How Do Software Developers Use ChatGPT? An Exploratory Study on GitHub Pull Requests

Anonymous Author(s)*

ABSTRACT

Nowadays, Large Language Models (LLMs) play a pivotal role in software engineering. Developers can use LLMs to address software development-related tasks such as documentation, code refactoring, debugging, and testing. ChatGPT, released by OpenAI, has become the most prominent LLM. In particular, ChatGPT is a cutting-edge tool for providing recommendations and solutions for developers in their pull requests (PRs). However, little is known about the characteristics of PRs that incorporate ChatGPT compared to those without it and what developers usually use it for. To this end, we quantitatively analyzed 243 PRs that listed at least one ChatGPT prompt against a representative sample of 384 PRs without any ChatGPT prompts. Our findings show that developers use ChatGPT in larger, time-consuming pull requests that are five times slower to be closed than PRs that do not use ChatGPT. Furthermore, we perform a qualitative analysis to build a taxonomy of the topics developers primarily address in their prompts. Our analysis results in a taxonomy comprising 8 topics and 32 sub-topics. Our findings highlight that ChatGPT is often used in review-intensive pull requests. Moreover, our taxonomy enriches our understanding of the developer's current applications of ChatGPT.

CCS CONCEPTS

• **Software and its engineering** → **Collaboration in software development**.

KEYWORDS

Large Language Models, ChatGPT, Manual analysis, Mining Software Repositories, Pull Requests

ACM Reference Format:

Anonymous Author(s). 2018. How Do Software Developers Use ChatGPT? An Exploratory Study on GitHub Pull Requests. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The emergence of large language models (LLMs) has revolutionized the landscape of artificial intelligence and its integration into our daily lives. These advanced models, such as OpenAI's GPT-3, have demonstrated an unprecedented ability to understand and generate human-like text in a variety of contexts, including writing [25], translation [35], and content creation [32, 37], LLMs have become indispensable tools in many industries. Their ability to understand the natural language and generate context-sensitive responses has brought forward a new era of human-machine interaction, changing the way we communicate, work, and navigate in the digital world [34, 37, 43].

Software engineering is no exception to the adoption of LLMs where LLMs such as ChatGPT have been proven to be essential for understanding and generating code [31], documentation [33] and interactions in natural language [21]. This integration of LLMs into software development workflows not only remodels the coding landscape, but also contributes to the wider impact of these models on daily life [21].

Recent studies attempted to provide guidelines and patterns to help software engineers with prompting [41, 42]. These studies provided patterns and approaches on how to improve developers' prompting. However, little is known about the nature of questions that developers prompt ChatGPT with. Recently, OpenAI introduced the ChatGPT chat sharing feature [3] which allows users to share their ChatGPT chats using URLs. This feature made developers' collaboration more accessible in the context of pull request development.

The goal of this paper is to provide an understanding of how developers are currently using ChatGPT in the context of pull requests. We first quantitatively characterize pull requests with at least one ChatGPT link in terms of review effort and size. Then, we aim to identify the main issues that developers ask ChatGPT about. Thus, our research is guided by the following research questions:

RQ1. *How do pull requests with ChatGPT compare to pull requests without ChatGPT?*

RQ2. *What do developers request from ChatGPT in pull requests?*

To answer our research questions, we leveraged the recent dataset of DevGPT [1] consisting of a collection of ChatGPT links accompanied by their locations (*i.e.,* listed in pull requests, issues, and hacker rank forums). We focus the scope of this study on pull requests, and we analyze a set of pull requests from the DevGPT dataset that have at least one ChatGPT link. To answer RQ1, we compare these pull requests with a representative sample of pull requests randomly selected from 318K pull requests that do not involve ChatGPT links. The results indicate that pull requests with ChatGPT are larger and more review-intensive (*i.e.,* require more review effort) than those without ChatGPT links. To answer RQ2, we perform a thematic analysis [24], a commonly used technique in software engineering research [22, 23, 39] to characterize the topics of the questions that developers usually ask in their ChatGPT prompts. Our analysis reveals 8 topics (explanations, code generation, refactoring, bug fixing, DevOps, text generation, testing, and recommendations), that are associated with 32 sub-topics.

Replication package. We provide our custom scripts and data in a comprehensive replication package [2].

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 STUDY DESIGN

Our study is depicted in Figure 1 consisting of two main steps: (1) Data collection and (2) Data analysis.

2.1 Data collection

In this study, we used pull requests (PRs) data from the DevGPT dataset [1]. These 266 PRs represent the PRs with ChatGPT since they have at least one ChatGPT link in their descriptions/comments. As a next step, we mined all the pull requests of all repositories of the DevGPT pull requests. In total, we extracted 319,017 PRs.

2.2 Data analysis

2.2.1 Quantitative data analysis. To answer RQ1, we compare pull requests that contain at least a ChatGPT link (we refer to as *PRs with ChatGPT*) against pull requests without any ChatGPT link (*PRs without ChatGPT*). First, we filter the DevGPT data to only keep closed pull requests (*i.e.,* pull requests that are either merged or abandoned). After applying this filter, we ended up with 243 PRs from DevGPT that will constitute our experiment group. Thereafter, for each project that contains at least one of these PRs with ChatGPT, we collect all PRs without ChatGPT which account for a total of 318,674 PRs in total (We keep only closed PRs from the 319K+ set of PRs). Then, we randomly select a representative set of PRs (with a confidence level of 95%, and a confidence interval of 5%) resulting in 384 pull requests that will constitute our control group, similar to previous works [23, 26, 28, 40]. To compare PRs with ChatGPT and PRs without ChatGPT, we compute for each PR code review and PR size metrics [22, 28], namely, *NumberMessages*, *NumberInlineComments*, *NumberCommits*, *CodeChurn*, *Duration (hours)*, *NumberOfModifiedFiles*, and *DescriptionLength.* To investigate the statistical difference between both PR groups, we refer to the Mann-Whitney U test [29] since code review and pull request data is known to be highly skewed [22, 28]. The null hypothesis is that there is no variation between the metrics of both groups. In contrast, the alternative hypothesis indicates a significant difference in PR metrics with ChatGPT and the PRs without ChatGPT groups. Finally, we assess the effect size and the magnitude of the difference between the metrics of both groups using Cliff's delta (δ) effect size [27]. Following the recommendations of Romano et al. [38], Cliff's delta is interpreted as follows: Negligible if $|\delta| < 0.147$, small if $0.147 <= |\delta| < 0.33$, medium if $0.33 <= |\delta| < 0.474$ and large if $|\delta| >= 0.474$.

2.2.2 Qualitative data analysis. To answer RQ2, we adopt the thematic analysis technique following the guidelines of Cruzes et al. [30] to create a taxonomy of topics representing the main topic that developers discuss in their ChatGPT prompts. This approach involves analyzing the data to identify and develop themes ("topics"), within a collection of descriptive labels ("prompts purpose") as a technique commonly used in software engineering [22, 39]. Before starting the thematic analysis, we filter our 283 ChatGPT links by removing the links that are (*i*) deleted/no longer available, and (*ii*) the links that have another language than English. We ended up with 230 ChatGPT links. Our process for creating this taxonomy includes the following steps: 1) initial reading of ChatGPT prompts; 2) generation and refinement of initial descriptive labels; 3) review labels for merge opportunities; (4) translation of the labels into themes and creation of a model of higher-order themes and their sub-themes. The above-mentioned steps were performed independently by the first three authors as follows:

Step 1: Initial reading of ChatGPT prompts: In this step, the first three authors independently analyze all the ChatGPT prompts related to the pull requests to identify the main reason for the developer's conversation with ChatGPT.

Step 2: Generation and refinement of initial descriptive labels: Each author independently associates a prompt with a descriptive label. Once all prompt labels have been identified, the authors meet to discuss and refine the labels. A total of 255 labels were identified, with 61 prompts unavailable. Some of the identified labels were semantically equivalent and were subsequently merged, such as "code test" and "test", while two were conflicting ("solution recommendation", and "code generation") and required further discussion.

Step 3: Review labels for merge opportunities: We relate the labels identified in the first step to each other, and establish the link between them. After discussion with the co-authors, new labels were identified, such as "white drafts", labels were merged, such as "option execution management" and "option management", and others were relabeled. For example, since our analysis is not a one-step process, we have identified a new label related to "logging" and renamed "code security validation" to "security".

Step 4: Translate labels into themes: This step consists of identifying the generic themes that describe the grouped labels generated in the second step. This process identified 8 topics and 32 sub-topics that developers are discussing in their ChatGPT prompts.

3 STUDY RESULTS

3.1 RQ1: How do pull requests with ChatGPT compare to pull requests without ChatGPT?

Pull requests with ChatGPT links are more review intensive compared to pull requests without ChatGPT. Table 1 shows the statistical comparison of the code review metrics for pull requests with/without ChatGPT. The results indicate that the code review practices in pull requests using ChatGPT differ significantly from those of developers. In particular, developers exchange more messages and inline comments in pull requests with ChatGPT compared to pull requests without ChatGPT. From Table 1, we observe that pull requests with ChatGPT have more inline comments with a median of 2 significantly outnumbering the pull requests without ChatGPT having 0 inline comments in median with a medium effect size. Additionally, we observe that pull requests with ChatGPT have more commits (a median of 5 commits) compared to pull requests without ChatGPT (a median of 1 commits), with a medium effect size. Furthermore, we observe that pull requests with ChatGPT, take significantly longer duration (a median of 61.96 hours) compared to pull requests without ChatGPT (a median of 17.04 hours) with a small effect size.

Based on the obtained results, we speculate that developers often refer to ChatGPT in harder and more complex pull requests. These results are motivated by the finding in Table 1 in which we observe that ChatGPT pull requests are larger since they have significantly more code churn (code churn is 161 lines compared to 25 for pull requests without ChatGPT with a medium effect size). Additionally, pull requests with ChatGPT involve significantly more files (5

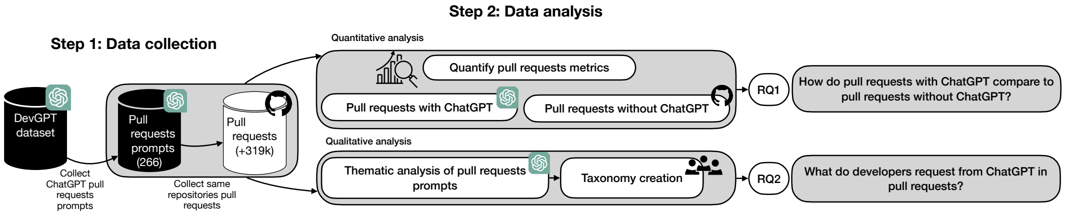

Figure 1: Approach overview.

files in median) than pull requests without ChatGPT (2 files in median) with a small effect size. Finally, we did not observe statistical difference in terms of description length (p-value is 0.46).

We provide in the following paragraph some examples of how ChatGPT is being used to help developers in large PRs. A first example is the pull request #3937 [8] from the *snapshot-labs/snapshot* [19] repository. This pull request is large since it has 7 modified files with 202 modified lines (130 additions and 72 deletions). In this example, the change's author intended to solve a spacing issue in their web pager. The reviewer commented about an issue with the developer's code. Both discussed the issue and then the reviewer provided an example from ChatGPT to convince the developer. Another example of successful use of ChatGPT is the pull request #105 [12] from the *code stream* [20]. This pull request involves changing 38 files with more than 1800 changed lines. In this example, the reviewer was confused about a changed line in the code, and the developer provided an example with ChatGPT to answer the reviewer's confusion.

3.2 RQ2: What do developers request from ChatGPT in pull requests?

Our manual analysis revealed 8 topics that developers discuss in their prompts, encompassing 32 sub-topics. As shown in Figure 2, our taxonomy consists of 8 generic topics: (1) Explanation, (2) Code generation, (3) Refactoring, (4) Bug fixing, (5) Text generation, (6) DevOps, (7) Testing, and (8) Recommendations.

(T1) Explanation: As shown in Figure 3, the most prevalent topic, namely "*Explanation*", accounts for 31% of developers' inquiries. Within this category, developers predominantly seek "Concepts/Techniques explanation", representing the majority with over 52%. This indicates a high number of developers need to understand and clarify various concepts and techniques. An example belonging to this subcategory can be found in the PR #897 [11], where a developer asked ChatGPT to explain the n8n tool. Additionally, 30% of the queries about "Translation", demonstrated a significant interest in linguistic or cross-language explanations, where ChatGPT is often used to check if a translation is correct[9]. "Code explanation" constitutes a minor portion, accounting for 1.45% of Explanation-related requests. Finally, "General questions" are posed in a small fraction, comprising 1.45% of inquiries from the Explanation topics.

(T2) Code Generation: Code generation stands as the second most frequently discussed topic, containing 20% of developers' prompts. Within this category, "Basic tasks" is the most prominent, accounting for 44% of the inquiries, indicating a high reliance on ChatGPT for generating code snippets related to fundamental programming tasks like JSON file parsing [7], REST APIs querying [4] and Strings processing [10]. "Web development" accounts for 24% of requests, highlighting a significant focus on generating code for web-related functionalities. "Production code" follows closely with 20%, indicating a substantial demand for generating production-ready code. In 8.89% of the cases, developers seek assistance in creating "White Drafts", which are illustrative code examples. Lastly, "UI design management" represents a minor fraction, comprising 2.22% of the Code Generation inquiries.

(T3) Refactoring: The third most discussed topic for developers using ChatGPT is related to "*Refactoring*", constituting 13% of the discussions. Most requests related to refactoring pertain to "Code optimization", accounting for 76%. In the PR #1775 [5], the developer gave a piece of code and requested optimization to make it run faster. Following this, requests are made for "Code restructuring", comprising 10%. The remaining 14% involve tasks such as renaming variables and functions, and cleaning up code.

(T4) Bug Fixing: The fourth most discussed topic is related to "*Bug fixing*" comprising 9%. Predominantly, in 95% of these cases, developers request ChatGPT to "Recommend fixes" for various issues. For instance, in the PR #301 [16], the developer explicitly states the issue and requested a fix: "*Why is that and can you fix it?*". The remaining 5%, of the cases, are requests related to "Debugging".

(T5) Text Generation: Emerging as the fifth most prevalent topic, "*Text generation*" accounting for 8% of developers' inquiries. Specifically, in around 95% of these cases, developers seek ChatGPT's assistance in "Documentation generation" tasks, where the model aids in creating comprehensive or changing the tone of the writing. For example, a developer asks ChatGPT to rewrite a text to make it sound positive [14]. In the remaining 5.26%, specific requests are related to "Grammar fixes".

(T6) DevOps: Occupying the sixth position in developers' topics of interest, "*DevOps*" comprises 8% of their inquiries. Within this category, developers seek guidance on a variety of tasks. Specifically, 26% of requests are related to "Configuration management", emphasizing the importance of managing and configuring system settings. In particular, we found Configuration management-related issues such as "Options management" (21%) and "Options execution order" (5%). For example, a developer seeks to understand the sequence of execution for configuration options derived from configuration files and command lines [18]as highlighted in Bessghaier et al. [23]. Additionally, "Dependencies management" and "CI/CD" hold equal weight, each accounting for 15.79%, suggesting a focus on managing project dependencies and implementing CI/CD practices. Furthermore, there are requests related to "Network Management" (10.53%). Lastly, "Containerization" represents (5.26%) of DevOps inquiries, indicating developers' interest in containerized deployment strategies.

(T7) Testing: We find "*Testing*" as the seventh most discussed topic, accounting for 38.46% of developers' inquiries. Within the

Table 1: Comparison of code review metrics with and without ChatGPT

Metric	With ChatGPT						Without ChatGPT						Statistical Analysis	
	min	Q1	Median	Mean	Q3	Max	min	Q1	Median	Mean	Q3	Max	P-value	Effect size (delta)
NumberMessages	0	0	1	6.76	5	148	0	0	1	2.46	2	70	0.00	Small (0.16)
NumberInlineComments	0	0	1	12.24	12	190	0	0	0	3.13	1	148	0.00	Small (0.32)
NumberCommits	1	2	5	12.31	11	161	1	1	1	4.56	4	231	0.00	Medium (0.44)
Code churn	0	44	161	2,260	587	182,39	0	5	25	498.55	143	29,797	0.00	Medium (0.41)
Duration (hours)	0.01	5.44	61.96	278.56	251.89	4,909	0.02	1.14	17.04	477.54	98.76	28,657	0.0	small (0.18)
NumberOfModifiedFiles	0	2	5	15.59	12	265	0	1	2	8.56	6	421	0.00	Small (0.27)
DescriptionLength	0	101	247	2,192	905	116,530	0	53	311	909.98	884	11,004	0.46	Negligible (0.03)

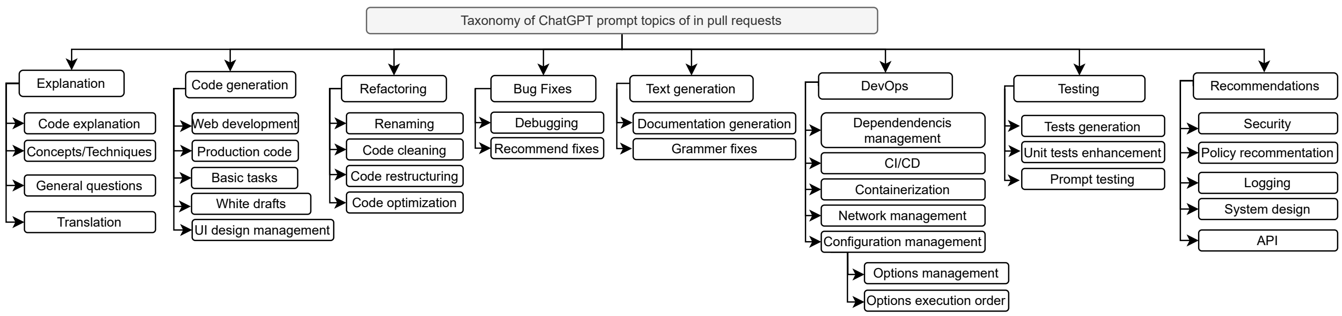

Figure 2: Taxonomy of the main topics developers discuss in their pull request prompts with ChatGPT.

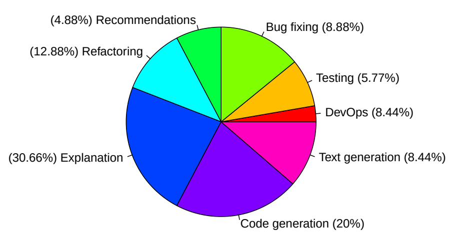

Figure 3: The partition of the 8 identified topics in pull requests related ChatGPT prompts.

testing cases, developers are particularly interested in "Tests generation", which constitutes 38.46% of the Testing category. for instance, developers request ChatGPT to generate tests for their regex expressions [15]. Additionally, developers seek to enhance existing code through "Unit tests enhancement", representing 23.08% of Testing-related inquiries. The remaining 38.46% is allocated to "Prompt testing", indicating an interest in evaluating and validating the effectiveness of prompts in code generation. For example, we found some cases where developers tested the quality of ChatGPT responses and, unexpectedly, expressed a recommendation to avoid utilizing ChatGPT, as explicitly stated: "*Example of why you shouldn't use ChatGPT...Everything except the last code block is hallucinated garbage.*" [6]

(T8) Recommendations: The last topic is related to "*Recommendations*" constituting 5% of prompts. "API" recommendations make up 36.36% of the category, indicating a high emphasis on seeking guidance and suggestions regarding API development. Furthermore, "System design", captures a substantial portion, with 27.27%, reflecting an interest in recommendations related to system architecture. Another portion, of 18.18%, pertains to "Policy recommendation", suggesting an interest in obtaining advice on recommended policies and strategies. For example, developers refer to ChatGPT to help define their backup policy[13]. Developers also show interest in "Logging", accounting for 9% of inquiries, indicating a focus on best practices for logging activities. Developers also seek guidance on "Security", representing 9% of the requests. For example, a developer provides ChatGPT with a code API and asks whether API users can breach the code and allow execution of arbitrary code [17].

