More Than Just Efficiency: Impact of Generative AI on Developer Productivity

Conference Paper · August 2024

CITATIONS
0

READS
421

5 authors, including:

Ernestine Dickhaut
Universität Kassel
40 PUBLICATIONS **198** CITATIONS
SEE PROFILE

Olivia Bruhin
University of St.Gallen
3 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Hendrik Wache
Technische Universität Chemnitz
21 PUBLICATIONS **106** CITATIONS
SEE PROFILE

August 2024

More Than Just Efficiency: Impact of Generative AI on Developer Productivity

Mahei Manhai Li
University of Kassel, mahei.li@uni-kassel.de

Ernestine Dickhaut
University of Kassel, ernestine.dickhaut@uni-kassel.de

Olivia Bruhin
University of St.Gallen, olivia.bruhin@unisg.ch

Hendrik Wache
Chemnitz University of Technology, hendrik.wache@wirtschaft.tu-chemnitz.de

Pauline Weritz
Industrial Engineering and Business Information Systems, p.weritz@utwente.nl

More Than Just Efficiency: Impact of Generative AI on Developer Productivity

Emergent Research Forum (ERF) Paper

Mahei Manhai Li
University of Kassel
mahei.li@uni-kassel.de

Ernestine Dickhaut
University of Kassel
ernestine.dickhaut@uni-kassel.de

Olivia Bruhin
University of St.Gallen
olivia.bruhin@unisg.ch

Hendrik Wache
TU Chemnitz
hendrik.wache@wirtschaft.tu-chemnitz.de

Pauline Weritz
University of Twente
p.weritz@utwente.nl

Abstract

The collaboration between genAI and humans in the field of information systems holds transformative potential echoing the co-creation ethos in digital ecosystems. GenAI's automated code generation capabilities present an opportunity for seamless cooperation with human developers. As genAI evolves, it can contribute to the generation of code with minimal human input, enabling developers to focus on higher-level conceptualization and problem-solving. The individual work changes by GenAI also have wider-reaching effects, which requires a holistic understanding of its impact. Our interview study with 15 software developers presents a shift towards a more balanced viewpoint on measuring the effects of genAI in software development environments, specifically the importance of human-centric indicators (e.g. satisfaction and wellbeing) in addition to traditional efficiency and effectiveness indicators. This insight underscores the balancing act between enhancing productivity and potentially undermining it, reflecting the interplay of co-creation and co-destruction in service ecosystems and calling for a more holistic socio-technical perspective.

Keywords

GenAI, system development, SPACE, value co-creation, value co-destruction.

Introduction

Digital transformation is imperative to remain competitive, leading to companies increasing focus on developing new digital products and services (Bexiga *et al.*, 2020). However, system development is hard to manage and is riddled with cost overruns, conflicting project requirements, overly long development times, and a lack of business-IT alignment (Carroll *et al.*, 2021; Klotz *et al.*, 2019). The Standish Group (2020) reports that only 31% of analyzed IT projects are completed successfully, while 50% cause higher costs, require a long development time, and 19% fail in the early stages. Thus, new development techniques and methods are required, with generative AI-supported (genAI) development tools promising to improve productivity and quality in system development to counteract the high IT project failure rates (Huber *et al.*, 2020). GenAI has been an umbrella term often representing transformer-based language models (Noy and Zhang, 2023). Such large language models (LLMs) are artificial intelligence that process and generate natural language in the form of text (Rahwan *et al.*, 2019). Recent controlled studies of such systems suggest efficiency increases between 20 – 70% while simultaneously indicating an increase in output quality (Noy and Zhang, 2023). Company's measure the effects of genAI with traditional business

model KPIs, such as the effects on cost optimization (17%) and revenue growth (26%) (Gartner, 2023). Yet, measurements on how genAI supporting services affect work systems holistically are still missing (Alter 2022). Thus, to shed light on the genAI-enabled work system facets (Alter 2021), we study system developers who use genAI applications for their development tasks.

While this value co-creation (Vargo & Lusch, 2017) between developers and genAI tools aims to increase efficiency and improve code quality, it also brings challenges that could lead to unintended consequences and highlights the balance between benefits and harms (e.g. value co-destruction, Lumivalo et al., 2024) in digital transformation efforts. Our research explores the extent to which the integration of genAI into coding practices represents a form of value co-creation that is consistent with themes of interaction between different actors in service ecosystems (Lusch & Nambisan, 2015). Thus, this study aims to analyze genAI use cases in system development projects, by investigating its impact on productivity and code quality. For this purpose, we draw from the well-established software development SPACE framework as methodological basis to measure productivity and code quality holistically – with the following research question (**RQ**): *How do you measure productivity of genAI-assistance tools for software development?*

The study is structured as follows to address the proposed research questions: First, we will present the general state of research in the domain of genAI in system development. Next, we will introduce the SPACE framework as the methodological basis for this study. In the subsequent chapter, we will highlight the initial findings from interviews with practitioners in the field of system development. Finally, we will highlight the next steps and implications that can be expected from the proposed study.

Theoretical Background

Initially, the incorporation of AI in software engineering may appear paradoxical, as artificial intelligence typically involves automating tasks through the use of "intelligent machines", whereas system development is a predominantly creative and knowledge-intensive endeavor, often reliant on human experts (Barenkamp *et al.*, 2020). The so-called “productivity paradox” leads to uncertainty on the part of companies, making them ambiguous about the use of genAI. However, AI has the capability to mechanize repetitive tasks in software development and testing, scrutinize vast data sets to identify patterns, and methodically assess data within neural networks. This can lead to accelerated development processes, cost savings, and increased efficiency, all while augmenting the creative capabilities of human developers (Barenkamp *et al.*, 2020). GenAI encompasses computational methods that can create apparently novel and meaningful content, such as text, images, or audio, based on the information from which it was trained. The broad adoption of this technology, as demonstrated by advancements like Dall-E 2, GPT-4, and CoPilot, is presently transforming how we collaborate and communicate (Feuerriegel *et al.*, 2023; Reinhard et al. 2024). Thus, system development is confronted with a new way of work through genAI support.

Research Methodology

To study the application of genAI support in system development, we adopted a qualitative inductive research design using the SPACE framework following the principles of Grounded Theory (Glaser and Strauss, 1967). The interviews with developers (N=15) that we conducted in the summer of 2023 reveal diverse potentials and limitations of genAI for system development. First, we searched for and selected suitable interview partners based on the recommendations of (Mayring, 2014). After conducting and transcribing the interviews, we analyzed our interviews iteratively (Charmaz 2006) and derived the advantages of genAI for system development. We use the SPACE framework to ensure the holistic work system perspective (Alter 2012), including productivity and code quality of genAI tools in system development. The SPACE framework introduced by Forsgren *et al.* (2021) is a structured model designed to comprehensively evaluate developer productivity in the software development industry via multidimensional KPIs. This framework provided us with the basis for our semi-structured interviews to analyze subjectively perceived productivity. It consists of five key aspects, each of which providing a different facet of productivity:

The first aspect of **satisfaction and well-being** assesses the contentment and well-being of developers. It recognizes that satisfied developers are more likely to be productive. Factors such as job satisfaction,

work-life balance, and a positive work environment are considered here. Secondly, **performance** measures the effectiveness and efficiency of developers in their work. It considers their ability to meet goals, deadlines, and deliver high-quality code. Metrics related to coding speed, accuracy, and problem-solving skills are essential in this dimension. Third, **activity** focuses on the nature of tasks and work performed by developers. It looks at the mix of activities, such as coding, debugging, testing, documentation, and meetings. Understanding how developers allocate their time is crucial for assessing productivity. The fourth element of **communication** evaluates the quality and effectiveness of interactions among team members and other stakeholders. Effective collaboration, information sharing, and clear communication channels contribute significantly to productivity. This aspect considers the flow of information and feedback within the development team. Lastly, the **environment** dimension considers the workplace conditions, tools, and resources available to developers. A conducive work environment with access to appropriate technology and infrastructure can enhance productivity, while obstacles and limitations can hinder it.

The SPACE framework acknowledges that developer productivity is not solely a matter of writing code efficiently. It recognizes the multifaceted nature of the development process and the various factors, including job satisfaction, performance, work activities, communication, and the work environment, that collectively contribute to the productivity of software developers. By considering and improving each of these aspects, organizations can work towards enhancing overall developer productivity and, consequently, the success of their software development projects.

Preliminary Findings

Our interviewees are continuously seeking innovative tools and techniques to streamline their workflows, enhance productivity, and deliver high-quality software efficiently. One such tool in practice is CoPilot, an AI-powered genAI code assistant by Microsoft that is changing the way system development is approached. By integrating AI-powered assistance tools like code autocompletion, bug detection, and documentation suggestions, its usage is transforming the way systems are created, offering several advantages for this new way of system development. The majority of respondents agreed on the potential of automated code generation through genAI. It is increasingly likely that genAI will play a significant role in generating code with minimal human intervention in the future. This evolution not only improves existing code and rapidly generates a significant amount of code but also provides case-specific content, thereby reducing the error rate associated with human errors. Our initial results focus on efficiency gains and the interaction between developers and output:

Enhanced Productivity: GenAI tools significantly accelerate the development process (Dell'Acqua *et al.*, 2023). By offering real-time code suggestions and intelligent autocompletion, developers can write code faster and with fewer errors (Sadowski and Zimmermann, 2019). This results in reduced development cycles and quicker time-to-market for software projects. In addition, developers can offload some of the more routine and repetitive coding tasks to co-piloting tools. This frees up their time to focus on more complex and creative aspects of software development, which can lead to innovative solutions and improved software design (Bouschery *et al.*, 2023). This evolution allows both novice programmers and experienced software developers to initiate genAI's code generation with minimal input, implementing initial ideas while minimizing errors.

Team Collaboration: Co-piloting platforms promote collaboration within development teams (Ziegler *et al.*, 2022). Developers can share code more effectively, communicate ideas, and review each other's work, making it easier for teams to work together, even when members are distributed across different locations (Meyer *et al.*, 2014). Firstly, AI-supported systems can think a few steps ahead of the user and thus to point out possible opportunities and obstacles in the development of IT artifacts in a preventive way. Also, more experienced developers ensure code quality (co-created by the tool) of more junior developers. However, since code generation is becoming less work-intensive, this makes already scarce resources of experienced developers more valuable and changes social dynamics within the team.

Improved Code Quality: AI-driven co-piloting tools can identify and flag potential issues and bugs in the code during the development process. This proactive error detection and correction can improve code quality, reduce the likelihood of post-release issues, and ultimately lead to more reliable software (Ziegler *et al.*, 2022). AI-powered co-piloting tools continually learn and adapt to the specific needs and

preferences of the development team. This means that over time, they become even more effective and aligned with the team's coding style and standards. Code quality implies the need to refactor the code base, which can mean that the code is written in a way that causes inefficiency at runtime or that it is not easily maintainable. The industry experts see the potential and the problems around genAI and how to ensure code quality and currently believe that expert developers need to review code for quality. For complex applications, there is a risk that the complex dependencies of the software could negatively affect the code quality for customizations.

Next Steps and Implications

Based on a preliminary set of 15 expert interviews, we explored how genAI tools are changing systems development in terms of human-AI collaboration, identifying a shift from time-consuming tasks to improving code quality and productivity. However, as a preliminary result, we have identified three themes that are improved by the collaboration of genAI tools and developers, namely enhanced productivity, team collaboration, and code quality. Issues such as employee satisfaction, burnout due to stress, and other typical SPACE dimensions require future research. This initial exploration of the use of genAI tools in coding tasks fits into the broader discourse on co-creation within digital service ecosystems facilitating collaborative dynamics between humans and AI. It also encourages careful consideration of the boundary points at which such collaboration could inadvertently tip into co-destructive scenarios, particularly if integration is not carefully managed or leads to over-dependence that could stifle creativity or learning. We further demonstrate that SPACE can be used as a starting framework for studying the effects of genAI and its different facets (Alter 2021).

In order to deepen the results of our interview study, we plan a multi-case study according to (Yin, 1981), including 23 additional interviews with both junior and experienced developers who are both using genAI tools for software development. In the case study, we will accompany three companies in their application of genAI tools for system development. Through this planned multi-case study, we aim to study how genAI tools in system development embody elements of co-creation and hold the potential for co-destruction. Via systematically analyzing the impact of genAI on software development work, we aim to explore dimensional configurations that lead to a balanced impact across relevant dimensions. Our future research aims to identify the benefits of genAI while ideally avoiding problems that could undermine the effectiveness of software development as an ecosystem and the well-being of its participants.

In conclusion, our preliminary results provide initial facets of a genAI-enabled work system facets (Alter 2021) for software development initiatives. We aim to identify work system facet configuration to explain under which circumstances genAI are particularly effective for improving software development. For practice, we aim to develop a measurement approach for assessing genAI impact on software development, considering a multi-faceted and multi-dimensional approach.

REFERENCES

Alter, Steven. "Facets of Work: Enriching the Description, Analysis, Design, and Evaluation of Systems in Organizations." Communications of the Association for Information Systems 49 (2021): 11.

Alter, Steven. "Understanding artificial intelligence in the context of usage: Contributions and smartness of algorithmic capabilities in work systems." International Journal of Information Management 67 (2022): 102392.

Barenkamp, M., Rebstadt, J. and Thomas, O. (2020), “Applications of AI in classical software engineering”, *AI Perspectives*, Vol. 2 No. 1.

Bexiga, M., Garbatov, S. and Seco, J.C. (2020), “Closing the gap between designers and developers in a low code ecosystem”, in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, New York, NY, USA*, ACM, New York, NY, USA.

Bouschery, S.G., Blazevic, V. and Piller, F.T. (2023), “Augmenting human innovation teams with artificial intelligence: Exploring transformer-based language models”, *Journal of Product Innovation Management*, Vol. 40 No. 2, pp. 139–153.

