Copilot's Island of Joy

Balancing Individual Satisfaction with Team Interaction in Agile Development

Viggo Tellefsen Wivestad[1(✉)], Astri Barbala[1], and Viktoria Stray[1,2]

[1] SINTEF Digital, 7034 Trondheim, Norway
viggo.wivestad@sintef.no
[2] University of Oslo, 0373 Oslo, Norway

Abstract. This study assesses the integration of GitHub Copilot into agile software development practices in one of Norway's largest public sector organizations. Through a quasi-experimental survey of 115 participants, we differentiate the attitudes of users and non-users of GitHub Copilot regarding their development routines. Findings reveal that Copilot users experience significantly greater focus on engaging tasks and less dependence on colleagues compared to non-users, while non-users maintain a more cautious stance on AI use in the public sector. Further, while users generally showed more positive attitudes and fewer frustrations, these differences were not statistically significant. The study advocates for a mindful adoption of AI tools in agile settings, balancing individual benefits with interdependence and team unity.

Keywords: Agile Software Development · GitHub Copilot · AI Tool Adoption · Collaboration · Team dependence

1 Introduction and Background

AI Coding Assistants, a subset of Generative AI (GenAI), have recently emerged as popular tools, reshaping how coding tasks are approached and executed [15]. Developers, managers, and Scrum Masters are all trying to evaluate the impact of such tools on the agile software development process [15]. However, the adoption of these tools introduces significant uncertainty in agile software development, necessitating a fundamental shift in managerial planning and execution [4].

The role of the Scrum Master in agile projects has evolved significantly over the last decade [1,11], with recent studies offering new insights into this transformation. These studies explore various aspects of the Scrum Master role, such as servant leadership, facilitation, and mentorship, all crucial for enhancing team collaboration and overcoming organizational challenges [8]. Moreover, the emergence of GenAI technologies could further influence the Scrum Master's role and agile ways of working. Today, while some agile team members readily embrace these tools, others may exhibit hesitation, reflecting diverse attitudes toward technological adoption in agile environments.

L. Marchesi et al. (Eds.): XP 2024 Workshops, LNBIP 524, pp. 123–129, 2025.
https://doi.org/10.1007/978-3-031-72781-8_13

While new research is continually appearing, the single largest study to our knowledge is still that of GitHub Research [6,17], which includes a survey with over 2000 respondents. The results from this study suggest several benefits of utilizing AI Coding Assistants, such as increased productivity, higher job satisfaction, and being able to spend time on more enjoyable tasks. Their study utilized the SPACE framework, which offers a comprehensive approach to understanding developer productivity beyond conventional metrics [5]. The framework encapsulates five critical dimensions: Satisfaction and well-being (S), Performance (P), Activity (A), Communication and Collaboration (C), and Efficiency and flow (E). In this study, we extend the original survey to answer the following research question: ***What benefits of using GitHub Copilot can be identified when comparing users to a control group?***.

To answer this RQ, we articulated a series of hypotheses based on an assumption that the adoption of AI Coding Assistants will yield predominantly favorable outcomes for its users, while simultaneously reducing challenges and friction in their daily routines.

To investigate whether the use or non-use of AI Coding Assistants made a difference in the everyday work of agile software developers, we conducted a quantitative study at the National Welfare Administration (NAV), a large public sector organization based in Norway. NAV has about 1000 employees in their IT department, and the teams follow agile principles of software development. The organization is known for being at the forefront in implementing new tools and practices for software development such as continuous software engineering [2], data-driven methods [3], and a widespread use of Slack as a coordination tool [14]. Recently, NAV has also looked into incorporating the use of GenAI tools in their software development work, and in September 2023 the organization gave 100 developers access to GitHub Copilot.

2 Method and Study Design

The research design can be described as an exploratory, cross-sectional, natural quasi-experiment involving two distinct groups: users and non-users of Copilot in NAV. While the non-users in the control group had not adopted Copilot, the experiment group consisted of 100 Copilot users who had volunteered and been pre-approved before our arrival, as well as external consultants who had obtained a license through other means. Our 114 respondents consisted of 67 users and 47 non-users. Among the respondents, 81% of respondents were in-house employees; 15% were women, 79% were men, 1% identified as "other" and 5% would not disclose gender.

The survey instrument was built upon GitHub Research's survey [16], which utilized the SPACE framework [5] to assess the perceived productivity of subjects participating in an unpaid technical preview of Copilot. Our study adapted a subset of relevant survey items and translated them. Given that the original survey was designed exclusively for users of Copilot, the survey items were transformed to fit a comparative analysis between users and non-users. Items that explicitly

reference the Copilot were therefore rephrased and generalized, focusing on outcomes. Further, a few new items were added, based on ideas and concerns that were identified during a preceding interview process, and hypotheses we wanted to test. The hypotheses are listed in Table 1, and the complete survey instrument is available online at: https://doi.org/10.5281/zenodo.10987170.