4 TAKEAWAYS AND CONCLUSION

In this paper, we performed a quantitative and qualitative analysis to characterize PRs with ChatGPT. We performed a thematic analysis to categorize the topics of the developers' prompt for ChatGPT. Our analysis reveals 8 topics and 32 sub-topics, with code generation being the topic of the highest occurrence among the others. Based on the obtained results, we have the following takeaways:

Takeaway #1: Pull requests with ChatGPT are larger and more review intensive than pull requests without ChatGPT. Our study reveals that developers often use ChatGPT when dealing with more complex issues. This implies that ChatGPT is particularly useful for helping developers solve issues and conflicts in the context of pull request development. On the other hand, developers should be aware that ChatGPT might not provide accurate answers/recommendations. Therefore, developers should refer to ChatGPT wisely.

Takeaway #2: Developers are found to use ChatGPT as a third developer in their pull requests. Our qualitative analysis showcases that developers commonly leverage the capabilities of ChatGPT not just for basic tasks, but also for critical activities such as refactoring and bug fixing. Therefore, we encourage the researchers to take our taxonomy as a starting point to expand it and cover newly emerging topics. Notably, instances were found where developers asked from ChatGPT to design and code an entire project, surpassing its inherent capabilities, prompting responses such as *"I strongly recommend hiring a professional web developer or development team for this project"* [36]. Therefore, researchers and practitioners can explore opportunities to fine-tune LLMs with customized capabilities to meet the developer's expectations and examine developer satisfaction with its responses.

