The Journal of Systems and Software 215 (2024) 112089

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

# What is discussed about Flutter on Stack Overflow (SO) question-and-answer (Q&A) website: An empirical study[☆]

Afit Alanazi[1], Reem Alfayez[1],[*]

*King Saud University, College of Computer and Information Sciences, Riyadh, Saudi Arabia*

ARTICLE INFO

*Keywords:*
Stack overflow
Flutter
Mobile development
Topic modeling
Empirical study

ABSTRACT

**Context:** Mobile applications are growing in prevalence. Cross-platform application is one of the main challenges within mobile application development. Aiming to facilitate the process of mobile application development, the software engineering community developed multiple solutions to support cross-platform application development. Flutter is a recent cross-platform software development kit (SDK) that has been rising in popularity.
**Objective:** This study aims to gain a better understanding of Flutter stance on Stack Overflow (SO).
**Method:** The study identified and analyzed 176,876 Flutter-related questions to understand the interest towards Flutter. The study utilized Latent Dirichlet Allocation (LDA) to identify Flutter-related topics discussed within the identified questions. Subsequently, a number of heuristics were utilized to gauge the popularity and difficulty of topics.
**Results:** The study revealed that interest towards Flutter was steadily increasing on SO until it dropped in 2023 and SO users discuss 12 Flutter-related topics, with the topic of setup and build automation being the most popular and most difficult.
**Conclusion:** This study highlights the interest towards Flutter on SO, identifies Flutter-related topics discussed on the website, and assess the topics popularity and difficulty. Software researchers, practitioners, educators, and Flutter contributors may utilize the results to steer their future Flutter-related efforts.

## 1. Introduction

The popularity of mobile applications has been continuously increasing for the last few years (Rosen and Shihab, 2016). The number of mobile applications that are currently available for download on Google Play store[2] and Apple App Store[3] is over 5 million applications. A recent report on mobile application usage has revealed that in the third quarter of 2021 Apple App Store generated around 22 billion U.S. dollars and Google Play store revenue was over 12 billion U.S. dollars (Ceci, 2021). In addition, the report estimated that the revenue of mobile applications will be significantly growing in the near future; mobile applications are projected to generate a revenue of over 613 billion U.S. dollars by 2025.

Though the mobile application market is a promising market with tremendous opportunities, mobile application development is not trivial and it has many unique challenges (Rosen and Shihab, 2016; Minelli and Lanza, 2013; Linares-Vásquez et al., 2013; Joorabchi et al., 2013). One of the main challenges within mobile application development is the issue of cross-platform applications. Mobile applications are expected to be running seamlessly on multiple operating systems, such as Android and iOS (Windmill, 2020; Rosen and Shihab, 2016; Joorabchi et al., 2013). Therefore, developers must have their applications be compatible with several platforms.

Fortunately, the software engineering community has made tremendous efforts to facilitate the development of mobile applications, with an aim to simplify, speed, and decrease cost of developing mobile applications (Windmill, 2020). Specifically, the community has created many effective solutions including practices, technologies, and tools to support mobile application development (Windmill, 2020; Rosen and Shihab, 2016). Cross-platform development is one of these solutions that, once leveraged, enables developers to create cross-platform applications quickly and cost-efficiently. Cross-platform development, as

[☆] Editor: Raffaela Mirandola.
[*] Corresponding author.
*E-mail address:* reealfayez@ksu.edu.sa (R. Alfayez).
[1] Both authors contributed equally to this manuscript.
[2] https://play.google.com/store
[3] https://apps.apple.com/

https://doi.org/10.1016/j.jss.2024.112089
Received 21 August 2022; Received in revised form 28 February 2024; Accepted 3 May 2024
Available online 7 May 2024

the name suggests, allows users to build a single application that can be run on multiple platforms (Windmill, 2020).

Flutter[4] is an open-source software development kit (SDK) created by Google for cross-platform mobile application development (Windmill, 2020). Flutter was officially released in late 2018, with an aim to facilitate application development by allowing for developing multi-platform applications from a single codebase. Though Flutter is relatively young, it is becoming the most popular cross-platform mobile solution (Vailshery, 2022). Moreover, Stack Overflow (SO) Stack Exchange question-and-Answer (Q&A) website conducts an annual survey to understand trends and preferences of technologies utilized by developers. In the 2021 annual survey,[5] in which around 80,000 developers from around the globe participated, Flutter was ranked among the top-10 technologies that are most loved. In addition, the survey revealed that Dart, an object-oriented programming language created by Google that is used to build Flutter applications, was identified as one of the most commonly used programming languages; over 5,000 developers who worked with Javascript wanted to work with Dart; and Dart was found to be one of the highest-paid languages.

When reviewing the software engineering literature, it was revealed that SO has proved its effectiveness in gaining a better understanding of a variety of subjects within the software engineering field (Bajaj et al., 2014; Silva et al., 2021; Rosen and Shihab, 2016; Ahmed and Bagherzadeh, 2018; Haque et al., 2020; Abdellatif et al., 2020), due to the website being one of the largest and most popular online Q&A forums that allows developers to gain peer answers to their developments questions in addition to answer other users questions (Rosen and Shihab, 2016; Silva et al., 2021). Particularly, analyzing questions posted on the website proved to be beneficial in understanding the interest towards a given subject, topics that are discussed within the subject, popularity of these topics, and their difficulty (Rosen and Shihab, 2016; Silva et al., 2021; Haque et al., 2020).

With an aim to gain a better understanding of the stance of Flutter on SO, we conducted an empirical study that is focused on SO Flutter-related questions. The study began by identifying 176,876 Flutter-related questions on SO. Then it examined trends within these questions to understand the interest towards Flutter. Afterward, the study leveraged Latent Dirichlet Allocation (LDA), a topic modeling algorithm, to reveal Flutter-related topics that are discussed within the set of the identified questions (Blei et al., 2003). Subsequently, the study utilized a set of heuristics to determine the popularity and difficulty of identified topics, in addition to examining the relationship between popularity and difficulty.

The remainder sections of this paper are organized as follows: Section 2 provides background information for the study. Section 3 details the setup of the study. Section 4 presents and discusses the study's results, and Section 5 summarizes the implications of the study. Potential threats to the study's validity are addressed in Section 6. Section 7 presents an overview on related, previous studies. Lastly, Section 8 concludes the study.

## 2. Background

This section presents background information for this study. Summarized descriptions of Flutter, Stack Exchange Network and SO, in addition to LDA algorithm are provided below, aiming to make this study self-contained. It should be noted that the included overview of Flutter is primarily sourced from its official documentation.[6] and that the overview of Stack Exchange and SO is sourced from the Stack Exchange Network website.[7]

### 2.1. Flutter

Flutter is an open-source mobile SDK that was built by Google in 2015 and officially released in 2018, with an aim to ease, speed up, reduce cost of, and lower complexity of mobile application production across platforms. A core idea in Flutter is building applications through composing widgets. A widget is a Dart class with a view structure that can specify any aspect of an application's view including layout, styles, animations, and more (Payne, 2019; Windmill, 2020). There exists an expansive set of pre-built Flutter widgets that can be readily used by developers, such as Button, TextField, and Row. Developers can also create their own Custom Widgets if they find the pre-built widgets unfitting or lacking (Windmill, 2020).

Flutter provides a wide set of tools to support its developments, such as Flutter framework, widgets, testing framework, and debugging tools. It also provides support for many tools including Android Studio, IntelliJ IDEA, and VS Code.

Not only Flutter is used inside Google, but it is also used by other outside developers from around the globe. Since its inception, Flutter was used to build over 400,000 applications that have been shipped to many hundreds of millions of devices. Some examples of organizations that leveraged Flutter to build applications are: BMW, TOYOTA, Nubank, Alibaba Group, Philips, and more.

### 2.2. Stack Exchange Network and Stack Overflow (SO)

The Stack Exchange network consists of over 170 Q&A websites that target a wide-variety of subjects in different fields. Though each Stack Exchange Q&A website is focused on a specific subject, all of these websites are based on the same concept: receiving answers to one's questions.

A user can post a question and the question will be answered by other users. A question must include at least one tag (i.e., a keyword or label to categorize a question with other similar questions to facilitate the search process) and can have up to five tags, as a question can be related to different subjects. Moreover, users can comment on questions and answers. They can also upvote or downvote questions, answers, and comments based on their usefulness and appropriateness.

SO is the most popular, largest, and oldest Stack Exchange community website. As of Dec 06, 2023, the website has 21,628,228 users, 23,919,429 questions, 89,779,301 comments, and 35,378,646 answers. It was also estimated that over 5,000 questions are posted daily on the website and that the website receives over 7 million daily visits. The website was created in 2008, with an aim to provide professional and amateur programmers with a platform in which they can find and provide answers to technical challenges. The website contains a wide variety of coding questions and answers including but not limited to a specific development problem, software tool, and software algorithm.

### 2.3. Latent Dirichlet Allocation (LDA)

Topic modeling is an unsupervised statistical machine learning technique that aims to extract key topics from a large collection of text. One of the most commonly used topic modeling algorithms is LDA (Silva et al., 2021), which was originally proposed in 2003 by (Blei et al., 2003).

LDA has been widely utilized in the software engineering field to understand what developers are discussing about different subjects, due to the algorithm's efficiency and effectiveness (Silva et al., 2021). Given a set of SO questions, the algorithm clusters words in these questions into topics based on the frequency of the words occurrence. Afterward, for each question, the algorithm assigns it a probability score that indicates its relevance to each identified topic. The highest probability score of a topic indicates that a question is most likely to be relevant to the topic (Rosen and Shihab, 2016; Silva et al., 2021). The algorithm

[4] https://flutter.dev
[5] https://insights.stackoverflow.com/survey/2021
[6] https://docs.flutter.dev/
[7] https://stackexchange.com/

also outputs the top-10 keywords of each identified topic (Rosen and Shihab, 2016; Haque et al., 2020; Silva et al., 2021).

There are two main challenges that lie in the usage of the algorithm. The first challenge is determining the optimal number of topics, as the quality of extracted topics that are clear, segregated, and meaningful relies heavily on the selected number of topics. If one were to assign a small number of topics, the algorithm may introduce overlapping topics that are hard to analyze. In contrast, if one were to assign a large number of topics, the algorithm might produce extremely coarse-grained topics, making the analysis of topics challenging (Rosen and Shihab, 2016; Blei et al., 2003).

Unfortunately, the algorithm does not identify the optimal number of topics; instead, the number of topics is one of the inputs to the algorithm. To overcome this challenge, one can run multiple preliminary experiments in which they create models with a varying number of input topics. In each experiment run, the number of topics should be increased by a predefined increment, and the coherence score, which is an indicator of the resulting model quality, is calculated. Once the experiments has been concluded, the model that results in the highest coherence score can be selected as the optimal model, and therefore, its number of topics can be considered as the optimal number of topics (Rosen and Shihab, 2016; Haque et al., 2020; Han et al., 2020).

The second challenge is that the algorithm only clusters different questions into topics based on their similarity without providing a name, a human-understandable meaning, to these identified topics (Silva et al., 2021). To overcome such a challenge, one can utilize many manual methods that rely on humans to name these identified topics (Silva et al., 2021).

One method that one can utilize is open card sort (Fincher and Tenenberg, 2005), which has been utilized in previous, related work (Haque et al., 2020; Yang et al., 2016; Bagherzadeh and Khatchadourian, 2019). In open card sort, there is no predefined set of topic names. Instead, those who participate in sorting the topics need to provide a name for each topic and criteria by which the topics were named. Thus, one can utilize the top-10 keywords that is outputted by the algorithm for each topic in addition to a number of the most relevant questions to each topic to assign it a name that best explains keywords and questions of each identified topic (Haque et al., 2020). Afterward, one can select random questions from each topic to verify the suitability of the assigned name (Haque et al., 2020; Rosen and Shihab, 2016).

## 3. Study setup

This section presents the setup of the study: goal and research questions, utilized dataset, and approach.

### 3.1. Goal and research questions (RQs)

We define the goal of this empirical study by using the Goal-Question-Metric (GQM) template (Caldiera and Rombach, 1994), as follows: Analyze SO Flutter-related questions for the purpose of identifying and characterizing issues faced by Flutter developers with respect to issues' topics, popularity, and difficulty from the point of view of SO users in the context of SO Flutter-related questions.

With an aim to achieve the above-mentioned goal, we formulated the following three RQs:

#### 3.1.1. Interest

Though it was only officially released in 2018, Flutter has gained a growing popularity. The spiking popularity of Flutter is evidenced by ranking the SDK among the top-10 technologies that are most loved in the SO 2021 survey and the increasing number of developers who are currently utilizing the SDK or wish to utilize it in future projects (Windmill, 2020; Vailshery, 2022). As this study aims to have a better understanding of Flutter through the lens of SO, we seek to understand whether the spiking popularity of Flutter is also reflected on SO. Therefore, this study's first research question is: ***RQ1:How is the interest towards Flutter on SO?***

#### 3.1.2. Topics

Mobile application development has been found to be distinct from traditional software development and has its unique set of challenges (Rosen and Shihab, 2016; Minelli and Lanza, 2013). A previous study identified mobile development topics that SO users asked about in general, such as user interface and media (Rosen and Shihab, 2016). Given that the study was published prior to the introduction of Flutter and that the SDK aims to facilitate the mobile application development process, we seek to understand the challenges that Flutter developers are facing when utilizing the SDK and whether the usage of the SDK still requires overcoming some of the inherited challenges of mobile application development (Rosen and Shihab, 2016). Moreover, identifying how the set of the identified topics evolve over time can provide clarity on how Flutter users' interest towards topics is shifting. For example, one can assume that the number of questions that are related to setting up the SDK will be high in the early years of the SDK and it will drop in the following years. Therefore, this study's second research question is: ***RQ2: What Flutter-related topics are discussed on SO and how they evolve over time?***

#### 3.1.3. Topic characteristics

The study by (Rosen and Shihab, 2016) noted that some mobile development topics, such as application distribution; mobile tools; and user interface development, can be considered more popular while other topics, such as mobile APIs; device input; and HTML5/browser, can be considered more difficult. As this study aims to gain a better understanding on the stance of Flutter on SO, we seek to assess the popularity and difficulty of the identified topics to understand whether these patterns are also present on Flutter. Moreover, having an understanding of the popularity of identified topics can prove to be fruitful in clarifying topics that are of more interest to developers, and knowing which Flutter-related topics are more difficult can be valuable in determining topics that are most crucial and require more attention to be addressed. Additionally, understanding the monotonic association between popularity and difficulty can provide a more holistic understanding of their relationship. Consequently, the third research question of this study is: ***RQ3: What are the characteristics of the identified topics, in terms of popularity and difficulty, and how is popularity and difficulty associated?***

### 3.2. Dataset

The first step to answer this study's research questions was to acquire a dataset that contains SO website questions that are related to Flutter. To do as such, we utilized the Stack Exchange Data Dump.[8] The data dump consists of XML files that contain anonymized data of all content contributed by users on the Stack Exchange websites. The downloaded dump that this study utilized is the SO Stack Exchange Data Dump that includes data up until Dec 06, 2023.

To identify Flutter-related questions, we applied a search method that encompasses tag-based and content-based filterings, following the approach utilized by (Haque et al., 2020). Mainly, we searched for any question on the dataset that contains the term "Flutter" in its title, body, or one of its tags without letter case distinction. The search step resulted in the retrieval of 176,876 questions. To ensure that the retrieved questions are Flutter-related, the authors followed the guidelines of (Ralph et al., 2020) by selecting a random representative sample (384 questions based on 95% confidence interval) (Medhi, 1992). Then the two authors independently reviewed each selected question to determine its relevance to Flutter. Once the independent review has been concluded, the authors compared their results in a joint meeting. The meeting was initiated by calculating the Cohen's Kappa coefficient to measure the inter-rater agreement (Ralph et al.,

[8] https://archive.org/details/stackexchange

2020). The resultant Cohen's Kappa coefficient equaled 1.00, indicating perfect agreement between the two authors (Landis and Koch, 1977). Moreover, the review indicated that all selected questions are Flutter-related. As a result, all of the 176,876 retrieved questions were utilized to answer this study's research questions.

### 3.3. Approach

This section details the approach that we followed to answer each of the study's research questions.

#### 3.3.1. RQ1: How is the interest towards Flutter on SO ?

With the aim of gaining an understanding of the interest towards Flutter in mind, we calculated a number of heuristics. Specifically, we calculated the frequency of asked Flutter-related questions, number of unique users who posted a Flutter-related question, and total number of views of each Flutter-related question, following the approach of (Rosen and Shihab, 2016).

The total number of Flutter-related questions can provide an approximate measure on the interest towards Flutter in general (Rosen and Shihab, 2016). The number of unique users who posted Flutter-related questions was calculated to gain an insight on whether Flutter-related questions are typically asked by few SO users who frequently ask or by a diverse, large group of users (Rosen and Shihab, 2016). We emphasize that the number of unique users counts each user who asked a Flutter-related question once. In other words, if a user posted a Flutter-related question in a year then said user posted another question in the same year or a following year, the user was only included once in the number of unique users who posted a Flutter-related question in a year; the first year the user posted a Flutter-related question.

Furthermore, we expanded the scope of our analysis by retrieving questions related to the following cross-platform SDKs/Frameworks[9] : Ionic, Kotlin, NativeScript, React Native, and Xamarin. Subsequently, we compared the frequency of Flutter-related questions and unique users who posted such questions with those pertaining to the aforementioned SDKs/Frameworks.

The total number of views was also calculated to measure the interest towards Flutter on SO. The number of views provides a unique insight into the interest towards Flutter, as the number of total views includes both registered and unregistered users within SO (Rosen and Shihab, 2016), which is particularly valuable given the fact that on SO the number of viewers who are not registered is higher than the number of registered users (Mamykina et al., 2011). We underscore that the provided information on the number of views each SO question received in the XML dump only includes the total count of views, without any distinction between registered and unregistered users nor the inclusion of number of views per year.

#### 3.3.2. RQ2: What Flutter-related topics are discussed on SO and how they evolve over time?

With an aim to identify Flutter-related topics that are discussed on SO, we followed the approach illustrated in Fig. 1. All acquired Flutter-related questions were analyzed using LDA to model them into topics (Blei et al., 2003). Preceding the application of LDA, we preprocessed the acquired set of questions to eliminate any noise that might impact the quality of the resulting LDA model (Barua et al., 2014). The preprocessing step was initiated through removing all of HTML elements, HTML tags, and URLs. Afterward, NLTK corpus was utilized to remove all stop words, such as "are" and "is" (Bird et al., 2009). Subsequently, words were reduced to their respective base words using the spaCY Lemmatizer,[10] such as the reduction of "animation" to "animate". Once the preprocessing step has been concluded, a bigram model was built using Gensim (Ř.ehůřek and Sojka, 2010), as bigram models proved their effectiveness in improving the quality of textual processing (Haque et al., 2020). Afterward, the LDA was applied to identify topics that are discussed within the obtained Flutter-related questions. To determine the optimal number of topics, we followed previous, related studies' approach by conducting multiple preliminary experiments (Rosen and Shihab, 2016; Silva et al., 2021; Haque et al., 2020). The experiments started by inputting 2 as a number of topics and incrementing the number of topics by 2 in each subsequent experiment run and calculating the resulting coherence score, a measurement of the resulting LDA model quality (Rosen and Shihab, 2016). The experiments were concluded after inputting 50 topics, due to the coherence score drastically decreasing. The highest obtained coherence score was .64, indicating that Flutter-related questions can be clustered into 12 topics.

To name the identified topics, we utilized a manual approach that was followed by previous, related work (Haque et al., 2020; Yang et al., 2016; Bagherzadeh and Khatchadourian, 2019). Particularly, we utilized open card sort (Fincher and Tenenberg, 2005). In open card sort, there is no predefined set of topic names. Instead, those who participate in sorting the topics need to provide a name for each topic and criteria by which the topics were named. Thus, for each identified topic, the two authors reviewed its top-10 keywords and its most-50-relevant questions (i.e., questions with 50 highest probability of belonging to topic) to assign it a name. Then the two authors selected a random, representative sample of each topic and verified the suitability of the assigned names (Haque et al., 2020; Rosen and Shihab, 2016).

Afterward, for each identified topic, we calculated the total number of questions posted in each year. Then we plotted the results in a trend line graph to observe any trends within the evolution of topics over time.

#### 3.3.3. RQ3: What are the characteristics of the identified topics, in terms of popularity and difficulty, and how is popularity and difficulty associated ?

We adopted a number of heuristics that were utilized in previous, related studies to measure popularity and difficulty of topics (Rosen and Shihab, 2016; Haque et al., 2020; Han et al., 2020). Precisely, to assess the popularity of a topic, we calculated the average number of view and score counts its questions received. The average number of views provides an insight towards the interest of both unregistered and registered users within a given topic (Rosen and Shihab, 2016). Moreover, the score of a question results from totaling the upvotes and downvotes said question has received from SO users, which can serve as an indicator of registered users preference for a topic (Haque et al., 2020).

The difficulty of a Flutter-identified topic was measured through the means of two heuristics that were utilized in previous, related studies (Rosen and Shihab, 2016; Haque et al., 2020; Han et al., 2020). Specifically, for each identified topic, we calculated the rate of questions that did not receive an accepted answer. An answer to a question can be accepted by the user who posted the question, indicating the user's satisfaction with the received answer (Rosen and Shihab, 2016). Additionally, for each identified topic, we calculated the median time elapsed in seconds from the posting of each of its questions until the receiving of its accepted answer, if any. The decision to rely upon the median as a measure of difficulty was due to long-latency answers. Particularly, the median was utilized to compensate for potential skewing of the mean resulting from long-latency answers, which was also followed by (Abdellatif et al., 2020; Rosen and Shihab, 2016; Haque et al., 2020).

Lastly, we applied the Kendall correlation test to examine the relationships among each popularity and difficulty heuristics. The decision was made to utilize Kendall correlation test due to the test being less sensitive to outliers and its ability to produce more robust results (Puth et al., 2015; Abdi, 2007).

[9] https://kotlinlang.org/docs/cross-platform-frameworks.html
[10] https://spacy.io/api/lemmatizer/

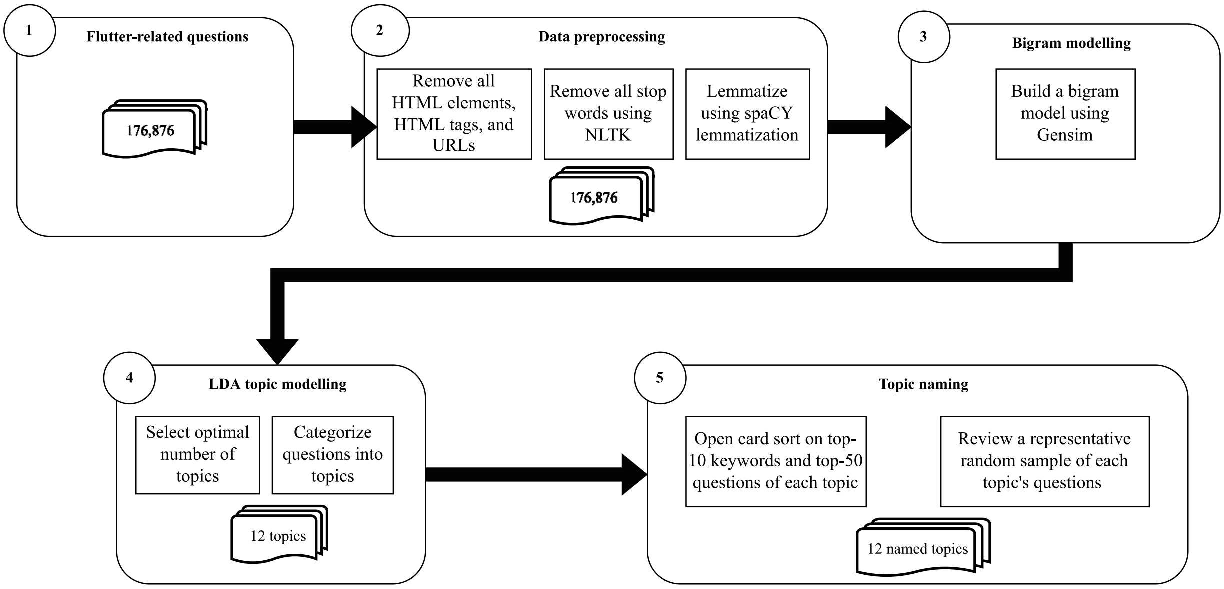
1
Flutter-related questions
176,876
2
Data preprocessing
Remove all HTML elements, HTML tags, and URLs
Remove all stop words using NLTK
Lemmatize using spaCY lemmatization
176,876
3
Bigram modelling
Build a bigram model using Gensim
4
LDA topic modelling
Select optimal number of topics
Categorize questions into topics
12 topics
5
Topic naming
Open card sort on top-10 keywords and top-50 questions of each topic
Review a representative random sample of each topic's questions
12 named topics

**Fig. 1.** Overview of topic identification approach.

## 4. Results and discussion

This section presents and discusses the findings of this study. The results are based on the aforementioned resultant set of 176,876 Flutter-related questions retrieved from SO.

### *4.1. RQ1: How is the interest towards Flutter on SO ?*

**Frequency of questions:** The number of questions related to each examined SDK/Framework asked on SO for each year was calculated and compared. Fig. 2 depicts the yearly trend of these questions. While the number of Flutter-related questions in 2017 was relatively low, there has been a consistent increase in questions since then. However, there was a noticeable decline in activity in 2023. In comparison to other SDKs/Frameworks, Flutter-related questions exceeded all others in 2021 and 2022. However, in 2023, Typescript had more questions.

We also calculated the percentage of Flutter-related questions to questions related to the examined SDKs/Frameworks in each year. The percentages we found are as follows: 1.04%, 8.20%, 18.21%, 27.15%, 30.89%, 32.28%, and 30.36% in 2017, 2018, 2019, 2020, 2021, 2022, and 2023, respectively. While the percentage was extremely low in 2017, it substantially increased in subsequent years until it slightly dropped in 2023.

In terms of the percentage of Flutter-related questions to the total number of questions that were asked on SO for each year, the percentages are as follows: 0.04%, 0.40%, 1.15%, 2%, 2.57%, 3.04%, and 3.15% in 2017, 2018, 2019, 2020, 2021, 2022, and 2023, respectively. Though the resulting percentage in 2017 is quite small, the percentages on the subsequent years were steadily increasing.

**Number of users:** It was revealed that 65,498 unique users have asked at least one Flutter-related question. The percentage of unique SO users who asked at least on Flutter-related question to the unique users who asked at least one question that is related to the examined SDKs/Frameworks is 19.56%. Moreover, the percentage of users who asked at least one Flutter-related question to the number of unique SO users who asked at least one question on any subject is 1.06%, and the percentage of unique SO users who posted at least one Flutter-related question in comparison to the number of total unique users on SO is 0.30%.

Fig. 3 depicts the yearly distribution of SO unique users who asked their first question related to each examined SDK/Framework. As the graph illustrates, the number of new users who asked a Flutter-related question steadily increased over the years before showing a slight decline in 2022 and a drop in 2023. Moreover, starting from 2020, the number of Flutter unique questioners surpassed those related to other SDKs/Frameworks, with the exception of TypeScript, which consistently maintained a larger unique questioners.

Besides exploring the number of users who posted Flutter-related questions, we examined the number of Flutter-related questions each user asked. The maximum number of questions asked by one user is 273, the mean is 2.7 questions per user, and the median is 1 question. Moreover, of the 65,498 SO users who posted a Flutter-related question, around 61.89% posted only one Flutter-related question, 15.52% posted twice, 13.65% posted between three and five questions, 5.30% posted between six and ten questions, and only 3.64% posted more than ten questions.

**Number of views:** Flutter-related questions were found to have a mean of 2,461 views per question. The mode and median of views are 40 and 404, respectively. The maximum number of views a Flutter-related question received is 958,526, and the minimum is 2 views.

> Flutter-related questions are present on SO, have been steadily increasing through the years, and asked by a yearly increasing number of unique users. However, there was a decline in both number of questions and unique users in 2023. These questions received a median of 404 views.

The aforementioned results revealed that interest towards Flutter on SO has been generally increasing since the early stages of the SDK. Given that Flutter can be considered a young SDK, it was unexpected to find that Flutter was discussed on SO in some capacity prior to its official, public release in 2018. Since 2017, interest and discussions revolving around Flutter on SO have continuously grown, with a peak in 2022. Although the number of Flutter-related questions dropped in 2023, the percentage of these questions relative to the total questions posted yearly on SO continued to increase. Moreover, the share of Flutter-related questions compared to the other examined cross-platform SDK/Framework questions grew over the years, with a slight decline in 2023.

### *4.2. RQ2: What Flutter-related topics are discussed on SO and how they evolve over time?*

The approach to RQ2 resulted in the identification of 12 topics. Table 1 presents all 12 topics, their associated top-10 keywords, and

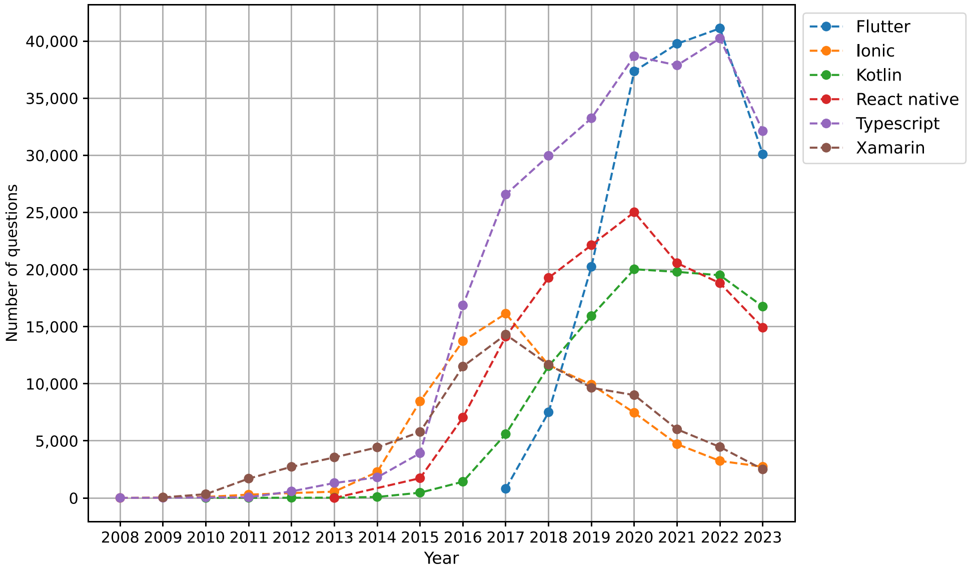

**Fig. 2.** Annual distribution of questions.

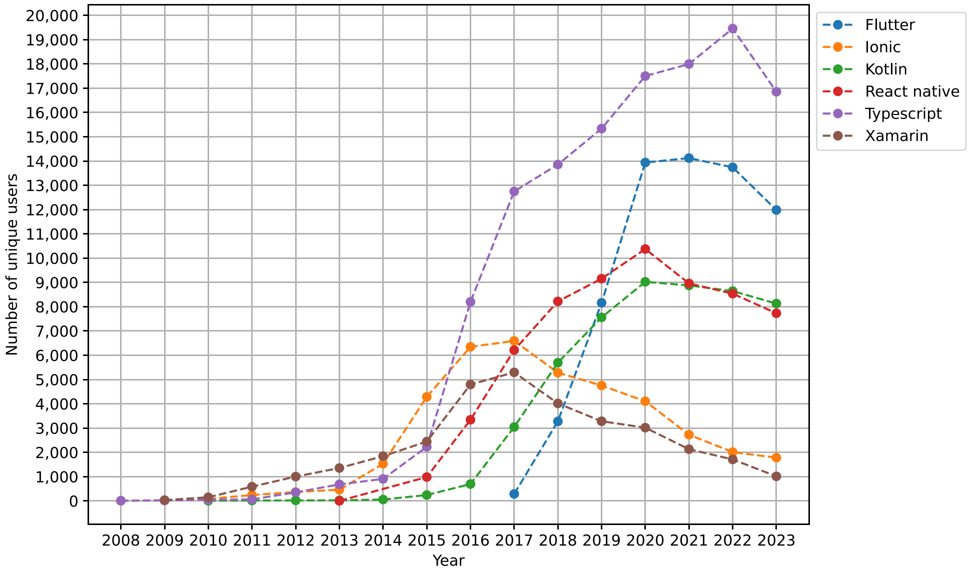

**Fig. 3.** Annual distribution of new users who asked a question.

their frequency. Below, we describe each topic in more detail, sorted based on decreasing frequency, and we provide a trend line analysis of these topics.

**Setup and build automation**: The topic of Flutter setup and build automation was found to include 22,765 questions (i.e., 12.87%). Questions within this topic typically seek guidelines and assistance in properly configuring Flutter, its dependencies, or associated plugins. This may involve utilizing dependency management or automated build tools, or it may pertain to manual setup processes. Questions also include users posting about an error that they encountered as a result of their setup, an error resulting from updating the SDK version, or an error resulting from changing the environment on which the SDK is running (e.g., switching from Windows to Mac). Examples of such questions are: "android app fails to build after upgrading to flutter 3, error class is defined multiple times ", "Flutter Desktop Windows says 'No supported devices found' ", and "Flutter Does Not Recognize Android SDK Build-Tools 25.0.3". Furthermore, questions also include users seeking solutions to issues encountered during automated build processes or while setting up automated build tools, as exemplified by "Flutter Maven error while trying to download jars from Android studio for Azure Notification Hub" and "assembleDebug keeps on failing".

**Layout**: The layout mechanism of Flutter relies on widgets, which form the core of the framework. In Flutter, everything is considered a widget, including the arrangement of grids, rows, and columns (Payne, 2019; Windmill, 2020). These layout widgets enable users to construct a structural interface, and a layout is created by composing widgets to build more complex structures (Windmill, 2020; Miola, 2020). Layout widgets and their proper construction was the topic of 21,301 questions (i.e., 12%). Examples of question in this topic can be of user asking how to compose a specific widget to properly construct a layout, such as "Flutter - Insert a Listview between two fixed containers". Another example of questions is users inquiring about how to fix a layout issue that they are facing, such as "Uneven borderRadius in ClipRrect - Flutter".

**Routing and navigation**: 17,685 questions (i.e., 10%) were found to be related to routing and navigation on Flutter. Questions in this topic discuss how one can navigate between routes (i.e., pages and screens) properly and resolve issues that were faced when attempting to navigate. Questions pertain to varying aspects of routing and navigation, such as the Navigation widget,[11] which is a Flutter widget that facilitates navigation between routes using a stack discipline that stacks a user actions and aids in navigating back from a route to the most recent previously visited route. Additionally, questions include inquiries regarding the Router, which ensures synchronization with the address bar on web platforms and manages deep links on Android and iOS. Questions within this topic range from seeking to attain a certain navigation goal to seeking a solution for an issue that was encountered while attempting to implement a navigation goal. “Flutter pop until reaching modal bottom sheet” and “How to close a screen from PageView class flutter” are examples of questions included in this topic.

**Networking**: Networking in Flutter refers to the process of establishing and managing communication channels between a Flutter application and an API or remote server over a network to enable applications to interact with external services and exchange data over. This topic is the center of 17,338 questions (i.e., 9.80%) that include multiple networking protocols, such as WebSockets and HTTP. These questions vary in complexity, with questions ranging from simple queries, such as “Check whether there is an Internet connection available on Flutter app?” and “How to get device MAC address in Flutter DART?”, to more complex ones, such as “Unable to connect with Mqtt using wss protocol ” and “HTTP and HTTP + Payload or SSL and SSL + Payload with Openvpn tunnel for Flutter”.

**Animation and canvas**: One of Flutter’s strengths is its ability to create custom animations, primarily facilitated through the Canvas widget, which communicates to Flutter what to paint on each pixel (Windmill, 2020). Animation and canvas are the central topics of 14,593 questions (i.e., 8.25%). In these questions SO users asked about the various nuances related to animations such as AnimationController, Ticker providers, AnimatedWidget, and more. Examples of these questions are: “how can I make animation in flutter when filling specific (path)?” ,“How to animate a widget by non-linear path?”, and “Flutter: AnimatedContainer - children widgets’ properties are NOT animating ”.

**Best practices and experiences:** The topic of best practices and experiences encompasses 14,249 questions (i.e., 8.06%). In this topic, SO users are not focusing on a particular Flutter-related topic; instead they seek insights and advice from other users based on the users’ collective experiences and practices. Questions within this topic span a wide spectrum, ranging from inquiries about optimal learning paths, such as “What Is The Best Order To Learn Beginner To Advanced Flutter Material?” to seeking opinions on design decisions, such as “Deciding between a Shared Folder and a Package for Shared Logic in Clean Architecture”. Users also seek advice on effective tactics to implement a design, such as “What’s the best way to implement this design layout?”. Additionally, queries extend to financial considerations, as exemplified by “Firebase Realtime Database billing for unauthorized requests, especially for overheads”.

**Database**: A total of 14,117 questions (i.e., 7.98%) were categorized under the database topic. Questions in this topic typically stem from challenges in persisting data by storing it into a database. Though questions contain varying type of databases, such as SQLite; Hive; Moor; and ObjectBox, we found that the majority of questions have referenced Firebase or Firestore (i.e., Firestore is Firebase’s newest database for mobile applications development). In fact, 58.14% of questions in this topic included “Firebase” or “Firestore” in its title, body, or one of its tags without letter case distinction. Questions in this topic can include an inquiry on how to achieve a specific goal or resolve an error that was encountered when attempting to achieve a specific goal. Examples of such questions are “How to make sqflite database faster?”, “How to cache query and mutation from GraphQL in Flutter?”, and “Query Date from Cloud Firestore and run two separate queries”.

**State management**: A state in Flutter refers to the information that determines how a stateful widget appears on a screen and how it responds to user actions; a stateful widget is a widget that can hold state and update its appearance in response to changes in that state. Flutter provides a wide range of stateful widgets, such as Button, TextField, Checkbox, and Slider widgets. Managing the state of stateful widgets is the topic of 13,086 questions (i.e., 7.40%), where users inquire about various aspects of state management, seeking guidance on achieving specific states or resolving issues encountered during implementation. These questions span a range of scenarios, from basic state manipulation to complex state interactions. Examples of such questions are “How to change the state of the checkbox in the dropdown list?”, “Change Between two Stateful Widgets while staying in the Same Location using a ToggleButton in Flutter ”, and “How do I disable a button if another button is clicked?”.

**Assets and media**: In the context of Flutter, assets (i.e., resources) refers to files that are bundled, deployed with an application, and accessible at runtime (Payne, 2019; Windmill, 2020). Assets files may include, but not limited to, JSON files; images; configuration files; and media files, such as videos and audios. Assets and media and how to properly handle them was the topic of 13,081 questions (i.e., 7.4%). Questions in this topic revolve around specifying, loading, bundling, testing, and displaying assets and media. Questions can include users asking how to achieve one of the previous actions or reporting on an error that was faced while performing one of the former actions, with an aim to know the cause of the error and how to overcome it. “keep getting this error when i try to load an image asset”, “Flutter - How to save audio as a file?”, and “how to play a video from a file using flutter? " are example questions related to this topic.

**Data mapping and conversion**: Data mapping and conversion was the topic of 12,459 questions (i.e., 7.04%). Questions in this topic revolve around how one can properly map different types of data, such as JSON and XML files, to Flutter objects or widgets. The questions can also involve asking how one can convert a data type to another type or a widget, such as converting a list to a ListView. One possible common source of data is a Future, an asynchronous computation that will eventually be completed with the requested data or an error, and users are asking how to handle Future returned data effectively. “ Flutter: Mapping JSON into List of Objects returns null”’ and “How to Convert String Date to Date Time??” are examples of questions in this topic.

**Authentication and authorization**: A total of 10,791 questions (i.e., 6.1%) were found to be discussing authentication and authorization in Flutter. Questions can include users asking about how to achieve a goal, possible causes of an encountered error and how to fix it, validity of a decision, or implications of a change that was made on an application’s security in terms of authentication and authorization. Examples of questions that were included in this topic are: “Is it okay to use an OAuth2 authorization code as proof of a successful login?”, “Appwrite with Flutter, Google OAuth provider issue”, and “How to update user data with uid in firestore in flutter”. Though the set of questions in this topic was not limited to Firebase, the majority of questions were discussing authentication and authorization in Flutter in the context of Firebase or Firestore; 37.3% of questions that were included in this topic have included “Firebase” or “Firestore” in its title, body, or one of its tags without letter case distinction.

**Unknown errors**: A total of 5,411 questions (i.e., 3.06%) were found to be related to the unknown errors topic. In this topic, users are posting about unknown errors that they encountered while utilizing Flutter. Questions can include users describing what they were exactly working on, the expected behavior, and the unexpected behavior that they encountered. In some instances, users had also provided the error messages that they received, and in other instances they indicated that they did not receive any error message. Users who are posting such questions are seeking to know what might be the possible cause of the encountered error and how to resolve it. Examples of questions under

[11] https://api.flutter.dev/flutter/widgets/Navigator-class.html

**Table 1**
Identified Flutter-related topics, associated top-10 keywords, and frequency.

| Topic name | Keywords | Frequency |
|---|---|---|
| Setup and build automation | build, android, version, error, project, gradle, fail, user, io, app | 22,765 |
| Layout | color, child, text, icon, padding, style, const, edgeinset, container, textstyle | 21,301 |
| Routing and navigation | context, key, extend, class, buildcontext, return, widget, title, state, text | 17,685 |
| Networking | message, error, response, print, request, api, notification, body, application, firebase | 17,338 |
| Animation and canvas | child, widget, size, height, container, width, context, listview, return, position | 14,593 |
| Best practices and experiences | create, test, function, code, question, variable, store, method, work, expect | 14,249 |
| Database | datum, return, snapshot, future, document, instance, builder, collection, database, stream | 14,117 |
| State management | app, work, screen, change, time, button, update, code, problem, set | 13,086 |
| Assets and media | image, file, asset, path, add, code, load, await, video, erro | 13,081 |
| Data mapping and conversion | list, string, map, final, item, int, type, class, require, return | 12,459 |
| Authentication and authorization | user, final, string, false, return, email, provider, true, null, await | 10,791 |
| Unknown errors | dart, package, import, src, widget, error, framework, main, material, object | 5,411 |

this topic are: “Flutter ticker leaking for no reason”, “Errors running a flutter application on the android studio’s virtual emulator, what does it mean?”, and “Dart Analysis server has terminated?”

> Flutter-related questions on SO were found to be revolving around 12 topics, which are as follows, based on decreasing order of frequency: setup and build automation, layout, routing and navigation, networking, animation and canvas, best practices and experiences, database, state management, assets and media, data mapping and conversion, authentication and authorization, and unknown errors.

The presence of some of the topics was expected due to the innate challenges within these topics not only within mobile application development, but in software development in general, such as authentication and authorization (Yang et al., 2016; Davis, 2005). Other topics were also expected to appear in Flutter-related questions, as these topics are generally considered to be common challenges in mobile application development. An example of such topics is layout, which was also found to be a common topic that is included in mobile developers discussions by previous, related studies (Rosen and Shihab, 2016; Beyer and Pinzger, 2014; Linares-Vásquez et al., 2013). On the other hand, it was unexpected to find that SO users were discussing some of the Flutter-related topics, such as routing and navigation, given that these topics were included on the official Flutter documentation that includes extensive Cookbook [12] (i.e., collection of recipes that demonstrate how to solve common problems encountered when writing Flutter applications), examples, video tutorials, and more.

Moreover, the trends of these topics over the examined years are presented in Fig. 4. As the trend line graph depicts, the trends of these topics are pretty similar; all of the identified topics were steadily increasing over time up until 2020. However, in 2021, the database topic began a declining trend. State management questions also decreased in 2021, then rose in 2022, only to fall again in 2023. Similarly, the number of setup and build automation questions experienced a slight decline starting in 2022. Additionally, all other topics showed a decline in 2023. Remarkably, both the topics of setup and build automation and layout were among the top two topics in most of the examined years.

### *4.3. RQ3: What are the characteristics of the identified topics, in terms of popularity and difficulty, and how is popularity and difficulty associated?*

The obtained statistics on the popularity of each identified Flutter-related topic are presented in Table 2. As the table depicts, the topic of setup and build automation has received the highest average number of views, followed by animation and canvas, layout, unknown errors, state management, routing and navigation, assets and media, best practices and experiences, networking, data mapping and conversion, authentication and authorization, and database. In terms of average score of identified topics, questions that address setup and build automation were also found to have the highest average score, followed by animation and canvas, state management, unknown errors, best practices and experiences, routing and navigation, layout, networking, assets and media, authentication and authorization, data mapping and conversion, and database.

**Table 2**
Popularity of identified topics.

| Topic name | AVG(views) | AVG(score) |
|---|---|---|
| Setup and build automation | 3,197.46 | 2.67 |
| Layout | 2,974.67 | 1.89 |
| Routing and navigation | 2,584.58 | 2.01 |
| Networking | 2,007.88 | 1.64 |
| Animation and canvas | 2,991.23 | 2.51 |
| Best practices and experiences | 2,069.74 | 2.20 |
| Database | 1,388.47 | 0.95 |
| State management | 2,834.51 | 2.43 |
| Assets and media | 2,232.74 | 1.59 |
| Data mapping and conversion | 1,926.23 | 1.08 |
| Authentication and authorization | 1,821.96 | 1.36 |
| Unknown errors | 2,939.13 | 2.25 |

Based on the results mentioned above, one can conclude that the topic of setup and build automation is the most popular topic among the identified Flutter-related topics discussed on SO, followed by animation and canvas, based on average view and score counts.

> The topic of setup and build automation was found to be the most popular topic, and the topic of animation and canvas followed as the second most popular topic, based on average view, score, and favorite counts.

The percentage of questions without an accepted answer and the median time elapsed to receive an accepted answer in seconds of each identified topic are displayed in “% w/o acc answer” and “Med time” columns in Table 3, respectively. It can be observed that setup and build automation was found to have the highest rate of questions that did not receive an accepted answer, followed by networking, assets and media, state management, unknown errors, best practices and experiences, authentication and authorization, animation and canvas, database, routing and navigation, data mapping and conversion, and layout.

In regards to median time elapsed to receive an accepted answer, setup and build automation was also found to have the highest median time to receive an accepted answer, followed by networking, assets and media, unknown errors, animation and canvas, state management, best practices and experiences, authentication and authorization, database, routing and navigation, data mapping and conversion, and layout.

Based on the aforementioned results, it can be concluded that setup and build automation is the most difficult Flutter-related topic among

[12] https://docs.flutter.dev/cookbook/navigation

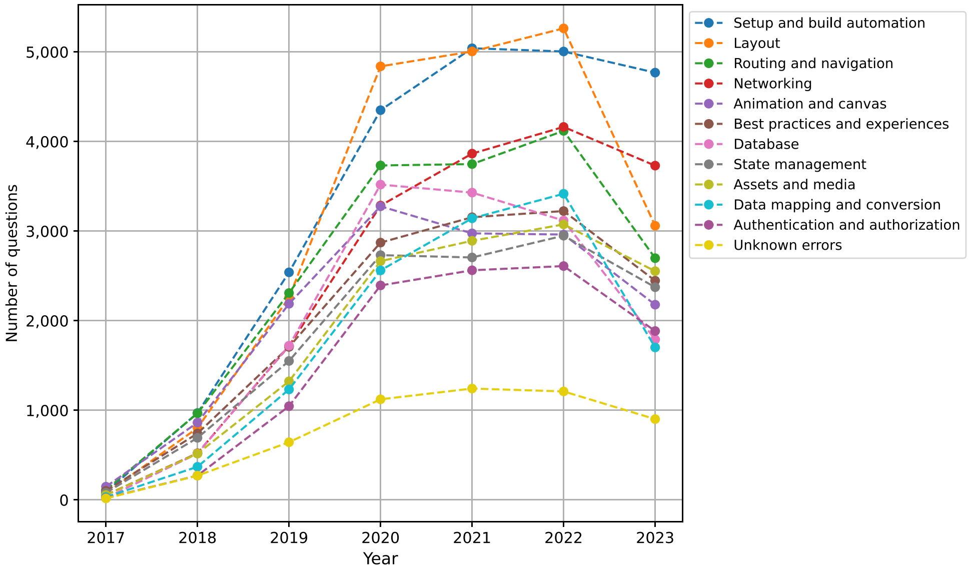

**Fig. 4.** Trend line of identified topics.

**Table 3**
Difficulty of identified topics.

| Topic name | % w/o acc answer | Med time |
|---|---|---|
| Setup and build automation | 0.72 | 16,053 |
| Layout | 0.50 | 2,464 |
| Routing and navigation | 0.54 | 3,100 |
| Networking | 0.71 | 11,399 |
| Animation and canvas | 0.57 | 4,608 |
| Best practices and experiences | 0.60 | 4,122 |
| Database | 0.57 | 3,360 |
| State management | 0.65 | 4,169 |
| Assets and media | 0.67 | 5,828 |
| Data mapping and conversion | 0.51 | 2,519 |
| Authentication and authorization | 0.59 | 3,916 |
| Unknown errors | 0.64 | 5,118 |

the identified Flutter-related topics discussed on SO, based on rate of questions without an accepted answer and median time required to receive an accepted answer. Additionally, networking can be considered as the second most difficult topic, while assets and media may be considered as the third, given the rate of questions that did not receive an accepted answer and the median time elapsed to receive an accepted answer.

> The topic of setup and build automation was found to be the most difficult topic, followed by networking and the topic of assets and media, based on rate of questions without an accepted answer and median time elapsed to receive an accepted answer.

We also examined the monotonic association between each popularity and difficulty heuristic using Kendall rank correlation. The examination of association resulted in P-values less than .01 in all instances, suggesting statistically significant associations. The resulting correlation coefficients are present in Table 4. As the table presents, all of the obtained correlation coefficients are pretty small (i.e., less than or equal to .2), which suggest a negligible correlations. Hence, one cannot assume any correlations among popularity and difficulty heuristics based on obtained results (Schober et al., 2018; Hubert and Rousseeuw, 2010).

The above results have revealed that the topic of setup and build automation can be considered both the most popular and the most difficult topic. While it was foreseeable that the topic would be popular given the young age of the SDK, finding it as the most difficult was unexpected, considering the wealth of materials provided in the Flutter official documentation that aim to facilitate and guide the SDK setup.[13] Moreover, the examination of the monotonic associations between popularity and difficulty heuristics resulted in very low correlation coefficients, indicating negligible associations between popularity and difficulty.

**Table 4**
Correlation coefficients of popularity and difficulty heuristics.

| | Views | Score |
|---|---|---|
| % w/o acc | 0.18 | 0.1 |
| Time to acc | 0.1 | 0.1 |

## 5. Implications

This study identifies and analyzes Flutter-related questions obtained from SO. This section summarizes how software engineering researchers, practitioners, educators, and Flutter contributors can leverage the findings of this study.

Software engineering researchers can utilize the findings of this study to steer their future research efforts related to Flutter. For example, the study revealed that the topic of setup and build automation remains one of the most difficult topics, regardless of how Flutter aimed to ease it in their applications by providing support for preexisting build tools, such as Gradle, and working examples in the SDK official documentation.[14] Researchers can focus their future efforts to pinpoint the exact issues developers are facing with setup and build automation in Flutter, where solutions can then be applied. Researchers can create new solutions that fill in the gaps that the current ones lack or improve upon the currently available solutions.

Software engineering practitioners can gain knowledge regarding Flutter-related topics that are discussed on SO and the characteristics of these topics. Having prior knowledge about challenges that might be faced when using Flutter can aid practitioners in anticipating issues that might be encountered to avoid them or be proactive about them. For example, practitioners can anticipate some issues when setting up the SDK, they can also precheck the compatibility of the SDK with their preexisting solutions and tools. Moreover, layout was one of the

[13] https://docs.flutter.dev/get-started/install

[14] https://docs.flutter.dev/add-to-app/android/plugin-setup

most frequent topics in all the studied years, and the topic has a relatively high accepted answers rate. Practitioners can review questions related to the topic to identify the trends and techniques that are utilized to handle layout in Flutter to consider including them in their applications. Additionally, the awareness of the most difficult Flutter-related topics can help practitioners realize that these difficulties are actually common among Flutter developers. As a result, practitioners can utilize SO to learn about other practitioners' solutions, opinions, and experience when having to deal with similar challenges, such as setup and build automation in addition to networking.

Software engineering educators can utilize the results of this study to hone their curricula that are related to mobile development. For example, the study highlighted the general increasing interest towards Flutter; software educators may consider including the SDK in their teaching. Educators may also consider focusing on Flutter-related topics that SO users are struggling with, such as setup and build automation as well as networking. The inclusion of such topics may help students (i.e., future software engineers) understand these topics, what they specifically entail, and some of the potential challenges that might be encountered.

Contributors of Flutter can utilize the results of this study to pinpoint challenges that SO users are facing and seeking help rom other users to resolve, such as setup and build automation. Having such knowledge can be beneficial when contributors aim to eliminate hurdles that contribute to such challenges or provide guidelines to mitigate these challenges. Moreover, the results revealed that SO users are asking about topics that have been extensively explained in Flutter documentation, such as setup and build automation, which was also identified as the most popular and difficult topic. This scenario provides Flutter contributors with opportunities to pinpoint any deficiencies or ambiguities in the current documentation, if any, and improve upon them. They can also leverage SO as a platform to promote the documentation by answering related questions through directing users to the relevant documentation sections.

## 6. Threats to validity

The validity of this study is subject to external, construct, and reliability threats. Each threat and its corresponding mitigation strategies, when applicable, are discussed below.

### 6.1. External validity

The external validity of a study is concerned with determining a domain to which the study's results can be applied (Kitchenham et al., 2015; Runeson et al., 2012). This empirical study and its results are based on Flutter-related questions on SO, with the main goal of the study being gaining a better understanding of Flutter-related questions on the website. Though the study has revealed many findings on Flutter, the generalizability of these findings cannot be claimed, as the study might have failed to include a Flutter-related topic that is discussed on other platforms. Another potential threat to this study's external validity is the identification of Flutter-related questions on SO. Though the applied search method aimed to include all Flutter-related questions on SO through the utilization of tag-based and content-based filterings, there is still a possibility that the search missed a few Flutter-related questions, as users might have utilized other terms to reference Flutter. Hence, the study and its findings are limited to questions that were identified using the described search method.

### 6.2. Construct validity

The construct validity of a study is concerned with the degree to which the utilized scales, metrics, and instruments actually measure what they are supposed to measure (Kitchenham et al., 2015). The first potential threat to this study's construct validity is the measurement of interest towards Flutter in addition to popularity and difficulty of Flutter-related topics, due to the fact that the results were derived solely from heuristics. Though the study only utilized known heuristics that were commonly employed in previous, related studies (Abdellatif et al., 2020; Rosen and Shihab, 2016; Haque et al., 2020), one should only interpret the findings based on the context of the heuristics that were utilized. The second potential threat to this study's construct validity is the usage of statistical measures and tests. The study has only utilized common statistical measures, such as percentages and median, to convey its findings; the Cohen's Kappa coefficient to measure inter-rater agreement; and the Kendall correlation test to examine the relationship between each popularity and difficulty heuristic. These statistical measures and tests are often utilized to measure what we intended to measure and the usages of most of these tests and measures is recommended by the empirical standards for software engineering research (Ralph et al., 2020; Gisev et al., 2013).

### 6.3. Reliability

The reliability of a study is concerned with the reproducibility of the study and its findings. When following the steps of the study, future researchers should reach the same conclusions and findings (Runeson et al., 2012). A potential threat to this study's reliability is the reliance on human judgment in assigning names to the identified Flutter-related topics, which may introduce subjectivity. To mitigate any potential introduced subjectivity, the authors named each identified topic through applying an open card sort method that was also utilized in previous studies (Haque et al., 2020; Fincher and Tenenberg, 2005). We note that any subjectivity resulting from human judgment when naming identified topics could not have been avoided, as the LDA is limited to only aggregating questions into word clusters based on their similarity without providing names and that one needs to manually assign a name to each word cluster (Silva et al., 2021; Rosen and Shihab, 2016).

## 7. Related work

This section includes studies that analyzed SO questions to better understand a software engineering subject. Specifically, the amassed summary detailed ones that leveraged SO to better understand subjects related to mobile development. A few additional studies that utilized SO to understand a subject other than mobile development were also listed below, as they served as the foundation for the design of this study.

Linares-Vásquez et al. (2013) conducted an exploratory analysis of mobile development on SO using LDA. The authors utilized LDA to extract the main discussion topics from more than 400,000 mobile-development-related questions. The study found that the set of identified questions revolve around 14 topics. Though the aforementioned study's focus is mobile development in general and it was published before the introduction of Flutter, around 36% of these topics also appeared in Flutter-related questions, which are: layout, data mapping and conversion, database, media, and unknown errors. The authors further analyzed these questions and topics to understand the answering trends. The study found that the majority of developers focus their content on one mobile platform, such as focusing on answering Android-related questions. The study noted that some mobile-development-related topics tend to have a higher rate of accepted answers, such as data types, compatibility, and layout. The study found that questions related to accessing/browsing web content and graphics editing have lower rates of accepted answers. Our study has also found that layout topic has a relatively high accepted answer rate, in comparison to other Flutter-related topics.

Beyer and Pinzger (2014) manually categorized 450 SO questions that are Android-related. The study found that most of these questions are of the types of “How to” and “What is the problem”. Moreover, the study categorized the set of questions into the following categories, sorted based on decreasing frequency: user interface, core elements, libs/APIs, Android system, input, webkit, media, database, networking, and other. The study has also revealed that the most asked questions are related to user interface and core elements. Though the study solely focused on Android-related questions and was published prior to Flutter release, our study found similar results. Our study has also found that Flutter-related questions include layout, networking, database, and media topics. Moreover, our study has also found that layout is one of the most discussed Flutter-related topics; a layout is a structure that supports the visual components of a user interface (Galitz, 2007).

Building upon the previous two studies, Rosen and Shihab (2016) analyzed over 13,000,000 SO mobile-related posts using LDA to understand what mobile developers ask about. The study found that developers ask about a variety of topics that can be categorized as: application distribution, mobile APIs, mobile tools, user interface development, sensors and context, and data management. The study found that application distribution, mobile tools, and user interface development are the most popular categories. Moreover, the study revealed that mobile APIs, device input, and HTML5/browser are among the most difficult categories. The study also explored whether there are platform specific issues and found that though different mobile platforms have different issues, app distribution; user interface development; and input topics are common among all platforms. Moreover, the study found that Android developers are more concerned about platforms’ extensive and open API and iOS developers are mostly concerned about distributing and deploying applications. The study also investigated the types (i.e., what, how, and why) of questions mobile developers ask and found that mobile developers mostly seek instructions on how to do something, which illustrates a need to include more working examples in current documentations.

Vendome et al. (2019) conducted a pilot study to examine how mobile developers utilize accessibility APIs. The authors analyzed over 13,000 Android applications and found that developers rarely use accessibility APIs and that the majority of analyzed applications lack assistive descriptions. Therefore, the study shifted its focus to understand the perspectives of SO mobile developers regarding mobile applications accessibility. Through a manual analysis of 366 SO discussion threads, the authors created a taxonomy of Android accessibility aspects discussed by developers, identified aspects that developers implemented, and identified aspects that developers are struggling with. The study found that support for visually impaired people is the most discussed aspect and that discussions related to using accessibility APIs for purposes not related to accessibility is the second most discussed aspect.

Multiple other studies have analyzed SO questions using LDA to gain a better understating of a number of software engineering subjects, without a specific focus on mobile development, such as Bajaj et al. (2014), Barua et al. (2014), Yang et al. (2016), Zou et al. (2017), Ahmed and Bagherzadeh (2018), Abdellatif et al. (2020), Haque et al. (2020), Han et al. (2020) and Alfayez et al. (2022). Moreover, Silva et al. (2021) conducted a survey on topic modeling in software engineering in general, without focusing on SO. The survey extensively analyzed the majority of the studies included in this section. Therefore, we recommend the reader to reference the survey to extend their knowledge on topic modeling in software engineering.

## 8. Conclusion

Flutter is a relatively new cross-platform SDK that aims to facilitate mobile application development. Regardless of the young age of Flutter, the SDK has been significantly increasing in popularity. Thus, with an aim to provide a better understanding of the current discussions revolving around Flutter on SO, this study presents an analysis of SO Flutter-related questions.

A total of 176,876 Flutter-related questions were acquired and analyzed. The results revealed that Flutter has been present on SO since Flutter’s early years and that the interest towards the SDK on SO has been continuously increasing, until it dropped in 2023. Moreover, the results imparted that Flutter-related questions primarily revolve around 12 topics, which are as follows: setup and build automation, layout, routing and navigation, networking, animation and canvas, best practices and experiences, database, state management, assets and media, data mapping and conversion, authentication and authorization, and unknown errors, sorted based on decreasing frequency. The results also revealed that the topic of setup and build automation is the most popular topic, based on average number of view and score counts. Moreover, the results indicated that the topic is the most difficult Flutter-related topic, based on rate of questions that lack an accepted answer and median time elapsed to receive an accepted answer.

This study highlights the general interest towards Flutter on SO. The study grants individuals with a broader perspective on Flutter and underscores a few of the challenges that developers face when utilizing the SDK, specifically through the means of SO. Software engineering researchers, practitioners, educators, and Flutter contributors can utilize this study for guidance on where to steer their future efforts with regards to Flutter.

## CRediT authorship contribution statement

**Afit Alanazi:** Conceptualization, Data curation, Methodology, Software, Validation, Writing – original draft. **Reem Alfayez:** Conceptualization, Methodology, Project administration, Resources, Software, Supervision, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

The authors would like to thank Deanship of Scientific Research (DSR) in King Saud Universityfor funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR).

## References

Abdellatif, A., Costa, D., Badran, K., Abdalkareem, R., Shihab, E., 2020. Challenges in chatbot development: A study of stack overflow posts. In: Proceedings of the 17th International Conference on Mining Software Repositories. pp. 174–185.

Abdi, H., 2007. The Kendall rank correlation coefficient. Encycl. Meas. Stat. 2, 508–510.

Ahmed, S., Bagherzadeh, M., 2018. 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. pp. 1–10.

Alfayez, R., Ding, Y., Winn, R., Alfayez, G., 2022. What is discussed about software engineering ethics on stack exchange (q&a) websites? A case study. In: 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications. SERA, IEEE, pp. 106–111.

Bagherzadeh, M., Khatchadourian, R., 2019. Going big: A large-scale study on what big data developers ask. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 432–442.

Bajaj, K., Pattabiraman, K., Mesbah, A., 2014. Mining questions asked by web developers. In: Proceedings of the 11th Working Conference on Mining Software Repositories. pp. 112–121.

Barua, A., Thomas, S.W., Hassan, A.E., 2014. What are developers talking about? an analysis of topics and trends in stack overflow. Empir. Softw. Eng. 19 (3), 619–654.

Beyer, S., Pinzger, M., 2014. A manual categorization of android app development issues on stack overflow. In: 2014 IEEE International Conference on Software Maintenance and Evolution. IEEE, pp. 531–535.

Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (Jan), 993–1022.

Caldiera, V.R.B.-G., Rombach, H.D., 1994. Goal question metric paradigm. Encycl. Softw. Eng. 1 (528–532), 6.

Ceci, L., 2021. Mobile app usage - Statistics & facts. URL https://www.statista.com/topics/1002/mobile-app-usage/#dossierKeyfigures.

Davis, N., 2005. Developing secure software. Softw. Tech News 8 (2), 3–7.

Fincher, S., Tenenberg, J., 2005. Making sense of card sorting data. Expert Syst. 22 (3), 89–93.

Galitz, W.O., 2007. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. John Wiley & Sons.

Gisev, N., Bell, J.S., Chen, T.F., 2013. Interrater agreement and interrater reliability: key concepts, approaches, and applications. Res. Soc. Adm. Pharm. 9 (3), 330–338.

Han, J., Shihab, E., Wan, Z., Deng, S., Xia, X., 2020. What do programmers discuss about deep learning frameworks. Empir. Softw. Eng. 25 (4), 2694–2747.

Haque, M.U., Iwaya, L.H., Babar, M.A., 2020. 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, pp. 1–11.

Hubert, M., Rousseeuw, P., 2010. International Encyclopedia of Statistical Science. Springer-Verlag.

Joorabchi, M.E., Mesbah, A., Kruchten, P., 2013. Real challenges in mobile app development. In: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 15–24.

Kitchenham, B.A., Budgen, D., Brereton, P., 2015. Evidence-Based Software Engineering and Systematic Reviews, vol. 4, CRC Press.

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 159–174.

Linares-Vásquez, M., Dit, B., Poshyvanyk, D., 2013. An exploratory analysis of mobile development issues using stack overflow. In: 2013 10th Working Conference on Mining Software Repositories. MSR, IEEE, pp. 93–96.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B., 2011. Design lessons from the fastest q&a site in the west. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2857–2866.

Medhi, J., 1992. Statistical Methods: An Introductory Text.

Minelli, R., Lanza, M., 2013. Software analytics for mobile applications–insights & lessons learned. In: 2013 17th European Conference on Software Maintenance and Reengineering. IEEE, pp. 144–153.

Miola, A., 2020. Flutter complete reference: Create beautiful, fast and native apps for any device. Independently published.

Payne, R., 2019. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps. A Press.

Puth, M.-T., Neuhäuser, M., Ruxton, G.D., 2015. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits. Anim. Behav. 102, 77–84.

Ralph, P., Ali, N.b., Baltes, S., Bianculli, D., Diaz, J., Dittrich, Y., Ernst, N., Felderer, M., Feldt, R., Filieri, A., et al., 2020. Empirical standards for software engineering research. arXiv preprint arXiv:2010.03525.

Rosen, C., Shihab, E., 2016. What are mobile developers asking about? a large scale study using stack overflow. Empir. Softw. Eng. 21 (3), 1192–1223.

Runeson, P., Host, M., Rainer, A., Regnell, B., 2012. Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons.

Schober, P., Boer, C., Schwarte, L.A., 2018. Correlation coefficients: appropriate use and interpretation. Anesth. Analg. 126 (5), 1763–1768.

Silva, C.C., Galster, M., Gilson, F., 2021. Topic modeling in software engineering research. Empir. Softw. Eng. 26 (6), 1–62.

Ř.ehůřek, R., Sojka, P., 2010. Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. ELRA, Valletta, Malta, pp. 45–50, http://is.muni.cz/publication/884893/en.

Vailshery, L.S., 2022. Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021. URL https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/.

Vendome, C., Solano, D., Liñán, S., Linares-Vásquez, M., 2019. Can everyone use my app? an empirical study on accessibility in android apps. In: 2019 IEEE International Conference on Software Maintenance and Evolution. ICSME, IEEE, pp. 41–52.

Windmill, E., 2020. Flutter in Action. Simon and Schuster.

Yang, X.L., Lo, D., Xia, X., Wan, Z.Y., Sun, J.L., 2016. What security questions do developers ask? a large-scale study of stack overflow posts. J. Comput. Sci. Tech. 31 (5), 910–924.

Zou, J., Xu, L., Yang, M., Zhang, X., Yang, D., 2017. Towards comprehending the non-functional requirements through developers' eyes: An exploration of Stack Overflow using topic analysis. Inf. Softw. Technol. 84, 19–32.