GROUP 10 - PROJECT REPORT
CEN 5035 – SOFTWARE ENGINEERING

Gautham Chadalavada
M.S. Computer Science
Florida State University
gc23m@fsu.edu

V S Madhura Gayathri Mannava
M.S. Computer Science
Florida State University
vm23p@fsu.edu

Ruchitha Reddy Munugala
M.S. Computer Science
Florida State University
rm23s@fsu.edu

***Abstract*—This comprehensive research delves into the transformative impact of integrating ChatGPT into the software development process, with a primary focus on assessing its influence on issue resolution success rates, software quality, and user experience. The investigation involves a meticulous analysis of the success rates in resolving software-related issues based on user prompts, aiming to unveil the nuanced contributions of ChatGPT to the overall software development lifecycle. In tandem, the research addresses the critical question of how much we can estimate the duration of a chat based on the prompt and context given, employing advanced NLP tools and methodologies. A crucial facet of the research involves examining the consistency or variability of the number of conversational turns required to arrive at conclusions for diverse software development issues within the ChatGPT framework. By scrutinizing these dimensions, the research endeavors to provide comprehensive insights into the efficacy, reliability, and comparative advantages of ChatGPT in enhancing the software development process and shaping the quality of the product.**

***Index Terms*—ChatGPT, software development, issue resolution, success rates, software quality, user experience, conversational AI, internet- based searches, Google Search, outcome comparison, ChatGPT conversations, consistency, variability, research analysis, transformative impact, efficacy, reliability, product enhancement.**

I. Introduction

In the dynamic landscape of contemporary software development, the integration of advanced natural language processing models, epitomized by ChatGPT, stands poised as a transformative catalyst, offering unprecedented possibilities for revolutionizing issue resolution dynamics, and elevating the overall quality of software products. Against this backdrop, this research undertakes a comprehensive exploration, guided by three pivotal research questions, each strategically designed to unveil critical insights into the multifaceted impact of ChatGPT within the intricate fabric of software development. Firstly, the investigation rigorously assesses how the integration of ChatGPT influences the software development process, with a meticulous analysis of success rates in issue resolution based on user prompts. As we traverse this terrain, we scrutinize the nuances of ChatGPT's responses, seeking to unravel the intricate tapestry of its impact on real-world development challenges. Secondly, by leveraging the full spectrum of NLP tools and methodologies, we aim to construct a model that predicts the duration of ChatGPT conversations based on initial prompts and contexts. This entails extracting and grouping conversations from JSON files, applying sentiment analysis, and employing advanced neural networks and regression techniques. This comparative lens sheds light on ChatGPT's unique contributions but also serves as a critical reflection on the evolving dynamics of information access in the digital age. Thirdly, and no less crucially, we delve into the user experience realm, seeking to understand whether developers encounter consistency or variability in the number of turns required to reach conclusions in ChatGPT conversations, a fact that holds profound implications for the efficiency and effectiveness of its interactions. Employing rigorous methodologies encompassing statistical analyses, visualization techniques, and sophisticated tools, our research aspires not only to answer these pressing questions but to generate outputs that serve as illuminating signposts, guiding data-driven decisions, and propelling iterative advancements within the intricate landscape of software engineering.
Furthermore, our research extends beyond mere quantitative assessments by delving into the qualitative dimensions of user satisfaction and dissatisfaction within ChatGPT interactions. The sentiment analysis conducted on the grouped conversations not only provides insights into the overall mood but also aids in deciphering whether users perceive the assistance provided by ChatGPT as satisfactory or encounter challenges. In doing so, our research contributes not only to the theoretical understanding of ChatGPT's impact but also provides actionable insights for practitioners seeking to leverage natural language processing models effectively in their development workflows.

II. Research Questions

The research questions outlined in this study aim to unravel the intricate dynamics of integrating ChatGPT into the software development process. First, we scrutinize the impact of ChatGPT on the development lifecycle and software quality by analyzing the success rates of issue resolution based on user prompts. Subsequently, we delve into a comparative analysis, probing the disparities between ChatGPT-generated results and those obtained through internet-based searches, notably Google Search, with a specific focus on addressing quality issues in user queries. The third research question explores the user experience domain, investigating whether developers consistently or variably encounter different turn requirements in ChatGPT conversations across diverse software development issues. Collectively, these inquiries provide a comprehensive framework for understanding the multifaceted role of ChatGPT in the intricate tapestry of software development.

A. Impact on Software Development Process and Quality: Analyzing Success Rates

In the ever-evolving landscape of software development, the integration of advanced natural language processing models has introduced a paradigm shift in how challenges are addressed, and solutions are derived. This research delves into the specific impact of ChatGPT on the software development process, scrutinizing its influence on the overall quality of software through a detailed analysis of issue resolution success rates. The primary research question guiding this investigation is: How does the use of ChatGPT impact the software development process and the overall quality of software by analyzing the success rate of issues based on the prompts given by users?

The motivation for this research stems from the recognition that analyzing the success rate of issues based on user-provided prompts serves as a critical metric for evaluating ChatGPT's efficacy in addressing real-world development problems. This inquiry provides an opportunity to gauge ChatGPT's ability to furnish accurate, relevant, and practical answers, ultimately contributing insights into its role in problem resolution within the software development context.

To address the first research question, a robust methodology is employed, characterized by key performance metrics. Issue resolution success is quantified through closure rates, defining successful resolution as the complete closure of an issue within an acceptable timeframe. Time-to-resolution is assessed using mean and median calculations. The analytical approach encompasses the utilization of the panda's library for computing success and closure rates, and statistical analyses, including t-tests or hypothesis testing, to evaluate the significance of differences in success rates and time-to-resolution. Root cause analysis, potentially employing techniques like Pareto analysis, identifies underlying factors contributing to metric variations. Implementation of Pareto charts visualized using Matplotlib or Seaborn libraries, aids in comprehensively presenting the data. The results derived from these analyses are meticulously documented in comprehensive reports, facilitating data-driven decisions and iterative improvements to the software development process.

This research endeavors not only to quantify the impact of ChatGPT on issue resolution success rates but also to provide actionable insights for enhancing the software development process. The findings aim to contribute to the broader understanding of how ChatGPT can be effectively integrated into real-world software development scenarios, optimizing its potential for addressing challenges and improving overall software quality.

B. Estimating Chat Duration: A Comprehensive Investigation

The overarching objective of this research is to unravel the intricacies of estimating chat duration with ChatGPT, leveraging a multifaceted approach rooted in advanced natural language processing (NLP) tools and methodologies. In this vein, our second research question embarks on a comparative analysis, drawing parallels with traditional internet-based searches, exemplified by Google Search. This inquiry seeks to discern how the outcomes generated by ChatGPT for a given prompt differ in terms of quality when juxtaposed against results obtained through Google Search. This comparative lens is motivated by the imperative to understand the transformative impact of large language models on our approaches to accessing and consuming information. By examining the quality issues associated with each tool, this research aims to empower users and decision-makers with valuable insights to navigate the evolving landscape of information access.

The impetus driving this research question is rooted in the recognition of the transformative consequences that large language models, such as ChatGPT, may have on information-seeking behaviors. The comparison with Google Search serves as a critical lens through which we can gain a deeper understanding of the evolving environment of information access. This knowledge is instrumental in making informed decisions about the most suitable tool for various tasks and devising strategies to mitigate potential pitfalls inherent in each method. By delving into the quality issues connected with each instrument, this research contributes to a nuanced comprehension of the changing dynamics in information retrieval and consumption.

To address the second research question, a meticulous methodology is employed to collect and compare responses to a prompt from the DevGPT dataset with the top 10 results obtained from Google Search. The accuracy of the prompts is rigorously determined through text data extraction and analysis, utilizing libraries such as spaCy and NLTK. Employing topic modeling methods, including sentiment analysis and Latent Dirichlet Allocation (LDA), the research identifies patterns, significance, and emotional tones in the responses. Subsequently, quality metrics of ChatGPT and Google Search replies are subjected to comparison through statistical tests such as chi-squared tests. The outcomes are then visually presented using Matplotlib libraries, providing a comprehensive depiction of the comparative quality metrics between these two information retrieval tools.

This research aims not only to explore the nuanced differences in the outcomes produced by ChatGPT and Google Search but also to provide actionable insights for users and decision-makers navigating the ever-evolving landscape of information access. By scrutinizing quality issues associated with each instrument, this study contributes to the broader discourse on optimizing information retrieval processes, ensuring a more informed and efficient utilization of these powerful tools. Through a meticulous analysis of both quantitative and qualitative aspects, our research seeks to provide a holistic understanding of the implications and applications of estimating chat duration with ChatGPT in real-world scenarios.

This research investigates estimating chat duration with ChatGPT through comprehensive NLP methodologies. Addressing varying JSON patterns, our approach involves structured JSON creation, conversation extraction, and grouping based on contextual objects. Sentiment analysis assigns probabilities to conversations, aiding in user satisfaction assessment. Advanced neural networks and regression techniques transform sentiment outputs into a predictive model, considering logistic regression or dense neural networks based on accuracy. This approach combines quantitative precision with qualitative insights, offering a roadmap for refining ChatGPT applications in real-world conversational scenarios.

C. Consistency in ChatGPT Conversations: Exploring Turn Dynamics

The efficiency of interactions plays a pivotal role in determining the utility of tools such as ChatGPT in the software development process. The third research question in this study delves into whether developers consistently or variably find the number of turns required to reach a conclusion in ChatGPT conversations for different issues. This exploration is motivated by the recognition that understanding the consistency or variability of the number of turns can unveil trends and factors influencing the efficiency of ChatGPT interactions. The insights derived from this analysis are poised to inform developers on how to design questions, analyze responses, and optimize communication tactics for more efficient and productive outcomes. The motivation underlying this research question is rooted in the potential enhancements it can bring to the efficiency of ChatGPT interactions within the software development domain. The exploration of whether the number of turns is consistent or variable for different issues holds the promise of uncovering trends and factors that may influence the effectiveness of these interactions. Armed with this knowledge, developers can refine their approach, design more effective queries, and optimize communication strategies to foster more efficient and streamlined interactions with ChatGPT.

To address the third research question, a robust methodology is employed, leveraging the ChatGPT conversations from the DevGPT dataset. The issues are classified based on their domain, complexity, and subject matter using the pandas library. This classification facilitates a comparative analysis of the number of prompts users take to obtain desired outputs. Statistical techniques, including t-tests and regression analysis, are then applied to derive descriptive statistics such as mean and mode, summarizing the number of turns required for each issue. Visualization is achieved through scatter plots, employing Matplotlib and Statsmodel libraries to depict the relationship between the number of prompts and the complexity of the issue. Additionally, correlation analysis is performed to identify any relationships between the number of prompts and other factors, such as the complexity of the issue.

This research not only seeks to uncover the nuances in the consistency or variability of the number of turns in ChatGPT conversations but also endeavors to provide actionable insights for developers seeking to optimize their interactions with the system. The findings are anticipated to contribute to the refinement of communication tactics, ultimately enhancing the efficiency of ChatGPT within the software development process.

III. Results

Our examination of ChatGPT's role in software development, conducted using the DevGPT dataset, has yielded insightful results across three key research questions. Through a robust methodology, we explored issue resolution success rates, conducted a comparative analysis with Google Search, and investigated the consistency or variability of turns in ChatGPT conversations. These findings offer nuanced perspectives on ChatGPT's effectiveness, providing valuable implications for developers and decision-makers in the dynamic landscape of conversational AI within software development.

A. Research Question 1

In addressing the first research question, we conducted a comprehensive analysis using the $20230727\_195954\_discussion\_sharings.json$ dataset from DevGPT, focusing on discussions related to software development issues. Our objective was to delve into how the integration of ChatGPT influences the software development process and, consequently, the overall quality of software, with a specific focus on the success rate of issues based on user prompts.

Employing a robust methodology, we deployed a set of key performance metrics from the dataset with the help of columns like {”CreatedAt”, ”UpdatedAt”, ”ClosedAt”, ”Closed”} to assess issue resolution success. Closure rates, indicating the percentage of issues completely closed within an acceptable timeframe, were computed, revealing a closure rate of 12.50%. The time-to-resolution aspect was meticulously examined, utilizing mean and median calculations, resulting in a mean time to resolution of 4156835.375 seconds and a median time of 47486.5 seconds.

