**Comparing the Abilities of AI Assistants in Web Development:**

**Using OpenAI ChatGPT-3.5 and GitHub Copilot**

Rylan Casanova

College of Technology, Grand Canyon University

CST-461: Current Trends in Computer Science

Dr. Christine Bakke

April 28, 2024

**Abstract**

This study investigates the comparative capabilities of GPT-3.5 and GitHub Copilot in coding assistance and communication within the context of Scrum and Agile methodologies. Drawing upon a review of peer-reviewed literature, which emphasizes effective Scrum Master roles, management techniques in Agile methodologies, and considerations of AI assistant capabilities and limitations, this research employs a mixed-methods approach to analyze Likert scores and qualitative observations. Results suggest GitHub Copilot's superior communication effectiveness and proficiency in handling tasks across multiple files, while GPT-3.5 exhibits a slight edge in smaller, more specific tasks, with both assistants demonstrating comparable debugging accuracy. These findings contribute to a deeper understanding of AI-powered coding tools and their potential in optimizing productivity and efficiency in software development processes, thereby providing valuable insights for developers and researchers alike.**Introduction**

In my comprehensive review of peer-reviewed articles, I explored various methodologies concerning the integration of AI assistants into code development processes, particularly within the framework of Scrum or Agile methodologies used in coding classes. This literature review serves as an introduction and overview of some of the research methodologies I learned from while completing my comparison of AI assistants’ abilities in web development using ChatGPT-3.5 and GitHub Copilot.

The literature reviewed provided valuable insights into the role of a Scrum Master [1] in effectively managing a development team [2], which in this case includes AI assistants. Effective communication practices were highlighted as crucial in order to maximize the potential of AI assistants within the Scrum framework. Additionally, the literature emphasized the significance of Scrum and Agile methodologies in managing various aspects of the development process, including time, scope, quality, risk, and resource management [3], which extends to the management of AI assistants.

Considering the capabilities and limitations [5] of AI assistants was key to my research. The literature wrote about the importance of understanding these factors to optimize the utilization of AI assistants effectively. Throughout the review and conducting my research, it became evident that proper communication practices are essential not only for human team members but also for AI assistants [6]. By integrating effective communication strategies, such as clear task assignments and progress updates (e.g. Jira/Kanban boarding), the collaboration between human developers and AI assistants can be significantly enhanced.

Furthermore, the literature revealed how AI assistants can play a pivotal role in code generation [7] and web app development [8]. Several studies demonstrated the potential of AI assistants in automating repetitive tasks, facilitating faster development cycles, and improving overall productivity in software development projects.

The literature review involved not only summarizing existing research but also synthesizing and analyzing the findings to draw meaningful insights. By overviewing a broader range of sources and delving deeper into the research-focus topics, the review provides a comprehensive understanding of the role of AI assistants in code development within the context of Scrum and Agile methodologies, setting the stage for my comparative analysis of ChatGPT-3.5 and GitHub Copilot in web development.

**Methods**

**Research Hypotheses/Objectives:**

I have three research hypotheses to help guide my research objectives and methodologies by focusing on key aspects of AI assistant performance in coding tasks, specifically addressing coding ability, debugging accuracy, and effectiveness of communication between GPT-3.5 and GitHub Copilot. Through structured observations, Likert scale ratings, and qualitative analysis, I aim to investigate and compare these aspects to determine any significant differences between the two AI assistants.

* H0: There is a significant difference in the coding ability between GPT-3.5 and GitHub Copilot
* H1: The debugging accuracy of GPT-3.5 significantly differs from the debugging accuracy of GitHub Copilot
* H2: There is a significant difference in the effectiveness of communication between GPT-3.5 and GitHub Copilot

**Mixed Methods Research Framework:**

My methodology employs a mixed methods research framework to comprehensively assess the performance of AI assistants in web development tasks. This approach combines quantitative data, such as Likert scores and keywords, with qualitative data including observations, notes, and identification of overall trends and themes.