The final questionnaire consisted of 57 questions (28 for both groups and an additional 29 for the users), most of them being 5-point Likert Agreement scale-type questions. The survey had a required, initial segmentation question that asked whether the respondent had used Copilot or not, which was used to label the participant as either a user or non-user.

The first part of the analysis consisted of a systematic comparison of the two groups, comparing the proportion of respondents who agreed with a statement (strongly agree + agree) against the total number of respondents. This approach, known as a Top 2-Box (T2B) analysis, is useful when focusing on an area of interest [12], but does not consider the entire distribution of replies (those who disagree or are neutral). The subsequent analysis entailed hypothesis testing. Considering the ordinal nature of Likert-scale responses, we employed the non-parametric Mann-Whitney U Test [9] to assess differences between the groups' distributions, complemented by the Rank-Biserial Correlation (RBC) [7] and the Common Language Effect Size (CLES) [10] for evaluating effect magnitude. We used Bonferroni corrected $p < 0.0056$ (0.05/9) as the threshold of statistical significance, to account for the multiple hypothesis testing problem. The analysis was predominantly done using Python.

3 Results

The final survey collected a total of 115 responses, comprising 67 users and 48 non-users of GitHub Copilot. The results from the Top 2-Box (T2B) analysis are depicted as a barplot in Fig. 1, and the results from the hypothesis testing are detailed in Table 1.

A key finding related to user benefits was the enhanced ability to focus on satisfying work; over 70% of users reported this benefit compared to only half of the non-users. Our related hypothesis, that users would experience a higher ability to focus on satisfying work, was statistically validated using the Mann-Whitney U test, and confirmed to be significant, with a meaningful effect size evidenced by both the Rank-Biserial Correlation (RBC) and the Common Language Effect Size (CLES). Note that had we not corrected for multiple hypothesis testing, our hypothesis that users experience higher job satisfaction would also have been confirmed ($p = 0.031$). This finding indicates that using AI Coding Assistants enables users to move faster past frustrating and demotivating tasks, spending more time on things they enjoy.

As shown in Fig. 1, users generally had higher agreement rates regarding job satisfaction, flow, and productivity, and disagreed more that they felt frustrated while coding. However, none of these were found to be significant, as shown in Table 1.

Table 1. Results from the Mann-Whitney U test, comparing attitudes differences between users and non-users. For each item, information is provided related to the one-sided test ($\mathbf{H_a}$), sample size of the user/non-user ($\mathbf{N_U}$, $\mathbf{N_{NU}}$), Mann-Whitney U value (**U**), P-value (**P**), Rank-Biserial Correlation (**RBC**), Common Language Effect Size (**CLES**), and whether or not to reject the null hypothesis given the Bonferroni corrected threshold.

Item	H_a	(N_U, N_{NU})	U	P	RBC	CLES	Reject H_0
Copilot is valuable	$\mu_U > \mu_{NU}$	(61, 35)	592	0.000	0.446	0.723	✓
Can focus on satisfying work	$\mu_U > \mu_{NU}$	(60, 36)	740	0.002	0.315	0.657	✓
Satisfied with job	$\mu_U > \mu_{NU}$	(61, 37)	894	0.031	0.207	0.604	
Is in flow when coding	$\mu_U > \mu_{NU}$	(63, 40)	1175	0.273	0.067	0.534	
Productive when coding	$\mu_U > \mu_{NU}$	(63, 40)	1256	0.490	0.003	0.502	
Critical of AI in public sector	$\mu_U < \mu_{NU}$	(59, 38)	660	0.000	-0.411	0.705	✓
Dependent on colleagues	$\mu_U < \mu_{NU}$	(60, 39)	800	0.003	-0.317	0.658	✓
Frustrated when coding	$\mu_U < \mu_{NU}$	(63, 40)	1054	0.075	-0.163	0.582	
Searches for information online	$\mu_U < \mu_{NU}$	(61, 39)	1142	0.360	-0.040	0.520	

Another finding was a significantly lower dependency on colleagues among the users of Copilot. This confirmed our hypothesis, with nearly half of the users disagreeing with the notion that they rely on assistance or guidance from colleagues, compared to just a fifth of the non-users.

We also confirmed our hypothesis that users are noticeably less concerned about using AI in the public sector, with "strongly disagree" being the most frequent answer (41%), with 64% disagreeing. For non-users, the answers were spread out on both sides, indicating a more nuanced and skeptical view. One non-user stated in the open-text field in the survey: *"I am uncomfortable with the idea that data collected by AI is used for purposes that I am unaware of, especially from a privacy perspective."*

The most prominent finding, aligning with our hypothesis and expectations, indicated that users perceived Copilot as significantly more valuable than non-users. Determining the cause is however difficult, as the groups were not randomly selected. Positive responses might reflect genuine benefits derived from using the tool, or simply capture the user's initial motivation. It is also worth noting that half of the non-users also agreed with the statement, suggesting a generally positive attitude towards the tool across both groups.

4 Discussion and Implications for Practice

In this paper, we outline the findings from a survey conducted among users and non-users of Github Copilot in a large, public-sector organization. When comparing the attitudes between the two groups, we identified only one significant user benefit: the enhanced ability to focus on satisfying work. Regarding other

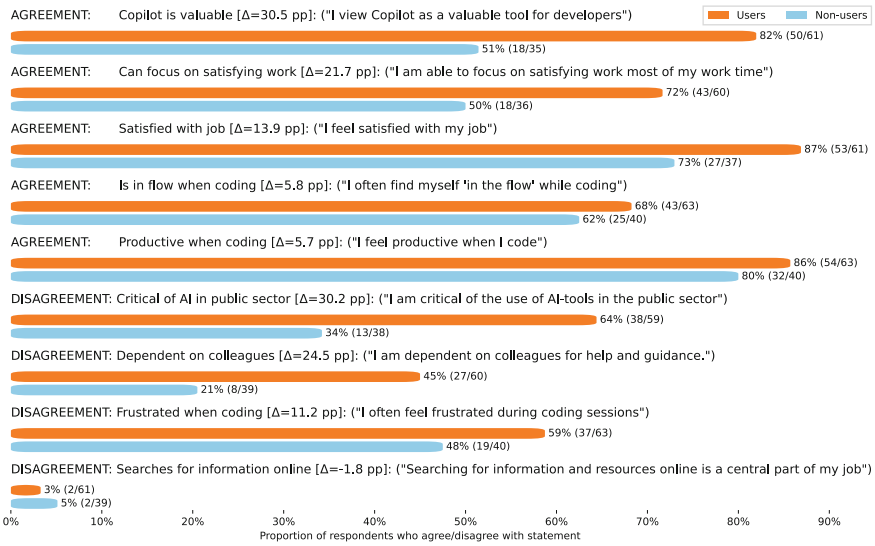

Fig. 1. Top 2-Box plot comparing the proportion of users and non-users who agreed/disagreed with statements related to attitudes and experiences among software developers.

proclaimed benefits, our study found no significant differences. This is surprising, given the proclaimed benefits by users in GitHub Research's study [6,17], such as productivity and flow. This divergence could be a user expectancy effect or an artifact of our general, and slightly vaguer, survey statement formulations.

Users also reported a notably higher job satisfaction than the non-users, with 87% of the users agreeing that they felt satisfied in their job, as compared to 73% of the non-users. While this could be a result of self-selection bias, it echoes the study of Ulfsnes et al. [15] who found that usage of GenAI tools empowered software developers by reducing tedious and repetitive tasks and freeing up time for more satisfying work. The same study also pointed out some negative consequences, with developers preferring to get help from their AI Coding Assistant rather than their colleagues. This finding resonates with our findings, with evident signs of decreased dependence, with 45% of the users outright disagreeing that they are dependent on colleagues for help and guidance.

Given these findings, we speculate that while AI Coding Assistants might enhance individual satisfaction and independence, they could pose long-term threats to teamwork, knowledge sharing, unity, and overall team interactions. Remote workers might be especially vulnerable, as the threshold to approach, and potentially disturb, a colleague is higher [13]. Such challenges would pose extraordinary disruption for agile teams and Scrum Masters, whose main tasks are to facilitate collaboration and remove team obstacles [11].

Scrum Masters can leverage these insights to balance the use of AI Coding Assistants within agile frameworks, thereby upholding the core agile value of

prioritizing individuals and interactions over processes and tools. Although AI Coding Assistants seem to enhance focus and satisfaction among developers, it is imperative to ensure that these benefits do not lead to isolation, where each team member finds themselves on a solitary 'island of joy.'

References

1. Alliata, Z., Alliata, D., Berzin, L.: Get IT! How to Start a Career in the New Information Technology: How to Start as an Agile Scrum Master. Better Karma LLC (2016)
2. Barbala, A., Sporsem, T., Stol, K.J.: A case study of continuous adoption in the norwegian public sector (2024)
3. Barbala, A., Sporsem, T., Stray, V.: Data-driven development in public sector: how agile product teams maneuver data privacy regulations. In: International Conference on Agile Software Development, pp. 165–180. Springer, Cham (2023)
4. Cleveland, S., Moschoglou, G., Millisor, E.J., Hansen, D.D.: Exploring project uncertainty and leadership strategies: domains, factors, categories, and competencies. Int. J. Smart Educ. Urban Soc. (IJSEUS) **13**(1), 1–13 (2022)
5. Forsgren, N., Storey, M.A., Maddila, C., Zimmermann, T., Houck, B., Butler, J.: The SPACE of developer productivity. Queue **19**(1), 20–48 (2021). https://doi.org/10.1145/3454122.3454124
6. Kalliamvakou, E.: Research: quantifying GitHub Copilot's impact on developer productivity and happiness (2022). https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/
7. Kerby, D.S.: The simple difference formula: an approach to teaching nonparametric correlation. Compr. Psychol. **3**, 11.IT.3.1 (2014)
8. Kristensen, S.H., Paasivaara, M.: What added value does a scrum master bring to the organisation? - a case study at nordea. In: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 270–278 (2021). https://doi.org/10.1109/SEAA53835.2021.00041
9. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. **18**(1), 50–60 (1947). https://doi.org/10.1214/aoms/1177730491. http://projecteuclid.org/euclid.aoms/1177730491
10. McGraw, K.O., Wong, S.P.: A common language effect size statistic. Psychol. Bull. **111**(2), 361–365 (1992). https://doi.org/10.1037/0033-2909.111.2.361. https://doi.apa.org/doi/10.1037/0033-2909.111.2.361
11. Shastri, Y., Hoda, R., Amor, R.: Spearheading agile: the role of the scrum master in agile projects. Empir. Softw. Eng. **26**, 1–31 (2021)
12. Shull, F., Singer, J., Sjøberg, D.I.K. (eds.): Guide to Advanced Empirical Software Engineering. Springer, London (2008). https://doi.org/10.1007/978-1-84800-044-5
13. Smite, D., Mikalsen, M., Moe, N.B., Stray, V., Klotins, E.: From collaboration to solitude and back: remote pair programming during COVID-19. In: Gregory, P., Lassenius, C., Wang, X., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 419, pp. 3–18. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78098-2_1
14. Stray, V., Barbala, A.: Slack use in large-scale agile organizations: ESN tools as catalysts for alignment? In: International Conference on Agile Software Development, pp. 20–35. Springer, Cham (2024)

15. Ulfsnes, R., Moe, N.B., Stray, V., Skarpen, M.: Transforming software development with generative AI: empirical insights on collaboration and workflow. In: Nguyen-Duc, A., Abrahamsson, P., Khomh, F. (eds.) Generative AI for Effective Software Development, pp. 219–234. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-55642-5_10
16. Ziegler, A., et al.: Productivity assessment of neural code completion. In: Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming, MAPS 2022, pp. 21–29. Association for Computing Machinery, New York (2022)
17. Ziegler, A., et al.: Measuring GitHub copilot's impact on productivity. Commun. ACM **67**(3), 54–63 (2024). https://doi.org/10.1145/3633453. https://dl.acm.org/doi/10.1145/3633453

Can ChatGPT Suggest Patterns? An Exploratory Study About Answers Given by AI-Assisted Tools to Design Problems

João José Maranhão Junior[1(✉)], Filipe F. Correia[2], and Eduardo Martins Guerra[3]

[1] Institute for Technological Research, São Paulo, Brazil
joao.junior@ensino.ipt.br
[2] Faculty of Engineering, University of Porto, Porto, Portugal
filipe.correia@fe.up.pt
[3] Free University of Bozen-Bolzano, Bolzano, Italy
eduardo.guerra@unibz.it

Abstract. General-purpose AI-assisted tools, such as ChatGPT, have recently gained much attention from the media and the general public. That raised questions about in which tasks we can apply such a tool. A good code design is essential for agile software development to keep it ready for change. In this context, identifying which design pattern can be appropriate for a given scenario can be considered an advanced skill that requires a high degree of abstraction and a good knowledge of object orientation. This paper aims to perform an exploratory study investigating the effectiveness of an AI-assisted tool in assisting developers in choosing a design pattern to solve design scenarios. To reach this goal, we gathered 56 existing questions used by teachers and public tenders that provide a concrete context and ask which design pattern would be suitable. We submitted these questions to ChatGPT and analyzed the answers. We found that 93% of the questions were answered correctly with a good level of detail, demonstrating the potential of such a tool as a valuable resource to help developers to apply design patterns and make design decisions.

Keywords: ChatGPT · Design Patterns · Artificial Intelligence · AI-assisted tools

1 Introduction

The use of design patterns [1–3] in software development is a practice that brings several advantages, including code reuse, better code organization, ease of maintenance, better scalability, and software quality [4,5]. In the context of agile methods, these characteristics are essential to keep the code ready to change, enabling adjustments in requirements in small iterations. However, becoming proficient in understanding and applying these patterns can take time and

L. Marchesi et al. (Eds.): XP 2024 Workshops, LNBIP 524, pp. 130–138, 2025.
https://doi.org/10.1007/978-3-031-72781-8_14