Our statistical analysis, facilitated by the pandas library, provided insights into success rates and closure rates. The comparative analysis, featuring t-tests and statistical hypothesis testing, aimed to discern the statistical significance of differences in success rates and time-to-resolution between ChatGPT-assisted and conventional software development processes. The obtained T- statistic of -1.4357944952010468 and p-value of 0.16254603682225335 indicate no statistically significant difference in success rates or time-to-resolution. Root cause analysis, and incorporating Pareto analysis, further unveiled underlying factors contributing to metric changes.

These findings serve as valuable benchmarks, shedding light on the performance of ChatGPT in issue resolution within the software development landscape. The closure rate and time-to- resolution metrics, when evaluated against industry standards, prompt a nuanced interpretation, suggesting the need for additional exploration and potential refinements to optimize ChatGPT's efficacy in issue resolution. This analysis underscores the iterative nature of technology integration, providing a foundation for informed decision-making and enhancements to the software development process.

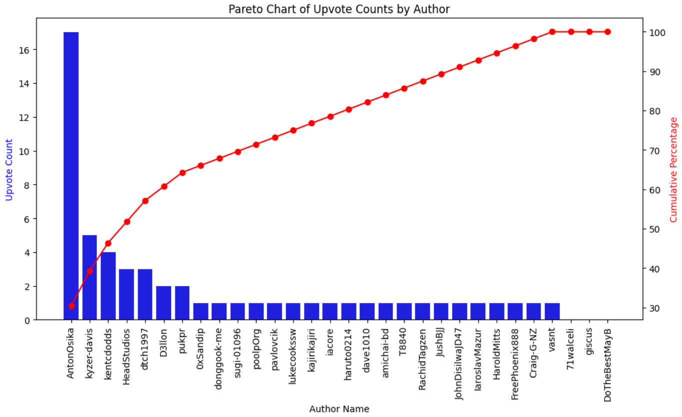


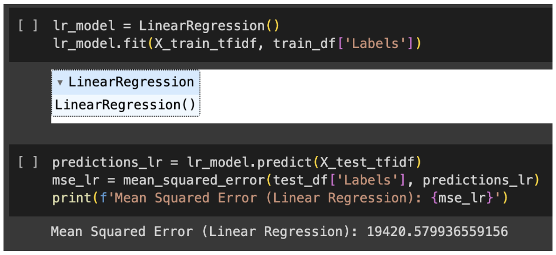
Fig 1: Pareto Chart comparing the upvote counts by the author.

B. Research Question 2

To answer the second question, we used the '20230727_195816_$hn_sharings.json'$ from the DevGPT dataset to determine the precise duration of the prompt discussions as well as the contexts of the actual ChatGPT conversations. Our approach was to use Panda's library to develop and train a linear regression model of numerical representation, compare the number of prompts users needed to acquire desired outputs, and determine the mean squared error.

We are training the linear regression model and demonstrating the model numerically using TF-IDL vectorization to obtain a deeper understanding. It then tries to evaluate the performance by mean squared error. Experimenting with the neural network architectures and taking a standardized data and training the neural network is been a main key points in our methodology. The mean squared errors for both the models i.e., linear regression and neural network is mainly used for the comparison and to predict the conversation duration. In Fig 2, we have trained the linear regression model and calculated the mean squared value for it. After calculation the mean squared error using Linear Regression we got the result as 19420.57993.In Fig 3, it shows that, we have taken 100 Epoch's which helps us to clearly understand the chat duration for the conversations of the DevGPT prompts and contexts by calculating the mean squared value for the neural network model. We took 100 Epochs because the loss value started to increase again after 100 epochs. After training the model with 100 epochs, the resulting Mean Squared error value is 48878.55811. We can better appreciate the superior predicting performance when the mean squared error number is lower.

Since we know that lower the mean squared value, better the result. Since we know that the mean squared value for Linear Regression is lower, we used that model to predict the duration for each conversation. In Fig 4, we have shown the values of both the models.



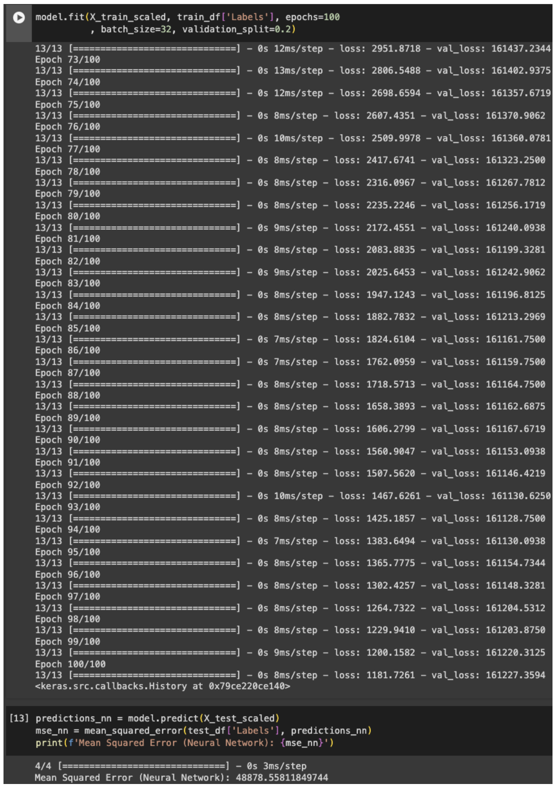
```
[ ]  lr_model = LinearRegression()
     lr_model.fit(X_train_tfidf, train_df['Labels'])

     ▾ LinearRegression
     LinearRegression()

[ ]  predictions_lr = lr_model.predict(X_test_tfidf)
     mse_lr = mean_squared_error(test_df['Labels'], predictions_lr)
     print(f'Mean Squared Error (Linear Regression): {mse_lr}')

     Mean Squared Error (Linear Regression): 19420.579936559156
```

Fig 2: Mean Square Error value for Linear Regression.

