th International Scientific Conference on Informatics, IEEE, 2022, pp. 304–310. http://doi.org/10.1109/Informatics57926.2022.10083403

duplication of code snippets that are often included on Stack Overflow. Zhang et al. [7] found that obsolete answers are sometimes posted on this website, and they are rarely updated.

Beyer et al. [8] tried to automatically classify questions on Stack Overflow into categories, such as API (application programming interface) usage, discrepancy, errors, review, or conceptual.

Some papers analyzed specific question topics on Stack Overflow: the challenges of chatbot development [3], questions about concurrent programming [4], web developers' discussions [9], privacy-related questions [10], and Docker challenges [11].

A few research works analyzed other sub-sites on the Stack Exchange platform. Among them, Kou and Gray [12] studied the evolution of UX (user experience) knowledge on UX Stack Exchange. Kamienski and Bezemer [13] used multiple question&answer websites for game developers, including Game Development Stack Exchange, as their data sources.

In contrast to the mentioned papers, we focused on the Software Engineering Stack Exchange website and its topic areas. Although a study by Verma et al. [5] also used the mentioned website as its data source, there are two key differences. First, they analyzed solely the topics and trends using LDA, while we included also research questions about (un)answered question quantities and the authors' sentiment in our study. Second, the study by Verma et al. focused only on the years 2015–2017, while we analyzed much broader year ranges.

III. Research Questions

We were interested in the following research questions:

RQ1 *What are the most common topics of questions asked at Software Engineering Stack Exchange?*

RQ2 *Which topics attract the most and least attention?*

RQ3 *What was the historical development of the topics?*

RQ4 *How did the number of all and unanswered newly-asked questions change over time?*

RQ5 *Is the sentiment of the answers related to the reputation of their authors?*

In the following sections, we will always describe the methods used to answer each of the research questions and then present the results.

IV. Common Topics

First, we would like to know to what topics the software engineers' questions on the platform most frequently pertain, which is the subject of RQ1.

A. Method

To obtain the data for analysis, we used the Stack Exchange Data Explorer[5]. Using an SQL query, we downloaded all questions that have a score of 1 or more (meaning they were marked as useful and relevant by at least one user) and have an accepted answer associated with them. 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

[5] https://data.stackexchange.com

TABLE I
THE TOP 20 MOST FREQUENT TOPICS OF THE ASKED QUESTIONS

Frequency	Topic
3.9 %	database systems
3.7 %	quality assurance
3.5 %	agile software development
3.4 %	software licensing
3.2 %	asynchronous program execution
3.1 %	jobs and career
3.0 %	version control systems
2.8 %	web application frontend
2.8 %	encapsulation
2.8 %	inheritance
2.7 %	education and research
2.6 %	domain-driven design
2.5 %	software configuration
2.4 %	functional programming
2.3 %	recursion
2.3 %	source code compilation
2.3 %	authentication and authorization
2.3 %	Model-View-Controller architecture
2.2 %	exception handling
2.0 %	network and backend

particularly 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.

TABLE II
THE 10 MOST ATTRACTIVE TOPICS

Score	Topic
69.6	jobs and career
67.1	teamwork problems
63.7	code readability
61.9	code refactoring
49.7	education and research
48.6	time management
48.4	choosing programming language
47.0	functional programming
45.9	version control systems
44.1	hardware and operating systems

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

Score	Topic
16.8	network and backend
19.3	sofware modeling
19.3	access control
19.4	user data collection
20.6	asynchronous program execution
21.0	Model-View-Controller architecture
21.6	recursion
22.0	software development business contract
22.8	web services and microservices
22.9	sorting algorithms

and falling trends.

A. Method

For each year, we calculated the relative frequency of every topic, using the LDA model data from the previous research questions. We divided the number of questions in a given year for which a given topic was the main one by the total number of questions in this year. Since the number of questions in the years 2008 and 2009 was too small to be relevant, and the year 2021 contained incomplete data, we included only the years 2010–2020 in the analysis.

Next, we marked the topics whose relative frequency was higher in 2020 than in 2010 as rising and the rest of them as falling. For both rising and falling trends, we found three topics with the highest difference between the maximum and minimum relative frequency during the analyzed years.

We also investigated the historical development of the three most frequent topics from section IV.

B. Results

In Fig. 1, there are plots of the three most rising trends: domain-driven design, asynchronous programming, and inheritance. The relative frequency of the most rising topic, domain-driven development, increased every year until 2018 but started to slightly decrease in the following years. As the domain-driven design is now almost a two-decade-old concept, this might mean the discussion about it starts losing intensity. However, it could also be only a temporary fluctuation.

Fig. 2 displays the plots of the three most falling trends: jobs and career, education and research, and software licensing.

For the first two mentioned topics, the relative frequency was extremely high in the first years – in 2010 more than 10% of questions concerned these topics. Then the frequency began to fall rapidly. We manually inspected a small sample of these questions, and we found that they are often closed as being off-topic. According to Software Engineering Stack Exchange rules, questions regarding career or education advice are considered off-topic because they are often meaningful only for a specific asker, and answers to them are almost always subjective. Licensing questions are also being closed since the website claims it does not aim to provide legal advice.

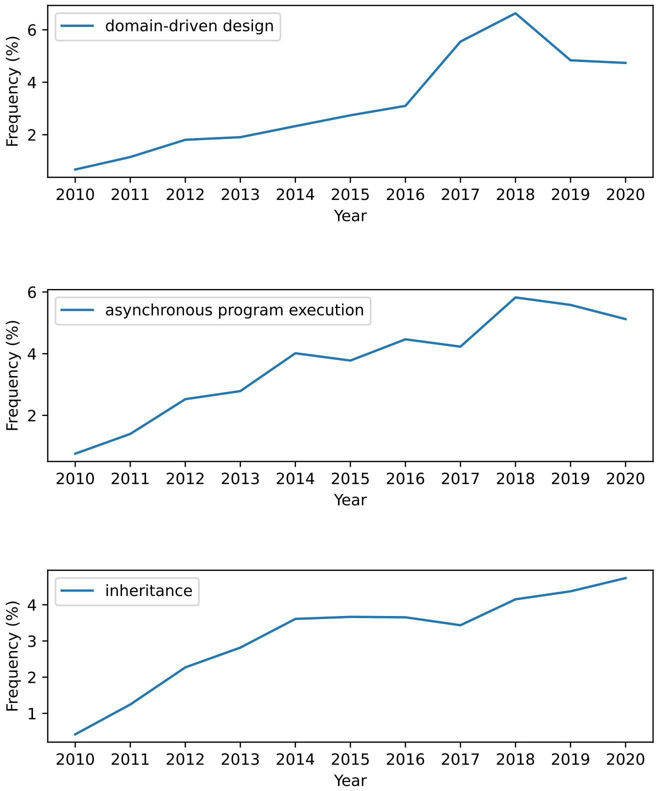

Fig. 1. The most rising topics.

Historical trends in the popularity of the three most frequent topics from section IV are displayed in Fig. 3. Database systems were rising until 2014, when they started to stagnate. The topic of quality assurance fluctuates, but it shows a generally positive trend. Finally, agile software development fell from about 6% to less than 2% over the decade.

VII. QUESTION COUNT

In RQ4, we would like to find whether the overall number of questions asked on Software Engineering Stack Exchange rises or falls over time. Furthermore, we will be interested in the historical development of the proportion of questions without an accepted answer. This will provide us insights about the health of the platform and its potential problems.

A. Method

Using a Stack Exchange Data Explorer query, we determined the total numbers of questions asked during each of the years from 2010 to 2020.

After a question is asked, its author should mark one of the answers as accepted in case of satisfaction, but in reality, many questions remain unresolved. Therefore, we also queried the counts of questions asked in the given years that do not have an accepted answer associated with them.

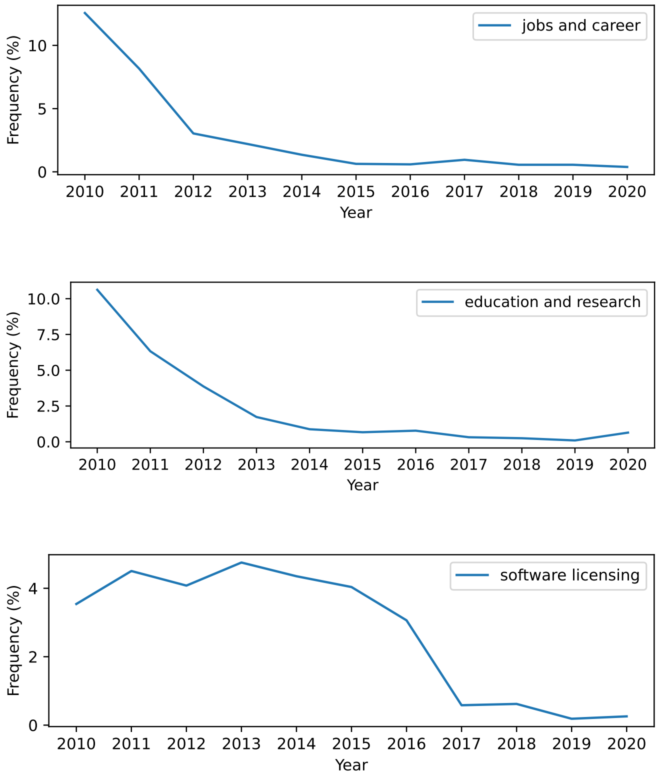

Fig. 2. The most falling topics.

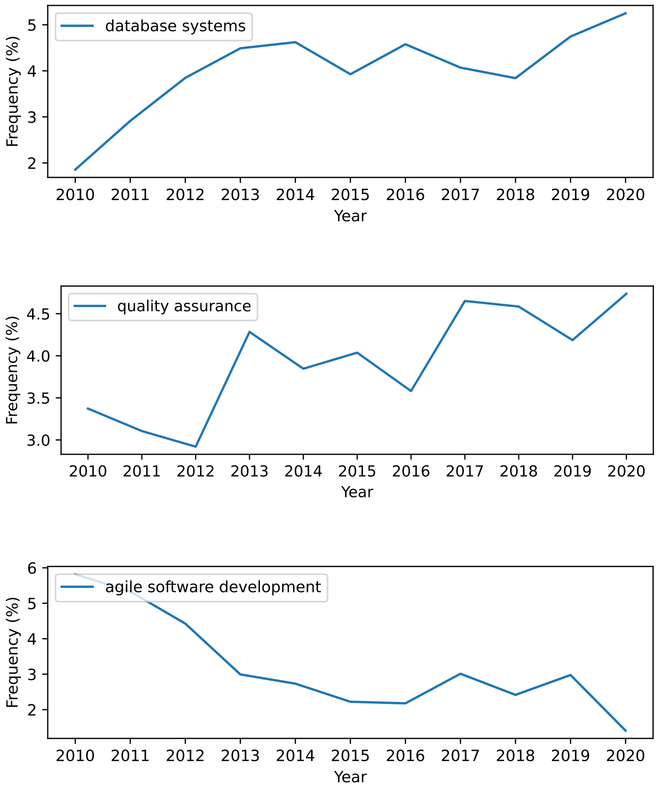

Fig. 3. The historical trends of the three most frequent topics.

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

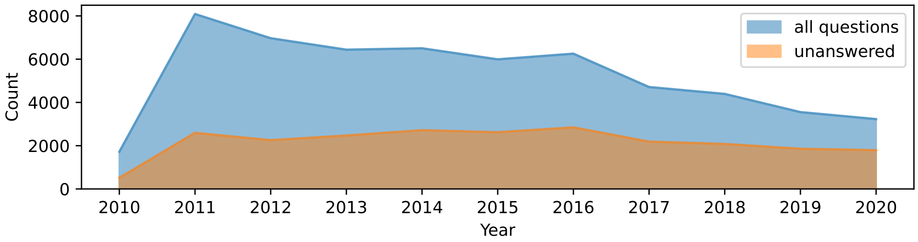

Fig. 4. The counts of all and unanswered questions asked in the given years.

