“Understanding GitHub Action Developer Information Needs - fix the title

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 leverage GitHub Actions (GA) to specify CI/CD. GA enable the automation of actions based on specific triggers, such as commits, pull requests, issues, comments, and more. This automation simplifies the process of building, testing, and deploying software projects, as developers can share actions across multiple repositories within the GitHub ecosystem.

Despite GA’s robust automation capabilities, developers frequently need supplementary information to explode its potential. These information needs are often articulated at technical forums such as Stack Overflow (SO) at sentence level [5]. For example, the post titled “How to combine Maven Build and Docker in GitHub Actions?”[1], illustrates the need for GA developer information need. The post’s author expressed multiple needs, detailing the specific issues he encountered, the steps he had already taken, and the goals he hoped to achieve with his build and deployment process. 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 [6]. PV ►*quizás esto puede ser la página con referencias de texto, en cualquier caso debe mejorar o eliminarse. Agregar relevancia de las IF a nivel de oración*◄

> **Stack Overflow post: Stack Overflow post:** *“How to combine Maven Build and Docker in GitHub Actions?” ... “I am attempting the following process in GitHub Actions: Build a .war file using maven (successful).” ... “This fails, with Docker unable to find the uploaded artifact.” ... “This fails, with Docker unable to find the .war file.” ... “This fails, with Docker unable to find the .war file.” ... ”This fails due to a permissions error.” ... “This fails, with Docker unable to find the .war file.”*

Previous research has extensively explored developer information needs. Ko *et al.* found that developers often sought information about artifacts and coworkers [7], while Buse and Zimmermann highlighted the variability of these needs, emphasizing tailored tools [8]. Studies on similar software artifacts, such as IaC and CaC, have also been conducted, highlighting the need for better support and documentation [9], [10]. Liu *et al.* stressed the importance of sentence-level analysis to capture developers’ specific needs accurately, revealing multiple and complex needs within single posts [5]. Zhang *et al.* investigated GitHub Actions, presenting a comprehensive taxonomy of GA problems and solutions, but did not focus on sentence-level analysis or developers’ information needs [6]. None of them, however, have been explicitly conducted regarding GA developers’ information needs, nor at the granular level of sentences.

We present an empirical study that characterizes GA developer information needs. Leveraging Liu *et al.*’s definition [11], we specify GA developer information needs as the questions related to GA posed on SO by individuals from diverse

[1] https://stackoverflow.com/q/73128465

backgrounds (e.g., students, professional developers), collectively referred to as developers. We identify and categorize these information needs at the sentence level to provide a detailed understanding of developers' queries. We conjecture that such a fine-grained approach can help better capture essential features of GA developer needs, especially when multiple information needs are involved in a single SO post.

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 PV ►*esta oracion puede mejorar*◄of the information needs that developers express on the Stack 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 [12]. 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 CI/CD tools within the GitHub ecosystem, and how is this interest reflected in repository trends?** Similar to Haj *et al.* [12] 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 specific information needs related to GitHub Actions are frequently expressed by users in Stack Overflow posts?** 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

PV ►*I am working on this section*◄ In this section, we give an overview of our study design comprising three major phases, namely data collection, data curation, and data analysis. We structure Sections III-A and **??** according to these individual phases for each RQ.

A. RQ1 - Level of interest on CI/CD Tools

PV ►*agregar intro breve*◄

Data collection: To answer RQ1, we collected a set of Stack Overflow posts on CI/CD tools. PV ►*crear archivos CSV con la busqueda, luego los post se agruparon por mes, luego plotear el numero de post por mes, para analizar ver la descripcion del resultado: analisis de tendencias por periodo*◄ We based our tool selection criteria on the 2023 JetBrains TeamCity survey, which identified Jenkins, GitHub Actions, GitLab CI, Azure DevOps, CircleCI, and Travis CI as the six most popular CI/CD tools[2]. To identify the associated posts, we used the tags linked to each of these tools on Stack Overflow, namely "github-actions," "gitlab-ci," "jenkins," "azure-devops," "circleci," and "travis-ci." Although there are additional derived tags related to these tools, our analysis focused on the parent tags PV ►*buscar referencia*◄ of each tool to compare trends across posts. We extracted the selected posts by querying the Stack Exchange Data Explorer[3] (SEDE) for the period from 2019 to October 2023.

As a result of this process, we extracted a total of 60,389 posts, distributed as follows: 9,622 with the tag ”github-actions”; 19,354 with ”jenkins”; 6,581 with ”gitlab-ci”; 22,797 with ”azure-devops”; 1,095 with ”circleci”; and 940 with ”travis-ci.” PV ►*alinear descripción herramientas-tags, corregir comillas a formato latex*◄

Data curation: PV ►*, qué procesamiento hubo acá (que preparación de datos tuviste que hacer para poder generar el gráfico)?,podrías listar los pasos si es que la hubo?. - IDEM para RQ2*◄

I

B. RQ2 - Information Needs

PV ►*agregar breve intro*◄

Data collection: To address RQ2, we gathered SO posts related to GA. We began by downloading the Stack Overflow data dump from the official Stack Exchange repository in January 2024 [**?**]. Following the guidelines provided by XXX, we extracted posts by querying tags, titles, and body content PV ►*argumentar mejor*◄. PV ►*creaste BD, y luego hicistes queries sobre la base de datos local*◄

We identified relevant tags using Stack Overflow's search interface[4]. PV ►*cómo definimos los tags?*◄Specifically, we found nine tags associated with the term "github actions," "including: 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." PV ►*en la barra de busqueda de tags, usamos el tag 'github actions'*◄ PV ►*query a la tabla de tags*◄

We also queried the post titles and bodies independently for mentions of GitHub Actions or its variations. We applied text normalization rules to account for different capitalizations and hyphenations (e.g., "github actions," "github-actions," "Github actions," "Github-actions," "Github Actions," "Github-Actions"). PV ►*query a las respectivas tablas y generaste archivos CSV* ◄

As a result, we created three datasets based on the aforementioned queries (tags, post titles, and post bodies) PV ►*dar mas detalles de esto*◄. We formed our final dataset by intersecting these three datasets, which included a total of 2,903 posts.

[2] linktothewebsite

[3] linktothewebsite

[4] linktothewebsite

The dataset encompasses posts starting from XXXX, with each entry providing details such as the type of post (e.g., question, answer, or wiki), creation date, tags, title, and body. Each question includes between one and five tags related to its topics and may have an accepted answer, indicating that the question's owner has found a satisfactory response among the answers provided. PV ►*añadir toda la estadística asociada a las oraciones*◄ PV ►*evaluar agregar una tabla que resuma todo esto*◄ PV ►*la interseccion fue por ID, merge por ID, descartando repetidos*◄

Data curation: PV ►*muestra aleatoria, 400 y testo manual, y eliminamos lo que tenian link malo, y terminamos con 340, luego parseamos el html a texto, y luego dividimos en oraciones, el ttiilo no lo preocesamos , el parseo incluyó codigo, block (enumeracion de cosas , bloques de texto enumerados), links y figuras, y luego dividir,* ◄

Analysis techniques: SH ►

1) *We reviewed the literature and previous taxonomies of Developer Needs (DN).*
2) *We opted to use a hybrid card sorting method. This method involves selecting a sample of cards to generate a classification and then proceeding with the rest. For our sample, we selected 463 sentences corresponding to 50 posts.*
3) *Two classifiers jointly classified the selected sentences. Initially, sentences deemed irrelevant in terms of expressing a DN or those too implicit were excluded. Implicit sentences required significant context from previous sentences to make sense.*
4) *The 146 sentences considered relevant were grouped into 10 categories after two runs.*
5) *We reviewed the remaining sentences from the other posts. During this review, it became evident that some sentences needed re-grouping. We recursively refined the classification as sentences that didn't fit the existing categories emerged until all sentences were classified.*
6) *Sentences were grouped into certain Relevant Information (RI) categories. The definitions of these groups were refined.*
7) *Finally, the Relevant Information categories were grouped into corresponding Developer Needs and assigned definitions.*

◄

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 [13], 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 - Information Needs

In our study, we analyzed a total of 340 posts, which comprised 3,176 sentences. This sample size provides us with a 95% confidence level and a 5% margin of error for our conclusions. Out of these sentences, we identified 1,000 that corresponded to one or more types of Relevant Information (RI). As illustrated in Table IV, 6 posts did not contain any sentences that could be classified into an RI category. In contrast, 4 posts contained sentences with up to 5 different types of RIs, while the majority of posts (152) contained sentences with 2 types of RIs.

For instance, post 68346302 did not have any identifiable RI due to the loss of information in the form of images, snipets of code, or links during the coding process. Similarly, post XXX1 lacked clarity regarding the questioner's intentions, making it difficult to categorize. Utilizing these posts, we examined the occurrence and frequency of various RI categories. The findings, presented in Table III, show the number of sentences and posts associated with each RI category. It is noteworthy that some posts contained multiple sentences with the same RI type. For additional context, Table II provides definitions and examples of these categories.

Our analysis revealed that the most common RI category is *Implementation Goal* (FI1), present in over 152 posts. In this category, users describe their objectives or directly inquire about how to implement specific functionality. The second most common category is *Learning Specific Functions* (LE1), identified in 111 posts, where users seek to learn how to perform specific tasks using GitHub Actions.

The relationship between RIs and Developer Needs (DNs) can be observed in Figure 2, while Table I provides detailed definitions, the number of posts, and the corresponding percentages for each category. *Error Handling* (EH) emerged as the most frequently occurring DN, found in 179 out of 340 posts (52.65%).

Following EH, the next most common DNs were *Functionality Implementation* (FI), *Orientation* (OR), and *GHA Learning* (LE), with percentages of 44.71%, 38.82%, and 35.88% respectively. These results suggest that developers often articulate their implementation goals or seek to learn how to use the tool in their posts. However, considering the

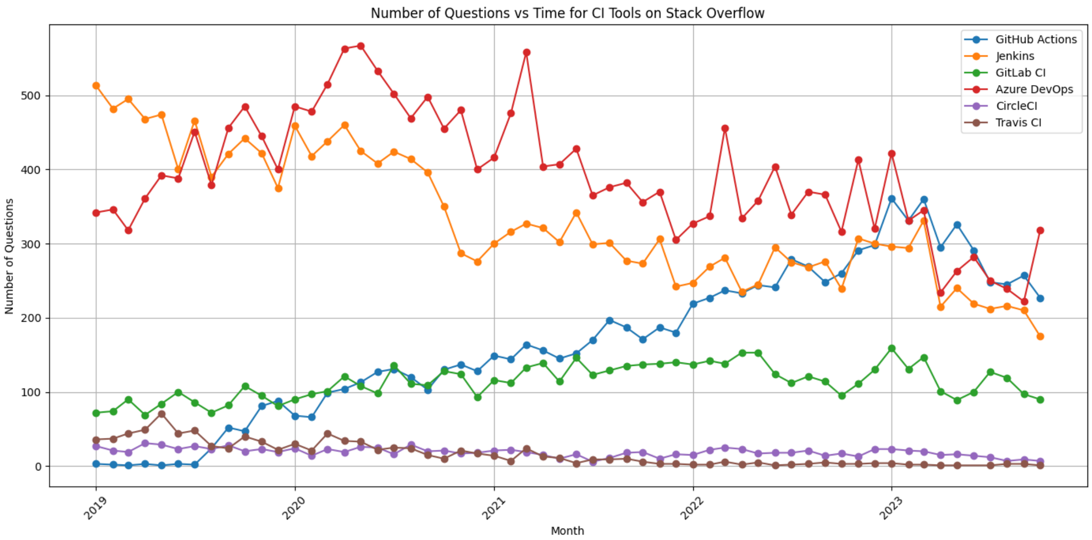

Fig. 1: Monthly number of questions with ‘github-actions’, ‘jenkins’, ‘gitlab-ci’, ‘azure-devops’, ‘circleci’, and ‘travis-ci’ tags correspondly in Stack Overflow. PV ►*probando tamaño*◄

overall frequency of DNs, FI and LE, although commonly mentioned, are not the most prevalent; other needs such as EH and OR are more dominant.