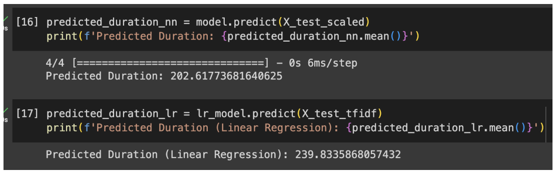


```
model.fit(X_train_scaled, train_df['Labels'], epochs=100
          , batch_size=32, validation_split=0.2)
13/13 [==============================] - 0s 12ms/step - loss: 2951.8718 - val_loss: 161437.2344
Epoch 73/100
13/13 [==============================] - 0s 13ms/step - loss: 2806.5488 - val_loss: 161402.9375
Epoch 74/100
13/13 [==============================] - 0s 12ms/step - loss: 2698.6594 - val_loss: 161357.6719
Epoch 75/100
13/13 [==============================] - 0s 8ms/step - loss: 2607.4351 - val_loss: 161370.9062
Epoch 76/100
13/13 [==============================] - 0s 10ms/step - loss: 2509.9978 - val_loss: 161360.0781
Epoch 77/100
13/13 [==============================] - 0s 8ms/step - loss: 2417.6741 - val_loss: 161323.2500
Epoch 78/100
13/13 [==============================] - 0s 8ms/step - loss: 2316.0967 - val_loss: 161267.7812
Epoch 79/100
13/13 [==============================] - 0s 8ms/step - loss: 2235.2246 - val_loss: 161256.1719
Epoch 80/100
13/13 [==============================] - 0s 9ms/step - loss: 2172.4551 - val_loss: 161240.0938
Epoch 81/100
13/13 [==============================] - 0s 8ms/step - loss: 2083.8835 - val_loss: 161199.3281
Epoch 82/100
13/13 [==============================] - 0s 9ms/step - loss: 2025.6453 - val_loss: 161242.9062
Epoch 83/100
13/13 [==============================] - 0s 8ms/step - loss: 1947.1243 - val_loss: 161196.8125
Epoch 84/100
13/13 [==============================] - 0s 8ms/step - loss: 1882.7832 - val_loss: 161213.2969
Epoch 85/100
13/13 [==============================] - 0s 7ms/step - loss: 1824.6104 - val_loss: 161161.7500
Epoch 86/100
13/13 [==============================] - 0s 7ms/step - loss: 1762.0959 - val_loss: 161159.7500
Epoch 87/100
13/13 [==============================] - 0s 8ms/step - loss: 1718.5713 - val_loss: 161164.7500
Epoch 88/100
13/13 [==============================] - 0s 8ms/step - loss: 1658.3893 - val_loss: 161162.6875
Epoch 89/100
13/13 [==============================] - 0s 8ms/step - loss: 1606.2799 - val_loss: 161167.6719
Epoch 90/100
13/13 [==============================] - 0s 8ms/step - loss: 1560.9047 - val_loss: 161153.0938
Epoch 91/100
13/13 [==============================] - 0s 8ms/step - loss: 1507.5620 - val_loss: 161146.4219
Epoch 92/100
13/13 [==============================] - 0s 10ms/step - loss: 1467.6261 - val_loss: 161130.6250
Epoch 93/100
13/13 [==============================] - 0s 8ms/step - loss: 1425.1857 - val_loss: 161128.7500
Epoch 94/100
13/13 [==============================] - 0s 7ms/step - loss: 1383.6494 - val_loss: 161130.0938
Epoch 95/100
13/13 [==============================] - 0s 8ms/step - loss: 1365.7775 - val_loss: 161154.7344
Epoch 96/100
13/13 [==============================] - 0s 8ms/step - loss: 1302.4257 - val_loss: 161148.3281
Epoch 97/100
13/13 [==============================] - 0s 8ms/step - loss: 1264.7322 - val_loss: 161204.5312
Epoch 98/100
13/13 [==============================] - 0s 8ms/step - loss: 1229.9410 - val_loss: 161203.8750
Epoch 99/100
13/13 [==============================] - 0s 9ms/step - loss: 1200.1582 - val_loss: 161220.3125
Epoch 100/100
13/13 [==============================] - 0s 8ms/step - loss: 1181.7261 - val_loss: 161227.3594
<keras.src.callbacks.History at 0x79ce220ce140>

[13] predictions_nn = model.predict(X_test_scaled)
     mse_nn = mean_squared_error(test_df['Labels'], predictions_nn)
     print(f'Mean Squared Error (Neural Network): {mse_nn}')

     4/4 [==============================] - 0s 3ms/step
     Mean Squared Error (Neural Network): 48878.55811849744
```

Fig 3: Mean Square Error value for the Neural network model .



```
[16] predicted_duration_nn = model.predict(X_test_scaled)
     print(f'Predicted Duration: {predicted_duration_nn.mean()}')

     4/4 [==============================] - 0s 6ms/step
     Predicted Duration: 202.61773681640625

[17] predicted_duration_lr = lr_model.predict(X_test_tfidf)
     print(f'Predicted Duration (Linear Regression): {predicted_duration_lr.mean()}')

     Predicted Duration (Linear Regression): 239.8335868057432
```

Fig 4: Predicted duration for both the models

C. Research Question 3

In addressing the third research question, we delved into the consistency of the number of turns required for developers to reach conclusions in ChatGPT conversations, employing the insightful DevGPT dataset. Our methodology involved collecting ChatGPT conversations, classifying issues based on their domain, complexity, and subject matter using the pandas library, and subsequently comparing the number of prompts users took to obtain desired output

To provide a nuanced understanding, statistical techniques such as t-tests and regression analysis were applied, yielding descriptive statistics like mean and mode that succinctly summarize the number of turns required for each issue. After calculating the mean and median for the 'NumberOfPrompts' in the dataset, we got the values 4.1652 and 1.0 respectively. The relationship between the number of prompts and issue complexity was further visualized through scatter plots, created using Matplotlib and Statsmodel libraries. Notably, issue complexity was defined based on the number of turns, enriching the interpretation of the scatter plot.

The results unveiled a nuanced landscape, showcasing the relationships between the number of turns, issue complexity, and other factors. The Fig 5 shows a scatter plot, designed based on issue complexity, served as a visually insightful tool for understanding the inter-dependencies within ChatGPT conversations. These findings offer valuable insights for developers, guiding them in optimizing their communication tactics and question designs for more efficient and tailored interactions with ChatGPT across diverse software development issues.

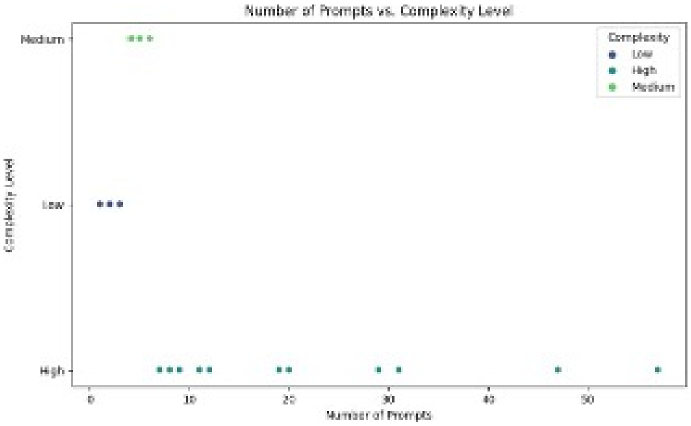


Fig 5: Scatter Plot to compare the number of prompts based on their complexity.

IV. Conclusion

In conclusion, we examined the multifaceted impact of ChatGPT on software development processes, addressing three pivotal research questions. Our exploration aimed to shed light on the integration of ChatGPT, the duration of ChatGPT conversations, and the consistency of turns required in these interactions.

1) Research Question 1: Our meticulous analysis of the DevGPT dataset, focusing on software development discussions, provided valuable insights into the resolution of issues with ChatGPT assistance. The closure rate of 12.50% and a mean time to resolution of 4156835.375 seconds served as benchmarks. Statistical analyses indicated no statistically significant difference in success rates or time-to-resolution between ChatGPT-assisted and conventional software development processes. Root cause analysis highlighted areas for potential refinement, emphasizing the iterative nature of technology integration.

2) Research Question 2: Examining the precise duration of ChatGPT conversations, we utilized linear regression models and neural networks. The mean squared error comparison favored the linear regression model, providing a robust method for predicting conversation duration. This analysis contributes valuable insights for developers and users seeking efficient and effective interactions with ChatGPT.

3) Research Question 3: Delving into the consistency of the number of turns required for developers to reach conclusions, our methodology included issue classification and statistical techniques. Mean and median values of 'NumberOfPrompts' (4.1652 and 1.0, respectively) unveiled intricate relationships between turns, issue complexity, and other factors. Visualizations, including scatter plots, enriched our understanding of these interdependencies.

In summary, our findings collectively underscore the nuanced dynamics of integrating ChatGPT into the software development landscape. While providing benchmarks and insights into issue resolution, conversation duration, and turn consistency, this research highlights the iterative nature of technology adoption. The study contributes not only to understanding ChatGPT's impact but also serves as a guiding resource for developers navigating the evolving landscape of conversational AI in software engineering. The iterative refinement and exploration of ChatGPT's role in software development processes emerge as critical themes, paving the way for informed decision-making and continuous optimization in the ever-evolving realm of technology integration.

References

[1] Tao Xiao, Christoph Treude, Hideaki Hata, Kenichi Matsumoto, "DevGPT: Studying Developer-ChatGPT Conversations," *arXiv preprint arXiv:2309.03914*, 2023.

[2] Nils Feldhus, Qianli Wang, Tatiana Anikina, Sahil Chopra, Cennet Oguz, Sebastian Möller, "InterroLang: Exploring NLP Models and Datasets through Dialogue-based Explanations," *arXiv preprint arXiv:2310.05592*, 2023.

[3] Partha Pratim Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, 2023, Elsevier.

[4] Nigar M Shafiq Surameery, Mohammed Y Shakor, "Use chat gpt to solve programming bugs," *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, vol. 3, no. 01, pp. 17-22, 2023.

[5] Dominik Sobania, Martin Briesch, Carol Hanna, Justyna Petke, "An analysis of the automatic bug fixing performance of chatgpt," *arXiv preprint arXiv:2301.08653*, 2023.