**Quantitative Data:**

I utilize a 1 through 5-star Likert scale to quantify the performance of AI assistants across various tasks. Each response by the AI platforms is rated on a Likert scale ranging from 1 (poor) to 5 (excellent) based on efficiency and effectiveness in handling the assigned tasks. Additionally, I analyze keywords extracted from the responses to identify patterns and trends in performance.

**Qualitative Data:**

Observations and notes are collected during the AI tests to provide qualitative insights into their performance. I pay close attention to the context, depth, and relevance of the responses generated by each AI assistant. This qualitative data allows me to gain a deeper understanding of the strengths and weaknesses of each assistant in different aspects of web development.

**Partially Structured Observations:**

My approach to conducting AI tests involves partially structured observations. Tasks are assigned equally among both assistants, and prompts are open-ended to allow for a range of responses. The observations collected are examined for patterns, similarities, differences, and emerging trends in the performance of the AI assistants.

**Experimental Design:**

I follow a set of procedures for recording and grading observations, particularly when using Likert scales. This ensures consistency and reliability in my assessments. By structuring my experimental design, I aim to minimize biases and variations in the evaluation process, allowing for a more accurate comparison of AI assistant performance.

**Results and Analysis**

I recorded 32 table entries (prompts, results, notes, and scores for each AI assistant)

A graph with blue and white lines

Description automatically generated A graph with blue and white bars

Description automatically generated

*Graphical representation of individual AI Likert data that was recorded during trials*

*A graph with blue and white bars

Description automatically generated*

*Graphical representation of overall combined AI Likert Rankings*

**ChatGPT-3.5 Likert Analysis:**

* Mean: 3.75 stars
* Median: 4 stars
* Mode: 4 stars

**GitHub Copilot Likert Analysis**

* Mean: 4.00 stars
* Median: 4 stars
* Mode: 5 stars

**Thematic Observations + Analysis**

* Copilot never scored below a 2
* Both AIs gave a functioning response a majority (>50%) of the time
* Copilot’s Likert data having a mode of 5 stars demonstrates its high level of communication
* Copilot is better than ChatGPT with tasks that require modification of multiple files
* ChatGPT has a slight edge over Copilot with tasks that are smaller and more specific

**Group Data Analysis**

As a group with Kyungchan Im and Caleb Klinger, we compared and combined AI assistant data which we collected over the course of our individual research endeavors. Our research objectives and hypotheses were generally aimed toward achieving the same goal, and our group analysis focused on the Likert scale and thematic analysis of the three AI’s we used – ChatGPT-3.5, ChatGPT-4.0, and GitHub Copilot.

**Analysis 1: Likert Scale**

A graph of a number of different types of graphs

Description automatically generated with medium confidence

The analysis of the two AI platforms, ChatGPT 4.0 and ChatGPT 3.5, along with GitHub Copilot, was conducted based on three key parameters: code generation, answer quality, and efficiency. Each AI was tested on a variety of tasks and evaluated using a Likert scale where scores of 4 and 5 indicate high efficiency and performance.

**ChatGPT 4.0**

* Code Generation: Likely to score 5, considering it is designed to handle a variety of complex tasks, including programming-related queries.
* Answer Quality: Potentially an average 4, given its advanced understanding and response capabilities.
* Efficiency: Would be expected to score high, around 4 or 5, because it should require fewer prompts to arrive at a high-quality output.

**ChatGPT 3.5**

* Code Generation: Mostly scored around 4 to 5, as it's capable but less refined compared to GPT-4.0, with less context understanding for complex coding tasks.
* Answer Quality: Could be a 4, given that it was at the time one of the leading models for generating accurate and relevant text-based responses.
* Efficiency: Might be a 3 to 4, since it might need more iterations or clarifications for certain tasks compared to GPT-4.0.

**GitHub Copilot**

* Code Generation: Average score 3.5, since it is specifically trained on a corpus of code and is optimized for this task.
* Answer Quality: For coding questions, it would be a 4. However, for non-code related queries, it would not be applicable as it's not designed for general text generation.
* Efficiency: In terms of generating code, it would score a 4, as it provides code suggestions in real-time as the user types, which is highly efficient for the intended use case.