Brown, T., Mann, B., Ryder, N., ...& Amodei, D. (2020), “Language models are few-shot learners”, *Advances in neural information processing systems*.

Carroll, N., Móráin, L.Ó., Garrett, D. and Jamnadass, A. (2021), "The Importance of Citizen Development for Digital Transformation", *Cutter IT Journal*, Vol. 34 No. 3, pp. 5–9.
Charmaz, K. 2006. Constructing Grounded Theory - A Practical Guide through Qualitative Analysis, London: Sage Publications Ltd.
Dell'Acqua, F., McFowland, E., Mollick, E.R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F. and Lakhani, K.R. (2023), *Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality*.
Feuerriegel, S., Hartmann, J., Janiesch, C. and Zschech, P. (2023), "Generative AI", *Business & Information Systems Engineering*.
Forsgren, N., Storey, M.-A., Maddila, C., Zimmermann, T., Houck, B. and Butler, J. (2021), "The SPACE of developer productivity", *Communications of the ACM*, Vol. 64 No. 6, pp. 46–53.
Gartner (2023), "Generative AI: What Is It, Tools, Models, Applications and Use Cases", *https://www.gartner.com/en/topics/generative-ai*.
Glaser, B. and Strauss, A. (1967), "Grounded theory: The discovery of grounded theory", *Sociology the journal of the British sociological association*, No. 12 (1).
Huber, T.L., Winkler, M.A.E., Dibbern, J. and Brown, C.V. (2020), "The use of prototypes to bridge knowledge boundaries in agile software development", *Information Systems Journal*, Vol. 30 No. 2, pp. 270–294.
Klotz, S., Kopper, A., Westner, M. and Strahringer, S. (2019), "Causing factors, outcomes, and governance of Shadow IT and business-managed IT: a systematic literature review", *International Journal of Information Systems and Project Management*, Vol. 7 No. 1, pp. 15–43.
Lim, W.M., Gunasekara, A., Pallant, J.L., Pallant, J.I. and Pechenkina, E. (2023), "Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators", *The International Journal of Management Education*, Vol. 21 No. 2, p. 100790.
Lumivalo, J., Tuunanen, T., and Salo, M. 2024. "Value Co-Destruction: A Conceptual Review and Future Research Agenda," Journal of Service Research (27:2), SAGE Publications Inc., pp. 159–176. (https://doi.org/10.1177/10946705231177504).
Lusch, R. F., and Nambisan, S. 2015. "Service Innovation: A Service-Dominant Logic Perspective," MIS Quarterly (39:1), pp. 155–175.
Mayring, P. (2014), *Qualitative content analysis: theoretical foundation, basic procedures and software solution*.
Meyer, A.N., Fritz, T., Murphy, G.C. and Zimmermann, T. (2014), "Software developers' perceptions of productivity", *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 19–29.
Noy, S. and Zhang, W. (2023), *Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence*.
Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J.W., Christakis, N.A., Couzin, I.D., Jackson, M.O., Jennings, N.R., Kamar, E., Kloumann, I.M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D.C., Pentland, A.'S.', Roberts, M.E., Shariff, A., Tenenbaum, J.B. and Wellman, M. (2019), "Machine behaviour", *Nature*, Vol. 568 No. 7753, pp. 477–486.
Reinhard, Philipp; Li, Mahei Manhai; Peters, Christoph; Leimeister, Jan Marco (2024), "*Let Employees Train their own Chatbots: Design of Generative AI-enabled Delegation Systems"* European Conference on Information Systems (ECIS), Paphos, Cyprus.
Sadowski, C. and Zimmermann, T. (Eds.) (2019), *Rethinking Productivity in Software Engineering*, Springer Nature, Berkeley, CA.
Standish Group (2020), "CHAOS 2020 Beyond Infinity".
Vargo, S. L., and Lusch, R. F. 2017. "Service-Dominant Logic 2025," International Journal of Research in Marketing (34:1), pp. 46–67.
Yin, R.K. (1981), "The Case Study Crisis: Some Answers", *Administrative Science Quarterly*, Vol. 26 No. 1, p. 58.
Zhang, B., Liang, P., Zhou, X., Ahmad, A. and Waseem, M. (2023), "Practices and Challenges of Using GitHub Copilot: An Empirical Study", in *Proceedings of the 35th International Conference on Software Engineering and Knowledge Engineering, July 1-10, 2023*, KSI Research Inc, pp. 124–129.
Ziegler, A., Kalliamvakou, E., Li, X.A., Rice, A., Rifkin, D., Simister, S., Sittampalam, G. and Aftandilian, E. (2022), "Productivity assessment of neural code completion", *Proceedings of the 6th ACM SIGPLAN International Symposium on*, pp. 21–29.