"Now, I am not sure what to do!" Understanding GitHub Action Developer Information Needs

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract. `TO DO` ►*Lo escribimos al final*◄

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

The software industry has widely adopted Continuous Integration and Delivery (CI/CD) practices to automate software engineering tasks [1]. These practices help minimize integration issues, enable frequent integration, automatically deploy changes, and speed up feedback loops for software developers [2]–[4].

Within the GitHub ecosystem, developers can automate software engineering tasks through GitHub Actions (GA), a widely adopted tool for implementing CI/CD [1]. GA provides technical mechanisms to define, execute, and manage workflows using `YAML` files, allowing developers to specify automated tasks triggered by events such as code pushes, and pull requests.

While GA provides mechanisms to automate software engineering tasks, developers still require specific additional information to exploit its capabilities. For instance, a Stack Overflow (SO) user expressed their need for further assistance after attempting various alternatives to download a private repository using Cargo in GA. The user staged: *"Now I am not sure what to do"*[1], reflecting the need for specific information and guidance after trying several approaches to complete the task. This example aligns with previous studies, which show that developers often seek information to understand how to implement specific tasks, indicating a need for additional guidance that is not readily available and hinders their progress [5].

Previous research has extensively explored developer information needs. For instance, Ko *et al.* analyzed software developers' day-to-day information needs by observing seventeen developers at a large software company. They found that developers frequently sought information about artifacts and coworkers, such as awareness about coworkers' activities and understanding the intent behind existing code [6]. Buse and Zimmermann confirmed that information needs can vary widely depending on various factors, highlighting the importance of tailored information tools and the role of software analysts [7]. Other studies are in the context of similar software artifacts such as Infrastructure as Code (IaC). For example, Ouni *et al.* delved into the challenges and benefits of maintaining and evolving IaC scripts [8]. Similarly, Rahman *et al.* identified common questions about Configuration as Code (CaC) tools, such as syntax errors and provisioning instances, underscoring the need for better support and documentation [9]. None of them, however, have been explicitly conducted regarding GA information needs.

We present an empirical study aimed at characterizing the GA developer information needs. Similar to Liu *et al.* [10], we define GA developer information needs as the questions related to GA posed on SO by individuals from diverse backgrounds (e.g., students, professional developers, etc.), collectively referred to as developers. Our study is based on XXX SO posts associated with GA, involving XXX specific sentences that highlight developer information needs. The study focuses on XXX.

`PV` ►*Results: 1 paragraph* ◄
`PV` ►*Beneficiary: 1 paragraph* ◄
`PV` ►*Contributions: 1 paragraph* ◄

II. RESEARCH QUESTIONS

The goal of this study is to characterize the information needs of GA developers. The purpose is to define a taxonomy of the information needs that developers express on the Stack

---

[1]https://stackoverflow.com/questions/68692017/how-to-download-another-private-repository-in-github-actions-with-cargo

Overflow platform. We propose two concrete research questions to guide our study, some of which are inspired by prior empirical research in the Stack Overflow context [11]. With these RQs, we aim to better understand (i) the *current level of interest of CI/CD tools* (RQ$_1$), and (ii) the *types of information needs* of GA developers (RQ$_2$).

The study address the following Research Questions (RQ):

- **RQ1: What is the current level of interest in the topic?** Similar to Haj *et al.* [11] but this time with a focus on GA, this question aims to identify the trend in the number of SO posts related to GA and compare it with the trends of other popular tools in Continuous Integration. By conducting this comparison, we will determine the level of interest in GitHub Actions.
- **RQ2: What types of information needs are presented in SO posts about GitHub Actions?** This research question seeks to characterize the types of information needs of GA developers. Understanding these needs can help in the development of tools to automatically assist with targeted measures, ensuring developers receive timely support and guidance.

III. STUDY DESIGN

In this section, we give an overview of our study design comprising three major phases, namely data selection, data curation, and data analysis. We structure Sections III-A and III-B according to the individual steps of each of the first two phases. For the sake of avoiding redundant descriptions, Section III-C provides a more general introduction into the analysis techniques used in our study, as they are partially reused across our tow RQs. We will describe their applications in terms of the concrete analyses in more detail in Section **??**.

A. Data Collection

B. Data Curation

C. Analysis Techniques

PV ▶*Sandro, acá falta info. Podrías generar una lista de cada paso que realizaste para 1º) recolectar los datos y, 2º) para procesar los datos. Son dos procesos diferentes. Necesitamos describir en detalle el procedimiento que seguimos y con la lista lo yo lo puedo generar. Idem para RQ2*◀

RQ1 steps

1) Define the 6 most popular CI/CD tools. According to the results of a survey conducted by JetBrains TeamCity in 2023, the six most popular CI tools are Jenkins, GitHub Actions, GitLab CI, Azure DevOps, CircleCI, and Travis CI.
2) Select the primary tags related to these tools. We identified the most significant tags related to each of these tools on StackOverflow, which are: 'github-actions', 'jenkins', 'gitlab-ci', 'azure-devops', 'circleci', and 'travis-ci'. Although there are additional derivative tags related to these tools, our analysis focuses on the primary tags to effectively compare the trends in the posts. We extracted the

monthly number of posts tagged with the most popular CI tools from the beginning of 2019 to October 2023
3) Data was queried from Stack Exchange Data Explorer (SEDE). Query link: https://data.stackexchange.com/stackoverflow/query/1802537
4) This dataset includes a total of 60,389 posts, distributed as follows: 9622 with the tag 'github-actions'; 19354 with 'jenkins'; 6581 with 'gitlab-ci'; 22797 with 'azure-devops'; 1095 with 'circleci'; and 940 with 'travis-ci'.
5) Group number of questions by month and plot them.