REFERENCES

[1] ___. 2023. The DevGPT dataset: https://github.com/NAIST-SE/DevGPT.
[2] ___. 2023. The reolication package for our study: https://figshare.com/s/2bc2c11e694997262a87.
[3] 2023. ChatGPT Shared Links FAQ. https://help.openai.com/en/articles/7925741-chatgpt-shared-links-faq. Accessed on: Dec 01, 2023.
[4] 2023. Pull Request #1. https://github.com/Wissididom/Bonbon-Reminder-StreamElements/pull/1 Accessed on: Dec 01, 2023.
[5] 2023. Pull Request #1775. https://github.com/ggerganov/llama.cpp/pull/1775 Accessed on: Dec 01, 2023.
[6] 2023. Pull Request #2. https://github.com/DogeisCut/extensions/pull/2 Accessed on: Dec 01, 2023.
[7] 2023. Pull Request #209. https://github.com/sCrypt-Inc/scryptTS-docs/pull/209 Accessed on: Dec 01, 2023.
[8] 2023. Pull Request #3937. https://github.com/snapshot-labs/snapshot/pull/3937 Accessed on: Dec 01, 2023.
[9] 2023. Pull Request #5602. https://github.com/rancher-sandbox/rancher-desktop/pull/5602 Accessed on: Dec 01, 2023.
[10] 2023. Pull Request #838. https://github.com/CodeIntelligenceTesting/cifuzz/pull/838 Accessed on: Dec 01, 2023.
[11] 2023. Pull Request #897. https://github.com/reworkd/AgentGPT/pull/897 Accessed on: Dec 01, 2023.
[12] 2023. Pull Request#1638. https://github.com/TeamCodeStream/codestream/pull/1638 Accessed on: Dec 01, 2023.
[13] 2023. Pull Request#219. https://github.com/pwncollege/dojo/pull/219 Accessed on: Dec 01, 2023.
[14] 2023. Pull Request#2230. https://github.com/faker-js/faker/pull/2230 Accessed on: Dec 01, 2023.
[15] 2023. Pull Request#2316. https://github.com/validatorjs/validator.js/pull/2316 Accessed on: Dec 01, 2023.
[16] 2023. Pull Request#301. https://github.com/OSRSB/OsrsBot/pull/301 Accessed on: Dec 01, 2023.
[17] 2023. Pull Request#40. https://github.com/rom-py/rompy/pull/40 Accessed on: Dec 01, 2023.
[18] 2023. Pull Request#8902. https://github.com/NixOS/nix/pull/8902 Accessed on: Dec 01, 2023.
[19] 2023. snapshot-labs/snapshot. https://github.com/snapshot-labs/snapshot Accessed on: Dec 01, 2023.
[20] 2023. TeamCodeStream. https://github.com/TeamCodeStream/codestream Accessed on: Dec 01, 2023.
[21] Aakash Ahmad, Muhammad Waseem, Peng Liang, Mahdi Fahmideh, Mst Shamima Aktar, and Tommi Mikkonen. 2023. Towards human-bot collaborative software architecting with chatgpt. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*. 279–285.
[22] Eman Abdullah AlOmar, Moataz Chouchen, Mohamed Wiem Mkaouer, and Ali Ouni. 2022. Code review practices for refactoring changes: An empirical study on openstack. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 689–701.
[23] Narjes Bessghaier, Mohammed Sayagh, Ali Ouni, and Mohamed Wiem Mkaouer. 2023. What Constitutes the Deployment and Run-time Configuration System? An Empirical Study on OpenStack Projects. *ACM Transactions on Software Engineering and Methodology* (2023).
[24] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.
[25] Tzeng-Ji Chen. 2023. ChatGPT and other artificial intelligence applications speed up scientific writing. *Journal of the Chinese Medical Association* 86, 4 (2023), 351–353.
[26] Moataz Chouchen, Ali Ouni, Raula Gaikovina Kula, Dong Wang, Patanamon Thongtanunam, Mohamed Wiem Mkaouer, and Kenichi Matsumoto. 2021. Anti-patterns in modern code review: Symptoms and prevalence. In *2021 IEEE international conference on software analysis, evolution and reengineering (SANER)*. IEEE, 531–535.
[27] Norman Cliff. 1993. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological bulletin* 114, 3 (1993), 494.
[28] Flávia Coelho, Nikolaos Tsantalis, Tiago Massoni, and Everton LG Alves. 2021. An empirical study on refactoring-inducing pull requests. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.
[29] William Jay Conover. 1999. *Practical nonparametric statistics*. Vol. 350. john wiley & sons.
[30] Daniela S Cruzes and Tore Dyba. 2011. Recommended steps for thematic synthesis in software engineering. In *2011 international symposium on empirical software engineering and measurement*. IEEE, 275–284.
[31] Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2023. Self-collaboration Code Generation via ChatGPT. *arXiv preprint arXiv:2304.07590* (2023).
[32] Hongyang Du, Zonghang Li, Dusit Niyato, Jiawen Kang, Zehui Xiong, Dong In Kim, et al. 2023. Enabling AI-generated content (AIGC) services in wireless edge networks. *arXiv preprint arXiv:2301.03220* (2023).
[33] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533* (2023).
[34] Abid Haleem, Mohd Javaid, and Ravi Pratap Singh. 2022. An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges. *BenchCouncil transactions on benchmarks, standards and evaluations* 2, 4 (2022), 100089.
[35] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT a good translator? A preliminary study. *arXiv preprint arXiv:2301.08745* (2023).
[36] OpenAI. 2023. Pull Request example:. https://chat.openai.com/share/cb1b4c4e-b2bc-4b1d-9df2-700be0cab72d Accessed on: Dec 01, 2023.
[37] Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* (2023).
[38] Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, and Jeff Skowronek. 2006. Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys. In *annual meeting of the Florida Association of Institutional Research*, Vol. 177. 34.
[39] Danilo Silva, Nikolaos Tsantalis, and Marco Tulio Valente. 2016. Why we refactor? confessions of github contributors. In *Proceedings of the 2016 24th acm sigsoft international symposium on foundations of software engineering*. 858–870.
[40] Dong Wang, Tao Xiao, Patanamon Thongtanunam, Raula Gaikovina Kula, and Kenichi Matsumoto. 2021. Understanding shared links and their intentions to meet information needs in modern code review: A case study of the OpenStack and Qt projects. *Empirical Software Engineering* 26 (2021), 1–32.
[41] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).
[42] Jules White, Sam Hays, Quchen Fu, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. *arXiv preprint arXiv:2303.07839* (2023).
[43] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* 10, 5 (2023), 1122–1136.