EH typically involves the developer’s need to address errors or failures within their implementation. This is likely because EH is a cross-cutting DN that frequently appears alongside other DNs, as evident from the co-occurrence measurements. In Figure 4, we calculated the overlap coefficient, also known as the Szymkiewicz-Simpson coefficient, to measure the similarity between two sets. This coefficient is defined as the size of the intersection divided by the size of the smaller set:

$$O(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

where $|A \cap B|$ is the number of posts labeled with both DNs A and B, and $\min(|A|, |B|)$ is the number of posts in the smaller set. The coefficient ranges from 0 to 1, where 1 indicates complete overlap and 0 indicates no overlap.

We observed that EH frequently co-occurs with other DNs, with the highest overlap percentage being with *Incompatibility* (IN) at 83%. This might be because Incompatibility can be considered a type of EH, but we chose to separate it as a special case of error handling due to its specific relevance to GitHub Actions, where issues arise locally but not in the tool.

For instance, in post 7281996, sentences like ”It says that it can’t find the -CODE- package, but the package is there.” are categorized as EH5. Similarly, ”I’m setting up the CI/CD process for a NextJS application and everything works fine locally, but when I run the workflow in GitHub Actions, the command -CODE- fails.” represents an implementation incompatibility between local and GA execution. To maintain a comprehensive taxonomy, we decided to keep EH and IN separate.

Additionally, there is a significant overlap between LE and MI (58%). *Migration* (MI) involves the developer’s need to transition from one CI/CD platform to GA, which can coincide with LE or OR. The overlap with OR is also relatively high (33%). Nonetheless, we deemed it important to identify MI separately as a special case relevant to GA.

V. Discussion

VI. Related Work

VII. Threats to Validity & Limitations

VIII. 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] 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, 2021.

[6] 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.

[7] 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.

[8] 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.

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

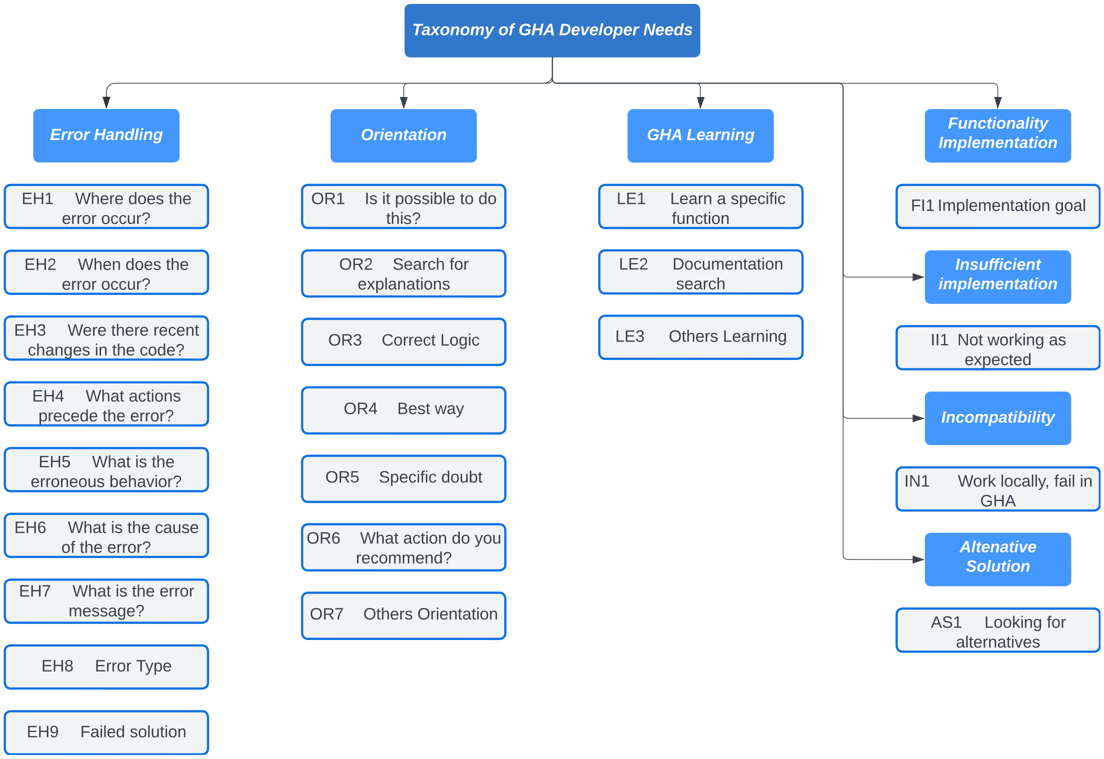
Taxonomy of GHA Developer Needs
Error Handling
EH1 Where does the error occur?
EH2 When does the error occur?
EH3 Were there recent changes in the code?
EH4 What actions precede the error?
EH5 What is the erroneous behavior?
EH6 What is the cause of the error?
EH7 What is the error message?
EH8 Error Type
EH9 Failed solution
Orientation
OR1 Is it possible to do this?
OR2 Search for explanations
OR3 Correct Logic
OR4 Best way
OR5 Specific doubt
OR6 What action do you recommend?
OR7 Others Orientation
GHA Learning
LE1 Learn a specific function
LE2 Documentation search
LE3 Others Learning
Functionality Implementation
FI1 Implementation goal
Insufficient implementation
II1 Not working as expected
Incompatibility
IN1 Work locally, fail in GHA
Altenative Solution
AS1 Looking for alternatives

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

[9] 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.

[10] 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.

[11] 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.

[12] A. H. Yahmed, A. A. Abbassi, A. Nikanjam, H. Li, and F. Khomh, "Deploying deep reinforcement learning systems: a taxonomy of chal-

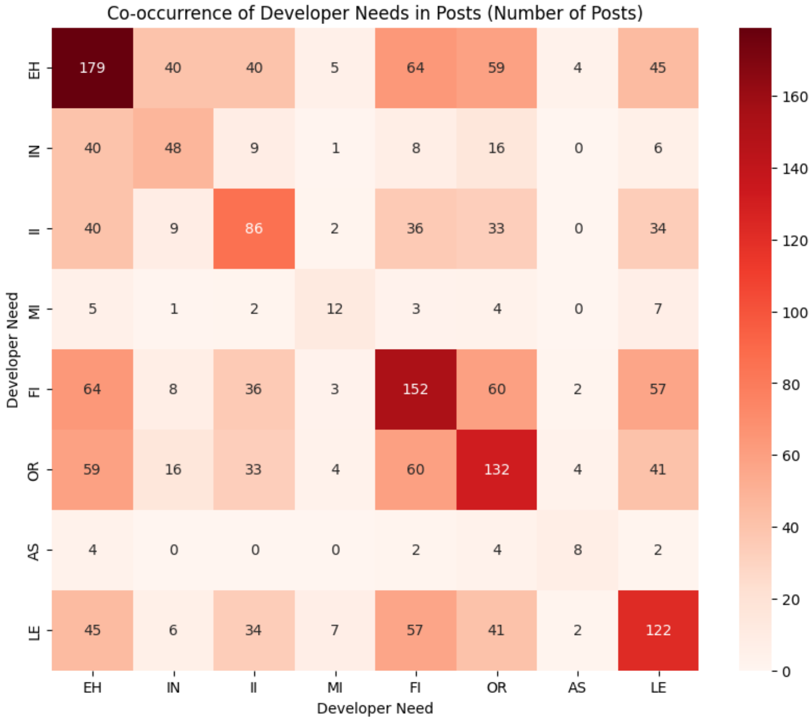

Fig. 3

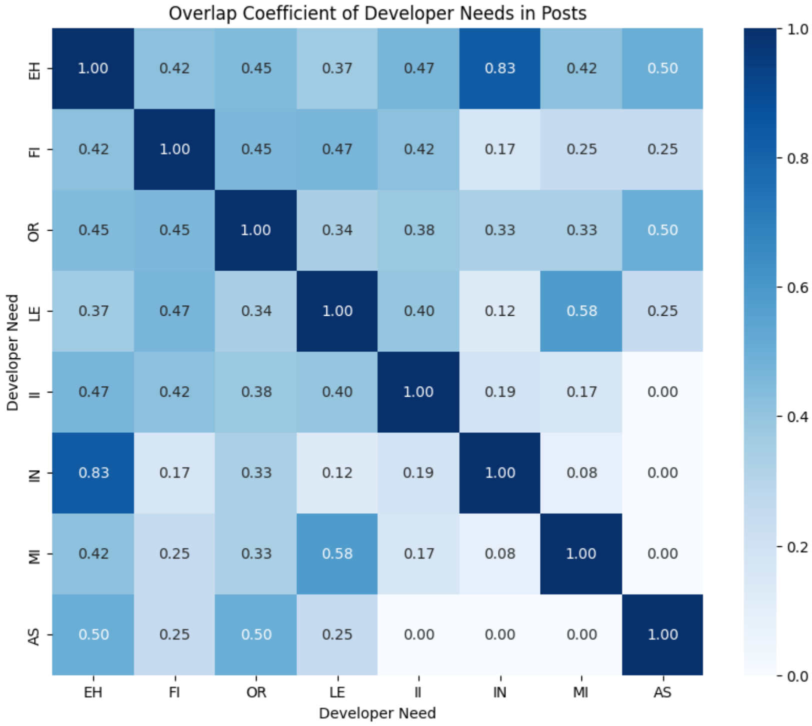

Fig. 4

lenges," in *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2023, pp. 26–38.

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

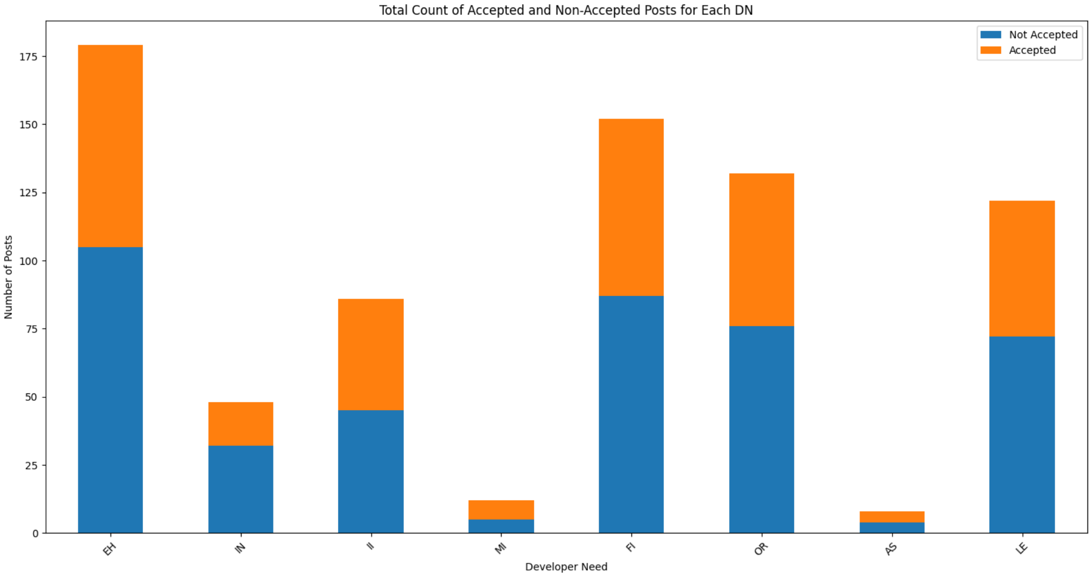

Fig. 5

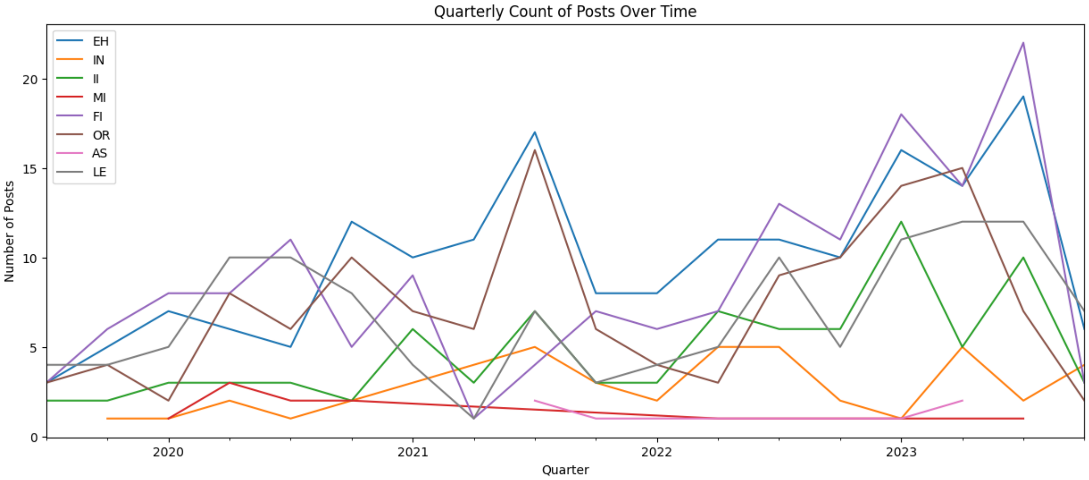

Fig. 6: Line plot showing the quarterly count of posts for each Developer Need (DN) category over time.

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

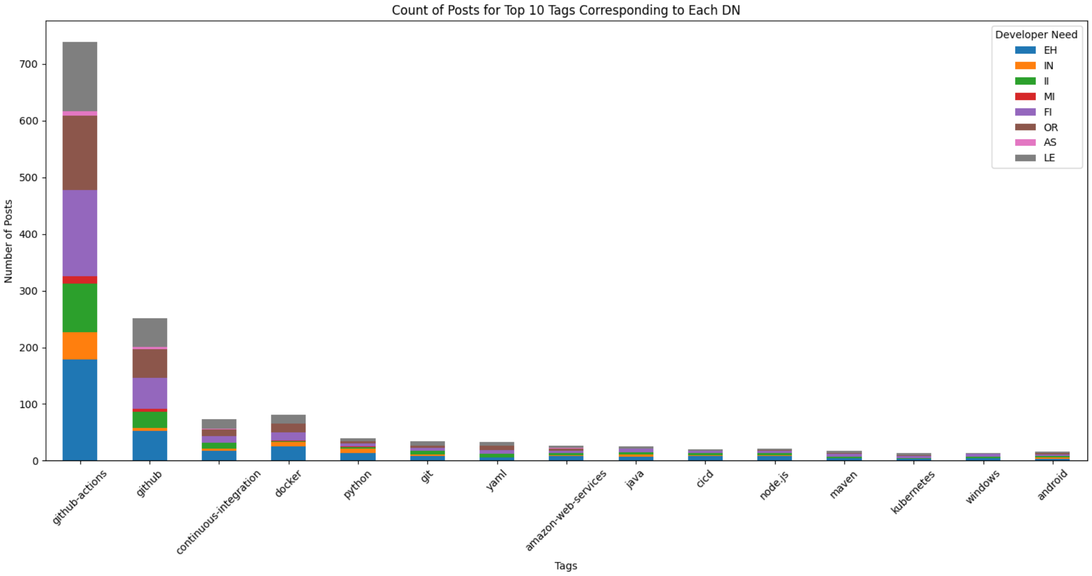

Fig. 7: Count of Posts for Top 10 Tags Corresponding to Each Developer Need (DN). The stacked bar plot shows the distribution of posts associated with each Developer Need (EH, IN, II, MI, FI, OR, AS, LE) across the top 10 most common tags.

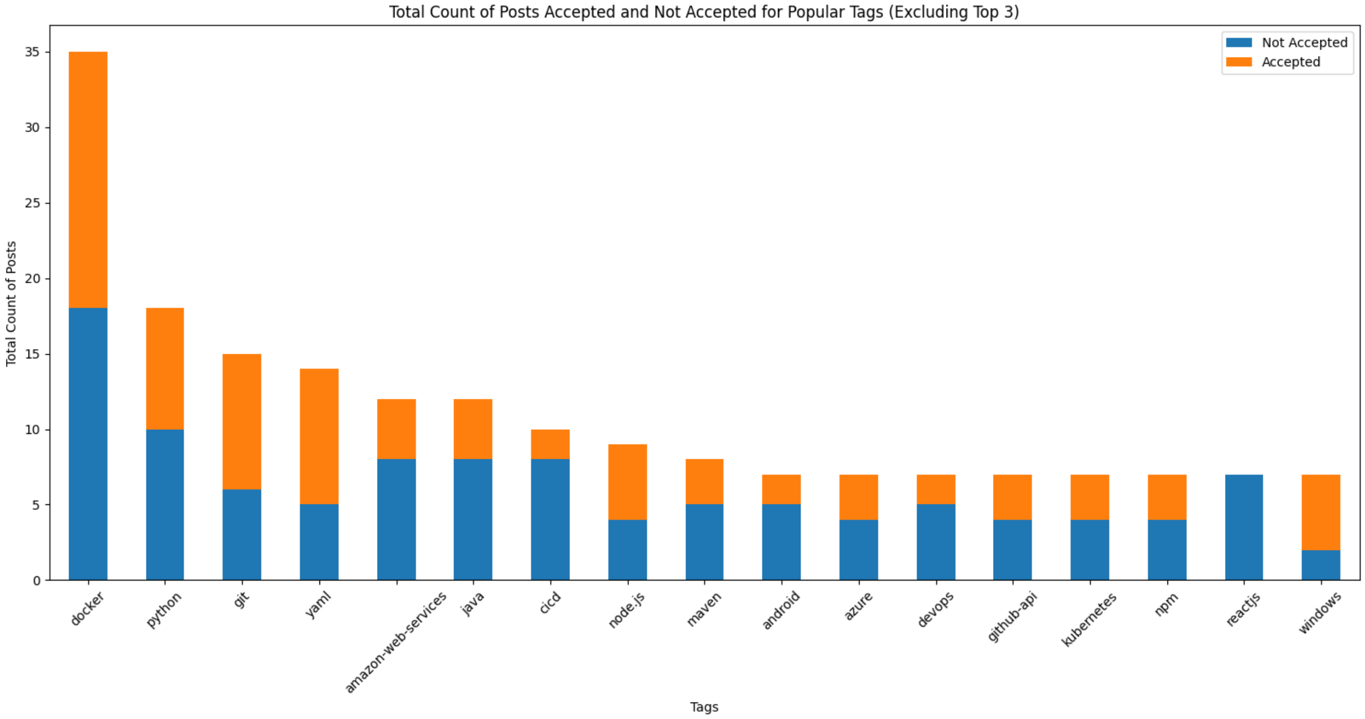

Fig. 8: Total Count of Posts Accepted and Not Accepted for Popular Tags (Excluding Top 3). This stacked bar plot shows the distribution of accepted and not accepted posts for the top popular tags, excluding the three most popular tags.

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