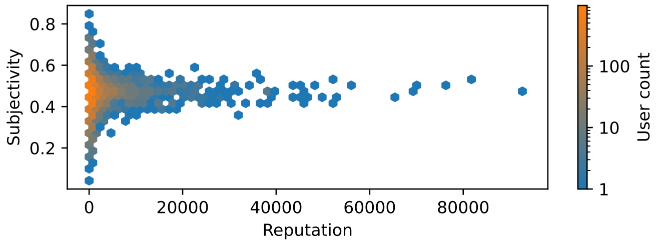

Fig. 5. The relationship between the users' reputation and the average subjectivity of their answers.

to be neither too positive nor too negative, but the average inclines slightly towards positivity. The computed Spearman's rank correlation coefficient is -0.01, i.e., there is no correlation between reputation and polarity.

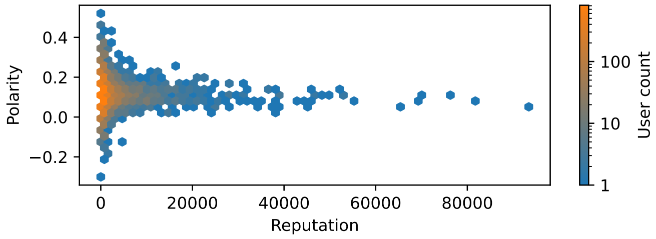

Fig. 6. The relationship between the users' reputation and the average polarity of their answers.

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

[6] https://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

[1] 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*, vol. 19, no. 3, pp. 619–654, 2014.

[2] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of Stack Overflow," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, 2013, pp. 97–100.

[3] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development: A study of Stack Overflow posts," ser. MSR '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 174–185.

[4] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about? A large-scale study using Stack Overflow," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '18. New York, NY, USA: Association for Computing Machinery, 2018.

[5] A. Verma, N. Sardana, and S. Lal, "Comprehensive analysis of trends in software engineering Q&A site," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2019, pp. 648–653.

[6] S. Baltes and C. Treude, "Code duplication on Stack Overflow," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, ser. ICSE-NIER '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 13–16.

[7] H. Zhang, S. Wang, T.-H. Chen, Y. Zou, and A. E. Hassan, "An empirical study of obsolete answers on Stack Overflow," *IEEE Transactions on Software Engineering*, vol. 47, no. 4, pp. 850–862, 2021.

[8] S. Beyer, C. Macho, M. Di Penta, and M. Pinzger, "What kind of questions do developers ask on Stack Overflow? A comparison of automated approaches to classify posts into question categories," *Empirical Software Engineering*, vol. 25, no. 3, pp. 2258–2301, May 2020.

[9] 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. New York, NY, USA: Association for Computing Machinery, 2014, pp. 112—-121.

[10] M. Tahaei, K. Vaniea, and N. Saphra, "Understanding privacy-related questions on Stack Overflow," ser. CHI '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–14.

[11] M. U. Haque, L. H. Iwaya, and M. A. Babar, "Challenges in Docker development: A large-scale study using Stack Overflow," in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ser. ESEM '20. New York, NY, USA: Association for Computing Machinery, 2020.

[12] Y. Kou and C. M. Gray, "A practice-led account of the conceptual evolution of UX knowledge," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–13.

[13] A. Kamienski and C.-P. Bezemer, "An empirical study of Q&A websites for game developers," *Empirical Software Engineering*, vol. 26, no. 6, pp. 1–39, 2021.

[14] A. K. McCallum, "MALLET: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[15] J. Porubän, M. Bačíková, S. Chodarev, and M. Nosáľ, "Teaching pragmatic model-driven software development," *Computer Science and Information Systems*, vol. 12, no. 2, pp. 683–705, 2015.

[16] S. Loria *et al.*, "TextBlob, version 0.16.0," 2020. [Online]. Available: https://textblob.readthedocs.io