RQ2 steps

1) Stack Overflow Data Dump was downloaded in January 2023.
2) A database was created and queried using PostgresSQL.
3) Query to create and upload Posts.xml file cand be found in `Taxonomy/data/Queries/PostsDBsetup.sql`
4) 3 queries were made to filter by title, body, and tags.
5) **Tag Filtering:** Posts were filtered using tags associated with GitHub Actions. These tags were identified through a search for "github actions" in the StackOverflow tag search bar, resulting in a list that included 'github-actions', 'building-github-actions', 'github-actions-self-hosted-runners', 'github-actions-runners', 'github-actions-services', 'github-actions-artifacts', 'github-actions-reusable-workflows', 'github-actions-workflows', and 'github-actions-marketplace'.
6) **Title and Body Filtering:** We also scrutinized the post titles or bodies for mentions of GitHub Actions or its variations. The rules that we established for variations of a word or phrase were considering various capitalizations of the first letter of the phrase or word, and hyphenations ('github actions', 'github-actions', 'Github actions', 'Github-actions', 'Github Actions', 'Github-Actions').
7) We used the intersection of these three datasets as our, in total 2903 posts.
8) **Sampling:** Given the nature of our study, we opted for a different approach compared to methodologies used in [10] or [12]. We chose not to select the top-rated posts since many relevant troubleshooting questions usually receive few votes. To ensure a democratic sampling of our data, we decided on a random sample.
   We calculated the necessary sample size n for estimating proportions within a finite population, using the established formula as described by [13]. We selected a confidence level of 95%, corresponding to an error e of 0.05 and a z-value of 1.96. The assumed value of p was set at 0.5. As a result, our data sample has 340 posts.
9) **Manual Filtering:** A manual inspection was applied to ensure that the post title were GA related. Also, the posts with broken links were dismissed.
10) **Coding** Each post has some metadata. For instance, the title and the body. The body comes in HTML. So we used BeautifulSoup to extract the text. Afterward, we replace

code blocks, blockquotes, and links with placeholders: -CODE-, -BLOCK-, and -LINK-.

11) The text from titles and bodies was then segmented into sentences, obtaining 3176 sentences from our data sample. This can be founded here: `Taxonomy/taxonomy.ipynb`

12) **Manual classification** We decided to apply an hybrid card sorting. We took the sentences of 50 posts. We analyzed and classified them together. We got a set 9 groups of questions. The rest of the sentences that were not classified were considered as Non-Relevant.

13) We tried to classify the full sample of questions. However, since not all the questions fit in those categories, we modified them. Finally, we had a taxonomy of the Relevant Information with definitions.

14) Afterwards, we analyzed the categories of RIs and grouped them into Information Needs classes. Definitions were given to these classes too.

1) Taxonomy evaluation

The classification process was carried out by two of the co-authors, with each creation of new categories involving thorough discussion. However, to ensure the reliability of the taxonomy, we decided to evaluate it as suggested in [Ref: TaxonomiesEvaluation]. This evaluation procedure consists of three stages:...

IV. ANALYSIS & RESULTS

A. RQ1: Level of interest

The monthly number of questions posted by developers on StackOverflow is illustrated in Figure 1. We can highlight two distinct periods: between 2019 and early 2023, and from mid-2023 onwards. In the first period, we observed a decline in the number of posts for tools like Jenkins and Azure DevOps, while GitHub Actions, released on November 13th, 2019, surged in relevance, eventually matching or surpassing these tools. This rapid growth in popularity among the developer community corroborates the findings from [14], which identified GitHub Actions as the most used tool in Continuous Integration. Notably, the number of questions for GitHub Actions reached over 350 per month.

In the second period, starting from mid-2023, there was a general decline in the number of questions across all tools, including GitHub Actions, which dropped to about 200 questions per month. This trend is not unique to GitHub Actions; similar pattern were observed for other CI tools. This decline could be attributed to the increasing popularity of large language model-based tools, such as ChatGPT. Therefore, despite the reduction in the number of questions about GitHub Actions, this could be a global effect of new querying tools rather than a decrease in interest in GitHub Actions itself.

B. RQ2: Types of Information Needs

The analysis of StackOverflow posts revealed a taxonomy of developer information needs (DN), which includes categories such as Error Handling (EH), Functionality Implementation (FI), Orientation (OR), GHA Learning (LE), Insufficient Implementation (II), Incompatibility (IN), Migration (MI), and Alternative Solution (AS). Each category is further divided into relevant information (RI) types that developers seek when addressing issues in their code or workflow 2.

Error Handling (EH) emerged as the most frequent category, accounting for 52.65% of the posts I. This indicates that developers often encounter errors in their code and seek solutions to diagnose and resolve these issues. Within this category, "Where does the error occur?" (EH1) and "What is the erroneous behavior?" (EH5) were the most common RI types, highlighting the need for specific information about the nature and location of errors I.

Functionality Implementation (FI) was the second most common category, with 44.71% of posts (see Table I). This suggests that many developers are focused on creating new features or enhancing existing ones using GitHub Actions. The primary RI type within this category was "Implementation goal" (FI1), reflecting the developers' need to understand the specific features or goals of their implementation III.

Orientation (OR) accounted for 38.82% of posts, showing that developers frequently seek guidance on how to proceed with their projects. Common RI types in this category included "Is it possible to do this?" (OR1) and "Search for explanations" (OR2), indicating that developers often need advice on feasibility and troubleshooting.

GHA Learning (LE), with 35.88% of posts, highlights the importance of acquiring knowledge about GitHub Actions. Developers commonly asked about learning specific functions (LE1) and searching for documentation (LE2), underscoring the need for educational resources and clear documentation.

Insufficient Implementation (II) and Incompatibility (IN) were also significant, with 25.29% and 14.12% of posts, respectively I. These categories reflect situations where developers' implementations do not meet their requirements or where code that works locally fails on GitHub Actions. The main RI types here were "Not working as expected" (II1) and "Work locally, fail in GHA" (IN1) I.

Migration (MI) and Alternative Solution (AS) were less frequent but still relevant, with 3.53% and 2.35% of posts. These categories indicate that developers sometimes need to transition their CI/CD code to GitHub Actions or explore different methods to solve their problems.

The distribution of the number of sentences and posts for each RI category, sorted by DN classification, shows that certain information types are more frequently sought after than others. For example, RI types like EH1, EH5, and FI1 had a high number of sentences and posts, emphasizing the critical areas where developers need support III.

Additionally, the analysis revealed that posts could be associated with multiple DN categories. The majority of posts (152) were linked to two DN categories, while some posts were linked to three, four, or five categories, indicating the complex and multifaceted nature of developers' information needs IV.
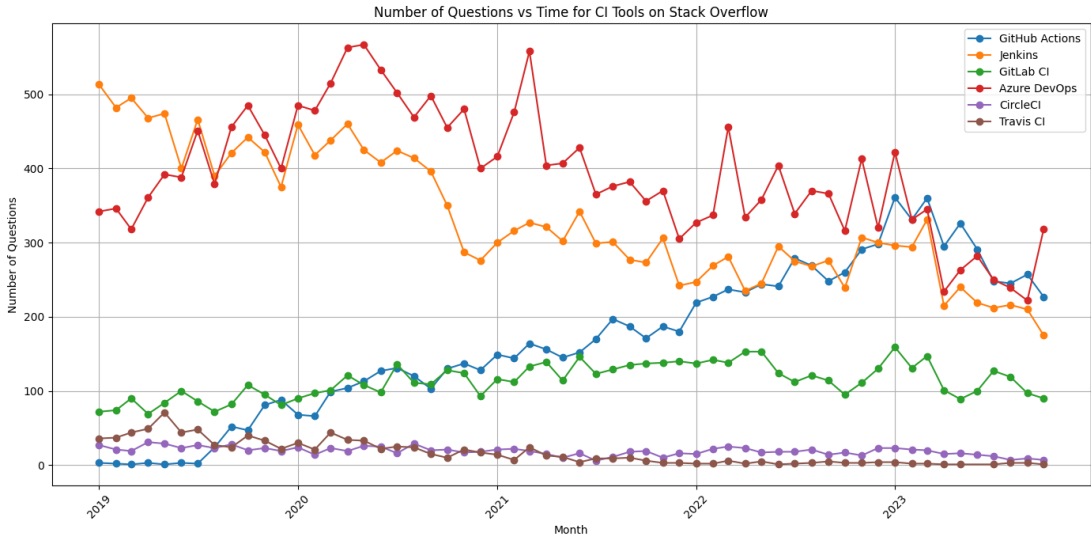
Fig. 1: Monthly number of questions with 'github-actions', 'jenkins', 'gitlab-ci', 'azure-devops', 'circleci', and 'travis-ci' tags correspondly in Stack Overflow. **PV** ▸*probando tamaño*◂

Overall, the results provide valuable insights into the common issues and information needs of developers using GitHub Actions. This taxonomy can guide the development of targeted documentation, tutorials, and support resources to better assist developers in their workflows. The findings also highlight the importance of community-validated solutions, as evidenced by the high number of accepted answers in categories like Error Handling (EH) and Functionality Implementation (FI) 5.

V. ANALYSIS AND RESULTS

In this section, we analyze the developer needs (DN) and relevant information (RI) for GitHub Actions based on the classified sentences from StackOverflow posts. The taxonomy of DNs and RIs, along with their definitions, are presented in Figure reffig , and the detailed results are provided in the subsequent tables and figures.

A. Taxonomy of Developer Needs and Relevant Information

Figure reffig shows the taxonomy of DNs and RIs used to classify the sentences. The DNs are categorized into eight main classes: Error Handling (EH), Functionality Implementation (FI), Orientation (OR), GHA Learning (LE), Insufficient Implementation (II), Incompatibility (IN), Migration (MI), and Alternative Solution (AS). Each DN class has associated RIs that further specify the type of information developers seek.

B. Developer Needs Distribution

Table reftable provides a detailed taxonomy of DNs with the number of posts and their percentages. The most prevalent DN is Error Handling (EH) with 179 posts (52.65%), followed by Functionality Implementation (FI) with 152 posts (44.71%) and Orientation (OR) with 132 posts (38.82%). Migration (MI) and Alternative Solution (AS) are the least common needs, representing only 3.53% and 2.35% of the posts, respectively.

C. Relevant Information Categories

The classification of sentences into RI categories is detailed in Table reftable . This table lists the RI categories, their definitions, and examples. The most frequently occurring RI is FI1 (Implementation goal), with 194 sentences. The least common RIs include OR6 (What action do you recommend?) with 2 sentences and LE2 (Documentation search) with 6 sentences.

D. Distribution of Sentences and Posts

Table reftable shows the number of sentences and posts associated with each RI. It is evident that EH1 (Where does the error occur?) and EH5 (What is the erroneous behavior?) are among the most common RIs within the Error Handling category, highlighting the importance of understanding the specific error and its manifestations in the code. The GHA Learning category, particularly LE1 (Learning specific functions), also has a significant number of sentences, indicating a strong need for learning and understanding specific GitHub Actions functionalities.

E. Non-Relevant Sentences

Out of the total 3176 sentences analyzed, 2176 sentences did not fall into any of the RI categories, as shown in Table reftable . This indicates that a substantial portion of the posts may contain general information or context that does not directly address specific information needs.

F. Co-occurrence of Developer Needs

The co-occurrence of DNs in posts is visualized in Figure reffig . This heatmap illustrates the frequency with which different DNs appear together in the same posts. It is notable that Error Handling (EH) often co-occurs with Functionality Implementation (FI) and Orientation (OR), suggesting that

developers frequently encounter errors while implementing new features or seeking guidance on best practices.

G. Accepted vs. Non-Accepted Posts

Figure reffig compares the number of accepted and non-accepted posts for each DN. Error Handling (EH) has the highest number of accepted answers, indicating that posts related to diagnosing and resolving errors are more likely to receive accepted solutions. Conversely, Migration (MI) and Alternative Solution (AS) have the lowest acceptance rates, possibly due to the complexity and variability of these tasks.

H. Number of DN Categories per Post

Table reftable presents the number of posts that belong to zero or more DN categories. Most posts fall into one or two DN categories, with only a few posts encompassing multiple DNs. This distribution suggests that while some developer needs are interconnected, many posts focus on addressing a single, specific need.

VI. Conclusion

The analysis of developer needs and relevant information in GitHub Actions reveals that Error Handling, Functionality Implementation, and Orientation are the most common needs among developers. The significant number of non-relevant sentences indicates the importance of filtering and focusing on specific information needs to better support developers. The co-occurrence of DNs and the distribution of accepted answers provide further insights into the challenges developers face and the types of information that are most useful in resolving their issues.

VII. Discussion

VIII. Related Work

IX. Threats to Validity & Limitations

X. Conclusion

Acknowledgment

References

[1] M. Golzadeh, A. Decan, and T. Mens, "On the rise and fall of CI services in GitHub," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022, pp. 662–672.

[2] M. Fowler and M. Foemmel, "Continuous integration," 2006.

[3] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.

[4] B. Fitzgerald and K.-J. Stol, "Continuous software engineering: A roadmap and agenda," *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017.

[5] Y. Zhang, Y. Wu, T. Chen, T. Wang, H. Liu, and H. Wang, "How do developers talk about GitHub actions? evidence from online software development community," in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.

[6] A. J. Ko, R. DeLine, and G. Venolia, "Information needs in collocated software development teams," in *29th International Conference on Software Engineering (ICSE'07)*. IEEE, 2007, pp. 344–353.

[7] R. P. Buse and T. Zimmermann, "Information needs for software development analytics," in *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 2012, pp. 987–996.

[8] A. Ouni, I. Saidani, E. Alomar, and M. W. Mkaouer, "An empirical study on continuous integration trends, topics and challenges in stack overflow," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, 2023, pp. 141–151.

[9] A. Rahman, A. Partho, P. Morrison, and L. Williams, "What questions do programmers ask about configuration as code?" in *Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering*, 2018, pp. 16–22.

[10] M. Liu, X. Peng, A. Marcus, S. Xing, C. Treude, and C. Zhao, "Api-related developer information needs in stack overflow," *IEEE Transactions on Software Engineering*, vol. 48, no. 11, pp. 4485–4500, 2022.

[11] A. H. Yahmed, A. A. Abbassi, A. Nikanjam, H. Li, and F. Khomh, "Deploying deep reinforcement learning systems: a taxonomy of challenges," in *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2023, pp. 26–38.

[12] S. Beyer, C. Macho, M. Di Penta, and M. Pinzger, "Automatically classifying posts into question categories on stack overflow," in *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, 2018, pp. 211–21 110.

[13] M. Triola, W. Goodman, G. LaBute, R. Law, and L. MacKay, *Elementary Statistics*. Pearson Education Canada, 2009. [Online]. Available: https://books.google.ch/books?id=qZIhPwAACAAJ

[14] T. Blog. (2023) Best continuous integration tools for 2023 – survey results. [Online]. Available: https://blog.jetbrains.com/teamcity/2023/07/best-ci-tools/

DN_id	Developer Need	Definition	N° of Posts	Percentage
EH	Error Handling	The developer identifies an issue within their code that causes unexpected behavior or failures and seeks solutions to diagnose and resolve the error.	179	52.65%
FI	Functionality Implementation	The developer aims to design and implement new features or enhancements within their project using GitHub Actions to automate workflows and processes.	152	44.71%
OR	Orientation	The developer looks for advice, best practices, or recommendations on how to proceed with a particular task or decision within their project, using GitHub Actions.	132	38.82%
LE	GHA Learning	The developer is looking to acquire knowledge and understanding of GitHub Actions, requiring documentation, tutorials, or examples to learn how to effectively use its features and capabilities.	122	35.88%
II	Insufficient Implementation	The developer finds that their current implementation falls short of the desired functionality or specifications, necessitating further enhancements or modifications.	86	25.29%
IN	Incompatibility	The developer's code functions correctly in their local environment but encounters issues or fails to execute as expected when run within GitHub Actions.	48	14.12%
MI	Migration	The developer seeks to transition their continuous integration and continuous deployment (CI/CD) processes from another platform to GitHub Actions, ensuring compatibility and functionality during the migration.	12	3.53%
AS	Alternative Solution	The developer has an existing solution in place but is interested in exploring different methods or tools that might offer better performance, efficiency, or simplicity.	8	2.35%

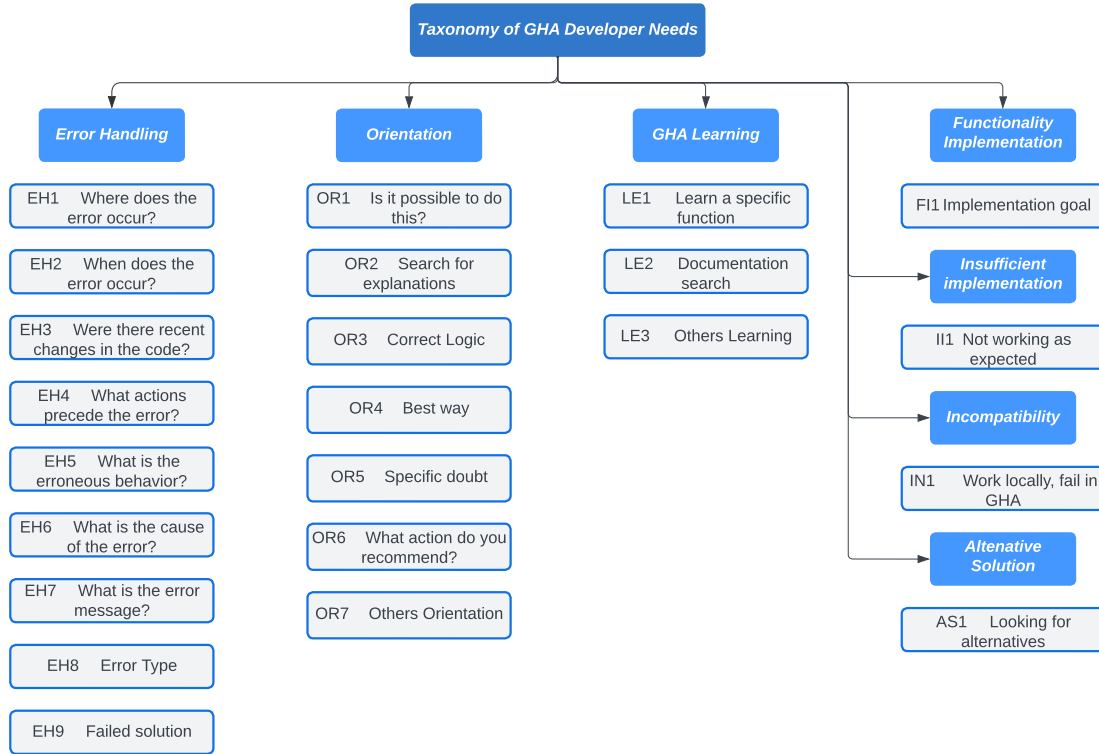TABLE I: Detailed Taxonomy of Developer Needs with Post Counts and Percentages

Fig. 2: Information Needs (IN), their Definitions, and Associated Relevant Information (RI)

RI_id	Relevant Information	Definition	Example
EH1	Where does the error occur?	This includes specific locations in the code such as functions, steps, jobs, stages, or modules where the error manifests.	The line that fails is: -CODE-.
EH2	When does the error occur?	This details the timing of the error, whether it happens at execution, after a certain period, or at the end. It also includes whether the error is constant or intermittent.	The 'strange' thing is that sometimes the test passes and sometimes it doesn't.
EH3	Were there recent changes in the code?	This involves describing any recent changes made by the developer to the code before the error appeared. These changes could be relevant to understanding the cause of the error.	I'm switching a Python project over to poetry for dependency and packaging management, and am running into issues getting my GitHub Actions unit tests working.
EH4	What actions precede the error?	This includes the specific actions or parts of the code that are executed just before the error occurs.	The build job passes but the deploy one fails.
EH5	What is the erroneous behavior?	This describes the incorrect behavior exhibited by the code that indicates an error.	The problem is when I try to use cache I see that the version in -CODE- and -CODE- are always changed so I can't use real cache here.
EH6	What is the cause of the error?	This provides a description of the possible reason or cause behind the error.	Running tests on GitHub Actions with FastAPI fails due to it trying to connect to hosted DB first.
EH7	What is the error message?	This includes the exact error message received, which can help in diagnosing the problem.	This is the error I am getting: -CODE-.
EH8	Error Type	Specifies the type of error encountered (e.g., syntax error, runtime error, etc.).	Running Angular e2e tests using GitHub Actions throws 'DevToolsActivePort file doesn't exist' error.
EH9	Failed solution	This includes explicit or implicit information about any attempted solutions that failed to resolve the issue.	I tried adding -CODE- in the Workflow YAML file, but it did not get reflected in how the container was created and the command still failed.
FI1	Implementation goal	This describes the specific features or goals of the implementation.	I'm trying to set up codecov monitoring for a public R package, where GitHub Actions will run -CODE-.
OR1	Is it possible to do this?	This includes queries about the feasibility of performing a specific action.	Is there a way to install ODBC drivers on github actions?
OR2	Search for explanations	This involves looking for explanations or reasons why something is not happening as expected.	Why does GitHub actions rest API download artifacts by creating a temporary URL?
OR3	Correct logic	Asking if the logic followed is correct. Questioning whether the approach or assumptions and actions are appropriate.	Should I create another entry in my matrix that only relies on the second branch?
OR4	Best way	Asking what the best or correct path is to take to accomplish something.	What's the best way to test my app using GitHub actions?
OR5	Specific doubt	Asking about a specific question related to GitHub Actions functions, policies, or behaviors.	Do I maybe need to add an authToken or something else?
OR6	What action do you recommend?	Asking which GitHub Actions is recommended for their specific need.	If I wanted to run an arbitrary command and make a PR to the repository, which GitHub Actions should I be looking at instead of reinventing my own Actions?
OR7	Others orientation	Seeking advice or clarification on how to resolve a specific issue or how to use a particular tool/feature in GitHub Actions.	Do I need to change to make sure that my playbook runs in GitHub actions?
LE1	Learning specific functions	Asking how to perform specific actions using GitHub Actions.	How can I reference multi-line secrets in GitHub Actions?
LE2	Documentation search	Searching for documentation or examples related to specific GitHub Actions features or commands.	Could you please provide an example for me to refer?
LE3	Others learning	Seeking to understand underlying concepts, mechanics, or broader aspects of GitHub Actions.	Are GitHub Action minutes deducted from the quota of the repo owner or the user who pushes the commit to github.com?
II1	Not working as expected	This describes situations where the developer's implementation does not meet their requirements or functions as expected.	I've found a starter workflow that runs the Gradle build on a commit but I haven't found a way to report the Checkstyle errors as pull-request annotations.
IN1	Work locally, fail in GHA	This indicates that the code works correctly in the local environment but fails when executed in GitHub Actions.	Running it locally works fine but once I push it to our repository in our company domain, I get error due to some extra headers missing.
MI1	Change CI/CD platform	This involves migrating the CI/CD code to GitHub Actions.	I am transitioning my CI/CD over to GitHub Actions and noticed that my prior scripts for deploying Firebase do not work.
AS1	Looking for alternatives	Seeking for alternative approaches to achieve a goal or solve a problem without doing what is already known.	I know that I can split them up into separate repositories and solve it that way, but I am looking for a solution where I don't have to do that.

TABLE II: Detailed Taxonomy of Relevant Information for GitHub Actions

RI_id	Relevant Information	N° of Sentences	N° of Posts
EH1	Where does the error occur?	94	75
EH2	When does the error occur?	8	6
EH3	Were there recent changes in the code?	4	4
EH4	What actions preceded the error?	14	14
EH5	What is the erroneous behavior?	80	70
EH6	What is the cause of the error?	11	10
EH7	What is the error message?	67	61
EH8	Error Type	63	50
EH9	Failed solution	61	50
FI1	Implementation goal	194	152
OR1	Is it possible to do this?	68	59
OR2	Search for explanations	39	32
OR3	Correct Logic	18	17
OR4	Best way	9	9
OR5	Specific doubt	25	21
OR6	What action do you recommend?	2	2
OR7	Others Orientation	16	16
LE1	Learning Specific Functions	141	111
LE2	Documentation Search	6	6
LE3	Others Learning	23	14
II1	Not working as expected	102	86
IN1	Work locally, fail in GHA	63	48
MI1	Change CI/CD platform	14	12
AS1	Looking for alternatives	8	8
NR	**Non-Relevant sentences**	**2176**	—
Total	**Total number of sentences**	**3176**	—

TABLE III: Number of sentences and posts associated to a Relevant Information.

N° of DN Classes	Post Count
0	6
1	71
2	152
3	84
4	23
5	4

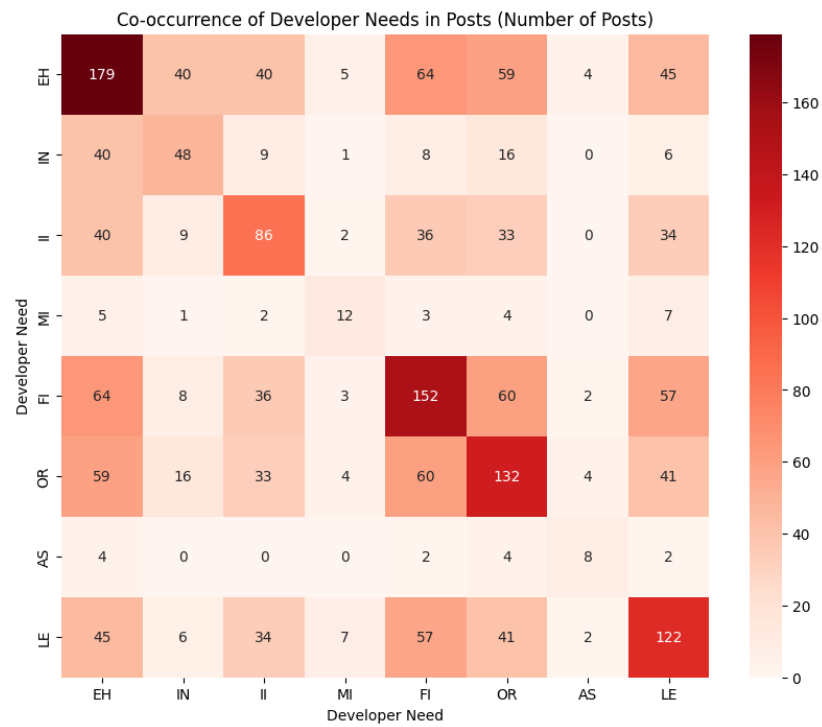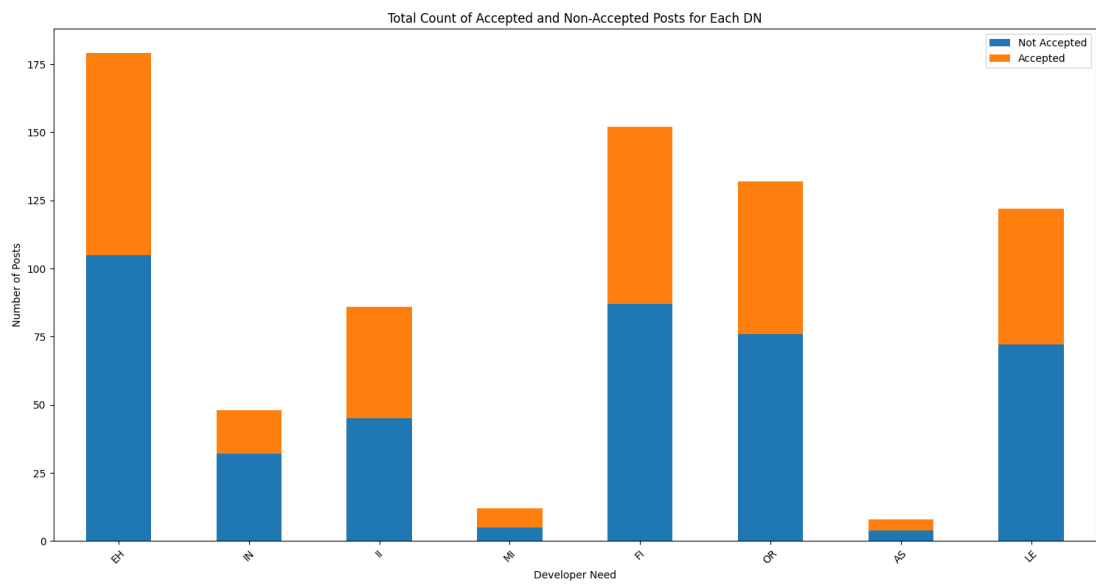TABLE IV: Number of DN Categories and Corresponding Post Counts
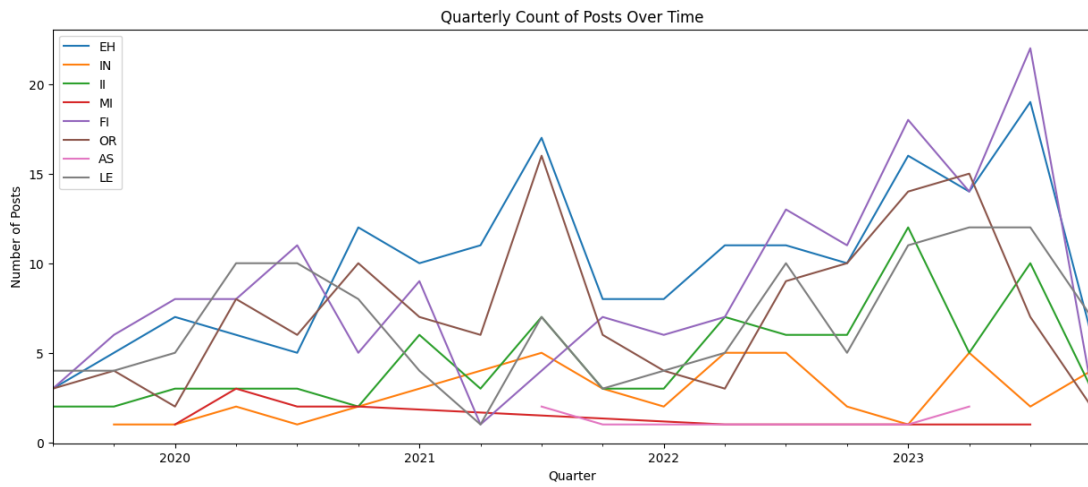
Fig. 3

Fig. 4

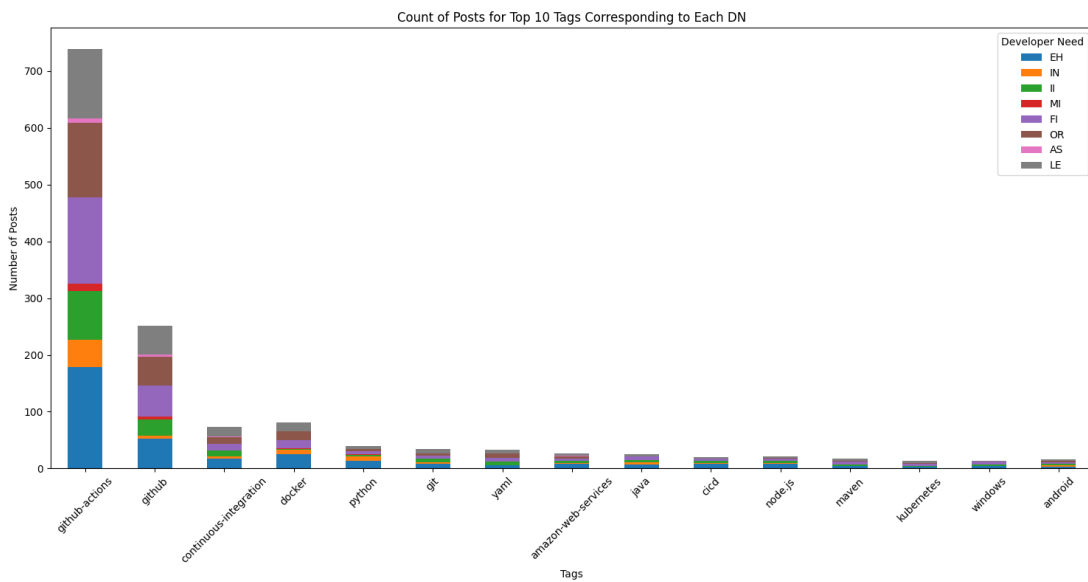Fig. 5: Line plot showing the quarterly count of posts for each Developer Need (DN) category over time.

Fig. 6: Count of Posts for Top 10 Tags Corresponding to Each Developer Need (DN). The stacked bar plot shows the distribution of posts associated with each Developer Need (EH, IN, II, MI, FI, OR, AS, LE) across the top 10 most common tags.
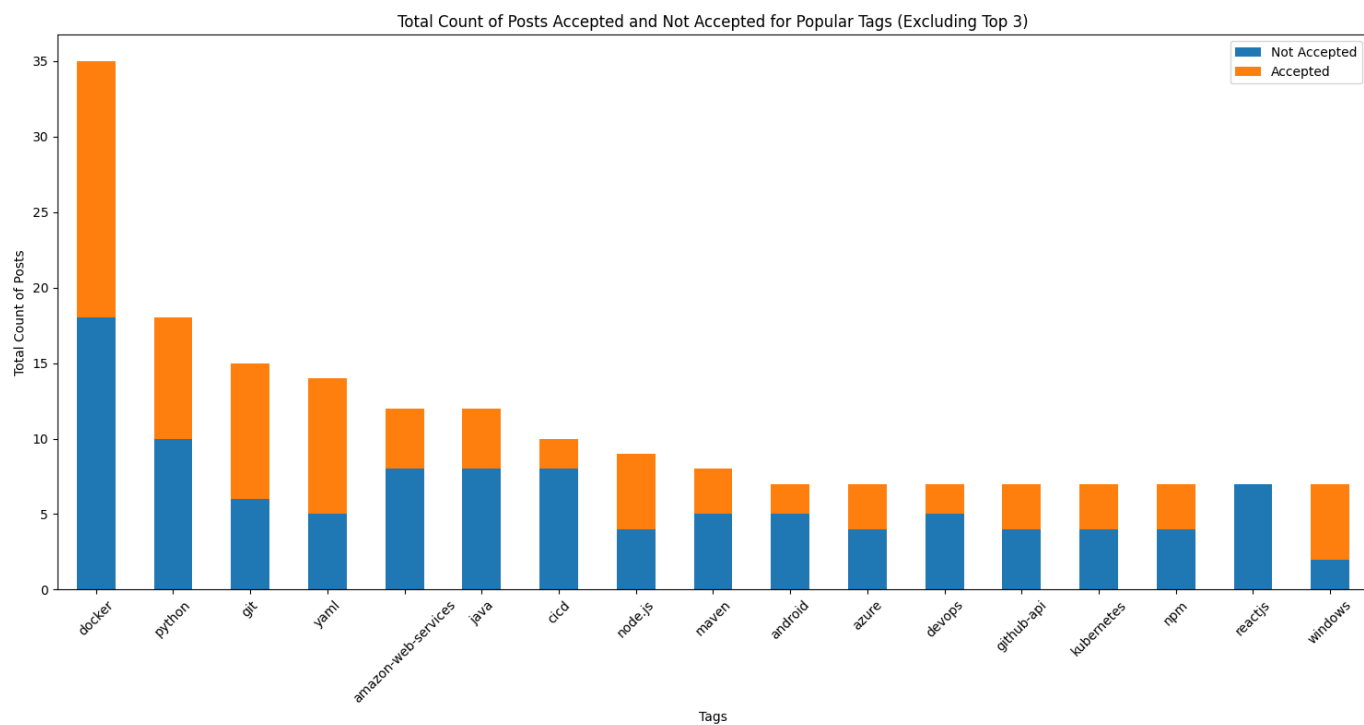
Fig. 7: Total Count of Posts Accepted and Not Accepted for Popular Tags (Excluding Top 3). This stacked bar plot shows the distribution of accepted and not accepted posts for the top popular tags, excluding the three most popular tags.

TABLE V: Most Common Tags for Each Developer Need (DN)

DN	Tag	Count
EH	github-actions	179.0
	github	52.0
	docker	25.0
	continuous-integration	17.0
	python	13.0
IN	github-actions	48.0
	python	9.0
	docker	8.0
	github	6.0
	continuous-integration	4.0
II	github-actions	86.0
	github	29.0
	continuous-integration	11.0
	git	6.0
	yaml	6.0
MI	github-actions	12.0
	github	4.0
	docker	1.0
	python	1.0
	git	1.0
FI	github-actions	152.0
	github	55.0
	docker	14.0
	continuous-integration	12.0
	yaml	7.0
OR	github-actions	132.0
	github	51.0
	docker	15.0
	continuous-integration	11.0
	yaml	7.0
AS	github-actions	8.0
	github	4.0
	continuous-integration	1.0
	docker	1.0
	amazon-web-services	1.0
LE	github-actions	122.0
	github	51.0
	continuous-integration	17.0
	docker	15.0
	git	8.0