**Analysis 2: Theme Analysis**

Our theme analysis of the three AI assistants consisted of creating word clouds for each AI model and use of an ANOVA large language model which is discussed in a later section. The word clouds are generated in order to highlight the frequency of different words in our AI assistant’s tasks, queries, responses, and other observations/notes.

**ChatGPT 4.0 Theme Word Cloud**

A word cloud of text

Description automatically generated

**ChatGPT 3.5 Theme Word Cloud**

A close-up of words

Description automatically generated

**GitHub Copilot Theme Word Cloud**

A close-up of words

Description automatically generated

**ANOVA Word Cloud Analysis**

This word cloud is generated by grabbing user questions and performing words count analysis of generated answers. We used a large language model to scrutinize and analyze the theme of each user's questions. From there, we can sort out the themes of each question and basically press it as a word cloud. We could not really see the significant difference between the efficiency of individual AI copilot with different tasks, so we performed ANOVA one-way testing.

A black screen with white numbers

Description automatically generated

The ANOVA statistic evaluated with 0.502 that the differences between group means are not much larger than the variability of performance ratings within those groups themselves.

P value is 0.606 means our hypothesis fails to reject the null hypothesis that the interaction between AI and Theme does not have a statistically significant effect on the performance ratings.

**Conclusion**

In conclusion, the research findings reveal nuanced differences in the capabilities of GPT-3.5 and GitHub Copilot in coding assistance and communication. While GitHub Copilot demonstrates superiority in communication effectiveness in context and handling tasks across multiple files, GPT-3.5 shows a slight advantage in smaller, more specific tasks. Copilot’s ability to use context and give a working response in a significantly lower average amount of prompts demonstrates its effectiveness of communication. However, both assistants exhibit comparable performance in debugging accuracy. These insights contribute to a better understanding of AI-powered coding tools and can guide developers and researchers in leveraging these technologies to enhance productivity and efficiency in software development workflows.

**Group Data Analysis Conclusion**

Our research aimed to investigate performance variations among different AI platforms in various tasks of the application development process. Formulating hypotheses guided our study, examining whether significant differences exist in AI assistant performance. Through structured observation and evaluation, including Likert scale ratings and qualitative analysis, we assessed the performance of ChatGPT 4.0, ChatGPT 3.5, and GitHub Copilot. While there were observable differences in performance, especially between ChatGPT-4.0 and others, statistical analysis revealed that these differences were not statistically significant. Thus, our study provides valuable insights into AI assistant performance in web app development tasks, but further research is needed to understand their practical implications fully.

**References**

[1] Paasivaara, M. (2021, May 1). *Teaching the Scrum Master Role using Professional Agile Coaches and Communities of Practice.* IEEE Xplore.

[2] Hidayati, A., Budiardjo, E. K., & Purwandari, B. (2020). *Hard and Soft Skills for Scrum Global Software Development Teams.* Proceedings of the 3rd International Conference on Software Engineering and Information Management.

[3] Hayat, F., Rehman, A., Arif, K., Wahab, K., & Abbas, M. (n.d.). *The Influence of Agile Methodology (Scrum) on Software Project Management*. IEEE Computer Society.

[4] Tokdemir, G., & Uguz, S. (2022). *Research Trends in Agile Software Development*. 2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA).

[5] Alexander, F., Abdiwijaya, E. A., Pherry, F., Gunawan, A. A. S., & Anderies. (2022). *Systematic Literature Review on Solving Competitive Programming Problem with Artificial Intelligence (AI)*. 2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT).

[6] Nghiem, K., Nguyen, A. M., & Bui, N. D. Q. (2024). *Envisioning the Next-Generation AI Coding Assistants: Insights & Proposals.*

[7] Zhang, D., Mohd, & Yamaguchi, S. (2022). *Human-AI Pair Programming by Data Stream and Its Application Example*. 2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia).

[8] Johari, D., Samaddar, S., & Kumar, S. (n.d.). *AI based WebApp development using Application Programming Interface (API).* IEEE Xplore; 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering.