December 2024

Auto-Pilot, Co-Pilot and Pilot: Human and Generative AI Configurations in Software Development

Narges Safari
Macquarie University, narges.safari@mq.edu.au

Angsana A. Techatassanasoontorn
Auckland University of Technology, angsana@aut.ac.nz

Antonio Diaz Andrade
University of Agder, antonio.diaz@uia.no

Auto-Pilot, Co-Pilot and Pilot: Human and Generative AI Configurations in Software Development

Short Paper

Narges Safari
Macquarie University
Sydney, Australia
narges.safari@mq.edu.au

Angsana A. Techatassanasoontorn
Auckland University of Technology
Auckland, New Zealand
angsana@aut.ac.nz

Antonio Díaz Andrade
University of Agder
Kristiansand, Norway
antonio.diaz@uia.no

Abstract

Generative AI (GenAI), with its versatility and general-purpose capability for content creation, has the potential to shape a wide range of professional work across different industries. This research-in-progress identifies emergent configurations of humans and GenAI to offer an understanding of the changing nature of software development tasks and work practice in general. The empirical materials are drawn from 15 interviews with software developers with different levels of experience who use GenAI as part of their work. Our abductive analysis, grounded in the notion of configurations and a work design perspective, identifies three configurations – i.e., GenAI as auto-pilot, GenAI as co-pilot, and human as pilot-in-command – that explain the manifestations of automation and augmentation and the distribution of responsibilities between humans and GenAI. Overall, the findings shed light on the changing nature of software development work where human agencies are crucial in managing emergent configurations with AI.

Keywords: Generative Artificial Intelligence, augmentation, automation, configuration, software development, work design

Introduction

Generative AI (GenAI) is an umbrella term referring to a class of computational techniques (e.g., deep neural networks), interfaces (e.g., conversational agents) and applications (e.g., content generation) (Feuerriegel et al., 2024). GenAI, such as ChatGPT, has received much attention due to its capability to identify patterns within existing data and create seemingly new content, including text, video, audio, and computer codes (Brynjolfsson et al., 2023). With its versatility and general-purpose capability for content creation, GenAI has far greater potential to shape a wide range of professional work across different industries compared to traditional AI designed to support domain-specific decision-making, such as medical diagnosis (Lebovitz et al., 2022) and policing (Waardenburg et al., 2022). GenAI's potential benefits and risks have been reported across different industries. On the one hand, studies show that GenAI substantially improves the productivity of customer support workers (Brynjolfsson et al., 2023) and professionals in their writing tasks across many fields (Noy & Zhang, 2023). On the other hand, GenAI can potentially replace all kinds of knowledge workers (Sætra, 2023). These views echo a persistent debate about augmentation, where GenAI collaborates with humans, vs. automation, where GenAI replaces humans (Raisch & Krakowski, 2021). However, such debate has often been discussed and examined in the context of traditional AI with analytical capabilities in specialized domains (Anthony et al., 2023; Lebovitz et al., 2022; Waardenburg et al., 2022). The role of GenAI deserves more scrutiny, especially on how its interaction with work is likely to be more complex, contextualized, and intertwined than the dystopian view on worker replacement (Sætra, 2023) or the utopian view on symbiotic human-AI super teams (Anthony et al., 2023; Wilson & Daugherty, 2019).

Our concern in this paper is with the everyday experience of software developers who use GenAI in their work practice. It was predicted that by 2027, 70% of developers will use GenAI coding tools, up from 10% today (Gartner, 2023). Although the scholarly literature shows an increasing interest in the use of GenAI to improve software developer productivity (Peng et al., 2023), most studies in the software development literature focus on GenAI's technical capabilities (e.g., Nguyen & Nadi, 2022; Treude, 2023) without paying enough attention to the effects of GenAI on software development practice.

Our research is underpinned by the following question: *How do software developers configure their relationships with GenAI in their practice?* To address our research question, we interviewed 15 software developers with varying experience levels. In our abductive analysis, we paid particular attention to the distribution of responsibilities between humans and GenAI and the roles of humans and GenAI at the task level.

GenAI and the Changing Nature of Work

With the capability to identify patterns within existing data and create new content, GenAI is not a technology or set of technologies but a continually evolving frontier of emerging computing capabilities (Berente et al., 2021). The emerging and complex nature of human and GenAI collaborations raises an important question of whether GenAI should be viewed as an instrumental workplace technology (Raisch & Krakowski, 2021) that can autonomously perform a variety of routine and non-routine tasks or a complementary actor that can partner with humans to develop collaborative intelligence (Wilson & Daugherty, 2019).

Previous research shows that GenAI shapes routine work and increases the productivity of humans by assisting them with suggested responses, with the most significant impact on novice workers (Brynjolfsson et al., 2023). In the context of software development, the adoption of GenAI could significantly transform the nature of the work software developers perform (Pothukuchi et al., 2023). In contrast to work domains like customer support, which involves mostly routine work, developing software requires creative activities to develop a solution and apply it to a specific context (Cooper, 2000). Software development often involves labor-intensive tasks requiring developers to integrate knowledge about an application problem domain and technical expertise to develop a software solution (Tiwana, 2010). The five phases of the software development process are requirements analysis and specification, design and development, testing, deployment, and maintenance and support (IBM, 2023).

A general-purpose GenAI (e.g., ChatGPT) and code generation systems (e.g., GitHub Copilot, Amazon CodeWhisperer) are currently the two most popular GenAI systems often used to support software

development tasks. Copilot, powered by OpenAI's GenAI model, is a tool that can generate code in different programming languages when provided with prompts (Peng et al., 2023). Research has evaluated GenAI technical capabilities for different types of problems and programming languages. For example, Nguyen and Nadi (2022) report that Copilot's performance for Java is better than JavaScript. Treude (2023) shows that ChatGPT allows for the generation of multiple source code solutions for the same query, increasing the efficiency of software development. Despite these technical and performance insights of GenAI, very little is known about the emergence of new forms of human-GenAI work (Baptista et al., 2020). In this study, we explore how humans endeavor to assemble arrangements of work tasks between themselves and GenAI and the forms of these emerging configurations of work.

AI Roles: Automation and Augmentation

The literature on AI and the changing nature of work suggests that AI systems have mastered two broad capabilities: AI-enabled automation and AI-enabled augmentation. AI-enabled automation refers to the capability of AI to automate structured and semi-structured tasks such as information acquisition and analysis (Grønsund & Aanestad, 2020). AI-enabled augmentation refers to the capability of AI to augment humans to perform a task (Raisch & Krakowski, 2021), approach a problem situation, gain comprehension about it, and derive possible solutions (Grønsund & Aanestad, 2020).

Scholarly investigations on automation and augmentation can be broadly classified into two streams. The first stream of studies emphasizes the importance of automation over augmentation. For example, Beese and Fahse (2023) show that sales forecasts produced by AI alone are more accurate than those produced by AI and adjusted by humans, suggesting automation outperforms augmentation. The second stream of studies emphasizes the importance of augmentation over automation. For example, a study found that AI is primarily adopted to augment human activities (55%) rather than to automate business activities (11%) in research and development teams (Johnson et al., 2022).

Although AI-enabled automation and augmentation have been increasingly discussed, much of the attention has been either on automation and managing AI in organizations (e.g., Berente et al., 2021), automation of selected business processes (e.g., Grønsund & Aanestad, 2020), or human-AI collaboration (Anthony et al., 2023). In addition, previous studies on AI often focus on traditional AI with industry-specific analytical capabilities (e.g., Lebovitz et al., 2022; Waardenburg et al., 2022). Considering that GenAI is likely to result in novel forms of technology-enabled work (Feuerriegel et al., 2024), it is important to theorize the constituent nature of automation and augmentation across various work tasks in a particular job role to better understand how GenAI shapes tasks and work design (Parker & Grote, 2022).

Our work builds on the premise that an encounter between humans and GenAI invites a multiplicity of dynamically enacted and mutually constituted actions and interactions between them. We use the interrelated concepts of "configuring" and "configuration" to ground our analysis (Suchman, 2007). Configuring work involves the effort and work that humans put into analyzing the specific capabilities of AI in their work setting and configuring their work practices (Baptista et al., 2020). The emergent forms of human-AI configurations reflect the relational association between human and GenAI while paying attention to their distinct roles and actions (Suchman, 2007). In addition, our theorizing is grounded in a work design perspective (Parker & Grote, 2022) as an explanatory vehicle of the shape of configured work. We pay special attention to the distribution of responsibilities, roles of humans and GenAI and manifestations of automation and augmentation to offer a nuanced understanding of how humans use GenAI to configure their work at the task level. We then aggregate the task-level, human-AI configurations to theorize the changing nature of software development and the evolving role of software developers.

Methodology

We interviewed 15 purposefully sampled (Patton, 2002) software developers based in New Zealand who use GenAI (ChatGPT, Copilot, Cursor, etc.) as part of their work between June 2023 and June 2024 (see Table 1). We followed an interview guideline designed to prompt participants to share their experiences, perceptions, concerns, and organizational policies promoting or restricting the use of GenAI in software development tasks. Our participants freely elaborated on the points they considered particularly relevant (Myers & Newman, 2007), especially regarding how they use GenAI.

Participants	Experience	Sector	Interview duration
Aaron	Senior	Energy	1 hr 5 min
Adam	Junior	Telecommunications	38 min
Alex	Senior	Energy	57 min
Anthony	Senior	E-commerce	40 min
Bruce	Intermediate	IT consultancy	35 min
Cameron	Senior	IT consultancy	27 min
Damian	Senior	Retail conglomerate	38 min
Elizabeth	Intermediate	Data analytics consultancy	54 min
Iris	Junior	Telecommunications	20 min
Jeremy	Senior	IT consultancy	42 min
Michael	Senior	Finance	45 min
Robert	Senior	Grocery retail	41 min
Shawn	Intermediate	Digital product services	37 min
Steve	Intermediate	IT consultancy	25 min
Thomas	Junior	IT consultancy	40 min

Table 1. Participant Information – Participants' names are pseudonyms

We followed an abductive thinking process (Tavory & Timmermans, 2014), by which we constantly moved between data and theory (Díaz Andrade, 2023). The literature on AI and GenAI gave us a theoretical understanding of the likely roles of GenAI in the changing nature of work in software development. As we progressed in the data gathering, we independently coded the interview data inductively and held regular discussions to compare our codes and interpretations, which were largely convergent.

The open coding process revealed differential ways software developers use GenAI at the task level; these practices were mapped against the automation and augmentation capabilities discussed in the literature. To develop a deeper understanding of the changing nature of software development work, we aggregated the patterns of automation and augmentation at the task level using the notion of configurations (Suchman, 2007) and a work design perspective (Parker & Grote, 2022) to highlight the roles of human software developers and GenAI and the distribution of their responsibilities.

Findings

Our abductive analysis identified three distinctive configurations concerning the evolving roles of software developers and GenAI: 1) GenAI as auto-pilot, 2) GenAI as co-pilot, and 3) human as pilot-in-command. These configurations describe different degrees of engagement of the human-GenAI dyad in software development tasks, thus revealing the emergence of a new assemblage of relations to perform tasks with distributed responsibilities. Although some may argue that the human as pilot-in-command does not involve GenAI in this emergent configuration, it is worth noting that the presence of GenAI is visible in how software developers carefully consider its possibilities, risks, and drawbacks and intentionally decide to put GenAI aside for specific software development tasks. Next, we present each configuration along with evidence from the interview data.

GenAI as Auto-Pilot: Automation Configuration

Software developers engage with GenAI as auto-pilot by delegating tasks they often refer to as "*grunt work*", "*boring work*", and "*repetitive tasks*" to GenAI. Our participants indicate, with different degrees of emphasis, that GenAI is well suited to do tedious tasks. Iris uses GenAI to automate "*mundane little tasks*" such as "*proofreading code*" to save time for "*creating better stuff.*" Anthony refers to the use of Copilot for code automation: "*It's almost like you don't even notice some of the things that [GenAI] does, like the simple code completions. I'll just be typing [code] and I'll be like 'Oh yeah, that's right'.*"

GenAI can also automate other tedious tasks. Michael offers an example: "*Maybe you are building a form, which has got a whole bunch of just very, very boilerplate code, and taking two or three hours to write … You can have all that done in two minutes with ChatGPT.*" He adds, "*I certainly would not say I miss*

writing the repetitive code." Similarly, Aaron assertively describes how he delegates tasks to GenAI: "*If it is a lot of this repetitive work, which is boring and takes a long time, just automate it*."

Producing test cases is another task developers rank among their least favorite. Robert reflects, "*We are passionate [about] writing the code, but when it comes to the test. Hmmm, writing test cases? Ah, we do not like it.*" That is why he passes the code he writes on to GenAI and "*ask[s] the ChatGPT, 'OK, can you write the test cases?' That is really helpful! I use it … most of the time*." Similarly, Thomas remembers how he used GenAI at his previous organization: "*One thing I did use ChatGPT for, fairly often … was generating test cases … because you can give it a function and ask it to unit test it*." Elizabeth also relies on GenAI to get suggestions of "*how I could test some piece of code better*." Alex uses GenAI to produce test cases, which he refers to as an ongoing "*burden*", while Aaron reasons that "*No one likes to write a lot of tests because tests are boring*."

Besides using GenAI for testing, Bruce delegates documentation tasks to GenAI because "*documentation is probably the most annoying part of software engineering*." Elizabeth adds that documentation is "*something that I neglect to do*." Steve confirms that GenAI "*speeds up documentation*" and Aaron admits that "*developers are usually even lazier [in documentation tasks] than when it comes to writing tests*." He illustrates the benefits of using GenAI for documentation: "*When you read through code, it is like, 'Oh, the code is doing this … and the overall reason is so it can do this thing', whereas GenAI can look at that and then just put it down very succinctly and very quickly*." Similarly, Adam explains: "*If I don't have any documentation for code I wrote, or if I'm using someone else's code and they don't have any documentation, I … tend to just copy everything [to] ChatGPT and ask it to write down documentation about this code.*" Bruce envisions that "*having an [Gen]AI being able to go through and produce that documentation, to a quality that exceeds that or at least meets that of a human, is a huge benefit*." The rationale behind his remark is that "*most engineers are quite bad at that anyway*."

GenAI as Co-Pilot: Augmentation Configuration

Most software developers are cognizant of different GenAI capabilities, their underlying algorithms, and their limitations. Therefore, software developers often balance taking advantage of GenAI's benefits while exercising caution in how they work with GenAI applications. Michael elaborates on how he strikes a balance: "*Being very open-minded … and also judicious about the inputs which you give it and how you judge the outputs that it gives you. Knowing that it is not necessarily going to be correct*."

Our participants' experiences show that they do not leave everything in the hands of GenAI. Instead, they selectively collaborate with GenAI on certain tasks to support their work. They are generally suspicious of the quality of code suggested by GenAI. Robert judges it as inadequate: "*If I ask [GenAI] to write code, most of the cases it is useless code*." His approach is to take control of the "logic part" and ask GenAI to produce what he labels as a "*code snippet*" for a well-delimited function with no business logic involved. Aaron emphasizes that GenAI can only be trusted for generating "*simple enough*" code. This simple code becomes the input for software developers to work on the tasks that demand an understanding of business requirements. For example, Jeremy emphasizes that he checks the codes generated from GenAI: *"I give ideas and … imagine or frame what I want to do and I feed that to GenAI … and then it spits out mostly rudimentary code … I fact check that and then make sure that it actually works*."

Similarly, Bruce uses GenAI to speed up the coding process: "*ChatGPT is very good at giving you a skeleton of code that might not be fully functional. It might be 80 to 90 percent correct, but for simple applications, you can generally get something that you can then modify.*" Another example of the collaboration between software developers and GenAI is code optimization. Robert explains how this collaboration happens after he has written some code: "*I will give existing code and ask GitHub Copilot or ChatGPT 'Can you make it more performance enhanced code? It will write an optimized code.*"

All our participants concur on praising GenAI for being agnostic about programming languages, which makes it appealing to support various needs in software development projects. Robert declares that he often uses GenAI as a "*language translator*" while Steve uses GenAI to generate "*rough codes*" in a programming language that he is not familiar with. Iris refers to GenAI as one of her "*best friends*" who helps her "*figure out differences in [programming] languages*." Bruce considers the capacity of "*code conversion between*

languages … really handy" and illustrates his point through an example: "*If you have a lot of experience in something like Python or C and you have an application that needs to be written in Golang … you could write things in a comfortable language … and then have the AI convert them*." When referring to language conversion possibilities, Thomas emphasizes the importance of understanding what GenAI does: "*If you do not know how to reverse a link list in Go, you can just write a comment, but you know how to use it because you learned the concepts behind it*." Michael observes that GenAI can save a substantial amount of developers' time when they encounter a new programming language: "*If there is a language that you are less familiar with, [it] is difficult and daunting. Whereas if [GenAI] spits out a bunch of code … 'Oh, cool! That is actually very close to what I want. I just need to tweak this or adjust that.*'" In addition to supporting coding activities, GenAI also helps with information-related activities. Thomas values the time he saves on searching for information: "*It just reduces the clicks, and that is a really big thing!*" He highlights the benefits he derives from GenAI compared to conventional specialized sites: "*It makes it much faster to look up things you do not know how to do. Personally, I do not really even have to use Stack Overflow anymore … Really, it just speeds it up a lot*."

Software developers emphasize that their collaboration with GenAI involves an iterative process to derive better results. Thomas explains the importance of giving GenAI "*specific quantifiers*" and describes how he adds comments to the code to prompt better responses from GenAI: "*When it gets something wrong, you just give it more specific examples … Generally, it can get more and more accurate results*." Bruce follows a similar process but emphasizes the importance of human knowledge: "*If you know what you want your software to do and you can put it into pseudo codes or comments, you can have the GenAI kind of flesh that out for you*." Michael corroborates Bruce's observation and elaborates on how software developers can closely collaborate with GenAI, which he labels as "*co-generation*": "*You can give [GenAI] context and then ask it to write things for you. I think that is kind of the magic, where that human interface and input will always be very useful.*"

Human as Pilot-In-Command: Unassisted Configuration

Good software requires developers to have a sound conceptual understanding of coding logic and knowledge about business domains to be in full control of the software they develop. This way of working is akin to airline pilots, who are ultimately responsible for the operation of their aircraft (Holford, 2022).

Our participants agree on the undeniable power of GenAI in assisting them with several software development tasks. Equally, they unanimously espouse the view of the unique, irreplaceable role humans play in software development tasks. Michael affirms developers should avoid the temptation of "*asking an AI to effectively think for you, to evaluate some data and give back a response*." Jeremy explains that GenAI cannot surpass humans in idea generation and imagination: "*As a human, I have more ideas and imagination … That is my edge against generative AI. I am the source of imagination. I am the source of an idea*." Along the same lines, Damian emphasizes that "*the main business logic is coming from my head and also my colleagues' heads … only people can understand it and easily adjust their ideas*." Elizabeth stresses that GenAI "*cannot generate novel ideas*." Cameron has seen colleagues trying to code *"key business logic areas … they spend more time trying to get ChatGPT to provide proper answers, than understanding requirements. And [the answers] are not necessarily mapping to the actual business requirement*." Shawn does not rely on GenAI for developing business solutions: "*[ChatGPT] is very confident [in saying] … 'This is the system that you need to use, it'll solve all your problems'. … You do a bunch of research and like 'No, no, no. This doesn't really meet our requirements at all. They're way off.*"

In addition, our participants clearly distinguish the tasks that can be delegated or done in collaboration with GenAI from those that can only be done by developers. For example, Aaron questions the intelligence attributed to GenAI and argues that those tasks that require creativity cannot be done by GenAI: "*What insights will be useful for your customer or your employer or whatever … That is where it is more like creative because when say AI, AI is not AI. It is just some machine learning, which cannot think properly*." Robert recognizes that "*they cannot use [GenAI] for the first couple of [development] phases. [GenAI] cannot analyze the requirements*." He emphasizes that humans should be in control of the design and development process, particularly for complex systems: " *Some software is very complex and some are so big with many components. Some components are running on multiple servers. It will still require the*

human designing them." Aaron offers a nuanced perspective on the use of GenAI for creating testing scenarios: "*You can use [GenAI] for either writing code or writing tests. If you start using it for writing both, that leads to a problem where Copilot will be testing software that it has written as well … Well, how do you trust it? Basically, it makes decisions and tests that it made the right decisions.*"

Figure 1 interweaves different configurations with the phases of software development.

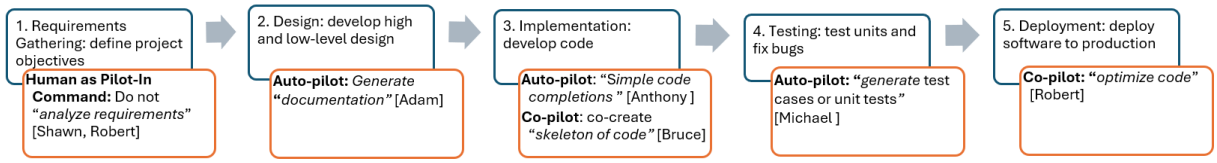

Figure 1. Human and GenAI Configurations in Software Development

Discussion and Expected Contribution

This study offers an understanding of emergent human-GenAI configurations in software development. Our findings identify three configurations that reveal the changing roles of human software developers and the emerging roles of GenAI in software development activities. For each configuration, we present elements of configuring work, including task characteristics, distribution of responsibilities between humans and GenAI, and the roles of a human software developer and GenAI. Table 2 presents a summary of these configurations, their elements of configuring work, and the ensuing benefits from GenAI.

The three configurations are the product of human effort and, in some cases, intervention to put GenAI in place in software development tasks and manage human-GenAI micro-level interactions. Therefore, it is important to recognize that human agency is at the center of the configuring of GenAI in work practices (Baptista et al., 2020). In the "GenAI as auto-pilot" configuration, a human software developer delegates narrowly scoped tasks (e.g., producing test cases) and carefully checks the quality of suggested outputs from GenAI. Thus, the symbiotic relationship between GenAI and a human software developer is like a "master-slave" relationship wherein a human directs the actions of the GenAI tool by focusing on GenAI-enabled automation capabilities (Aleksander, 2017). In the "GenAI as co-pilot" configuration, a human software developer involves GenAI in more complex technical tasks that require a careful choreography of human-AI collaboration with the human software developer configuring the working relationship, managing the interactions, and refining automated outputs from GenAI. Thus, the human-GenAI constituting relationship requires a careful balance of automation and augmentation (Raisch & Krakowski, 2021). Finally, the "human as pilot-in-command" configuration reflects how a human software developer exerts their agency and decides not to involve GenAI in tasks that GenAI will likely perform poorly due to its lack of contextual awareness or those tasks with high risks of privacy and intellectual property breaches. This is mainly because human software developers are ultimately responsible for their work. However, it is important to recognize the transient nature of these emergent configurations. As GenAI capabilities evolve, the locus of action, control, and authority may shift humans away from the center of some of these configurations (Baptista et al., 2020; Berente et al., 2021).

Configuring work	**GenAI as auto-pilot**	**GenAI as co-pilot**	**Human as pilot-in-command**
Task characteristics	Narrowly scoped tasks often perceived as mundane by software developers (e.g. generating test cases)	Technical-oriented tasks that do not involve business domain knowledge (e.g. optimizing code)	Tasks that require an understanding of the business domain (e.g. gathering client requirements)
Distribution of responsibilities	Developers delegate narrowly scoped tasks to GenAI	Developers cautiously collaborate with GenAI to co-create solutions	Developers consciously refuse to use GenAI for sensitive tasks

Role of developer	Delegator, validator	Delegator, validator, collaborator, supervisor	Discerner, thinker, creator, designer
Role of GenAI	Task automator	Task automator and collaborator	None
Benefits of GenAI	Efficiency (reduced workload, increase productivity)	Efficiency (reduced workload, increased productivity)	N/A

Table 2. Emergent Configurations of Human and GenAI in Software Development

The preliminary findings of this study contribute to the literature on AI and the changing nature of work and the discourse on the role automation and augmentation play in the shaping of work practices. Workplace technologies often reinforce and evolve work practices in organizations (Baptista et al., 2020). Scholars strongly agree that AI technologies, unlike the previous generations of workplace technologies, will fundamentally transform an organization in complex and unexpected manners (Berente et al., 2021). Yet little is known about how the use of GenAI changes the constitutive relationships among human workers and their tasks at a micro-level and their contribution to the structuring and redesigning of work (Parker & Grote, 2022). Our study offers a nuanced and contextualized understanding of the intertwined effects of GenAI-enabled automation and GenAI-enabled augmentation in shaping software development work. The context-specific and work practices perspective helps us build micro-level layers of understanding of the distinct effects of GenAI technologies (Parker & Grote, 2022). By taking a relational association between humans and GenAI in a software development context, our study extends an understanding of GenAI capabilities and their effects beyond studies that evaluate GenAI technical capabilities and performance in isolation (Nguyen & Nadi, 2022; Treude, 2023).

This study has implications for inviting organizations to embrace a human-centric principle in developing a responsible and mindful approach to managing GenAI technologies. Since employees usually explore the potential of GenAI to support their work mostly without the organization's sanction, organizations may want to use the collective learning from these employees' experiences with GenAI to develop formal policies on meaningful, responsible, and accountable GenAI use in organizations. Organizations should carefully deliberate on how work should be structured by taking an integrated consideration of how tasks, jobs, work, and GenAI should be configured. Special attention should be paid to how tasks might be best shared between humans and GenAI to achieve positive outcomes for both individuals and organizations.

Conclusion and Future Work

This research-in-progress study empirically investigates the emergent configurations of humans and GenAI to offer a better understanding of the changing nature of software development activities and work practice in general. In addition, our preliminary findings highlight the intertwined relationship between automation and augmentation in forming distinct human-AI configurations.

As we progress towards the final presentation for ICIS 2024 in December, several key areas of work remain that are pivotal for enhancing the comprehensiveness and impact of our study. First, we plan to extend our analysis by including an analysis of additional interviews that we collected from software developers to enrich our findings. This expansion will provide a more robust foundation for our findings and conclusions. Second, the next step is to enrich our findings by developing a finer-grained configuring process to explain how developers and GenAI are intertangled in each configuration.

References

Aleksander, I. (2017). Partners of humans: A realistic assessment of the role of robots in the foreseeable future. *Journal of Information Technology, 32*(1), 1–9.

Anthony, C., Bechky, B. A., & Fayard, A. (2023). Collaborating with AI: Taking a system view to explore the future of work. *Organization Science, 34*(5), 1672–1694.

Baptista, J., Stein, M.-K., Klein, S., Watson-Manheim, M. B., & Lee, J. (2020). Digital work and organisational transformation: Emergent digital/human work configurations in modern organisations. *Journal of Strategic Information Systems, 29*(2), 1–10.

Beese, J., & Fahse, T. (2023). Modern centaurs: How humans and AI systems interact in sales forecasting. *ECIS 2023 Research Papers*, (No. 267), 1–15.
Berente, N., Bin, G., Recker, J., & Santhanam, R. (2021). Managing artificial intelligence. *MIS Quarterly, 45*(3), 1433–1450.
Brynjolfsson, E., Li, D., & Raymond, L. R. (2023). Generative AI at work. *National Bureau of Economic Research Working Paper Series*, (No. 31161), 1–57.
Cooper, R. B. (2000). Information technology development creativity: A case study of attempted radical change. *MIS Quarterly, 24*(2), 245–276.
Díaz Andrade, A. (2023). Dancing between theory and data: Abductive reasoning. In R. M. Davison (Ed.), *Handbook of qualitative research methods for information systems* (pp. 274–287). Edward Elgar Publishing.
Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering, 66*(1), 111–126.
Gartner. (2023). Set up now for AI to augment software development. Retrieved March 2, 2024, from https://www.gartner.com/en/articles/set-up-now-for-ai-to-augment-software-development
Grønsund, T., & Aanestad, M. (2020). Augmenting the algorithm: Emerging human-in-the-loop work configurations. *Journal of Strategic Information Systems, 29*(2), Article 101614.
Holford, W. D. (2022). An ethical inquiry of the effect of cockpit automation on the responsibilities of airline pilots: Dissonance or meaningful control? *Journal of Business Ethics, 176*(1), 141–157.
IBM. (2023). What is software development? Retrieved October 10, 2023, from http://www.ibm.com
Johnson, P. C., Laurell, C., Ots, M., & Sandström, C. (2022). Digital innovation and the effects of artificial intelligence on firms' research and development – Automation or augmentation, exploration or exploitation? *Technological Forecasting and Social Change, 179*(121636), 1–12.
Lebovitz, S., Lifshitz-Assaf, H., & Levina, N. (2022). To engage or not to engage with AI for critical judgments: How professionals deal with opacity when using AI for medical diagnosis. *Organization Science, 33*(1), 126–148.
Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2–26.
Nguyen, N., & Nadi, S. (2022). An empirical evaluation of GitHub Copilot's code suggestions. In *Proceedings of the 19th International Conference on Mining Software Repositories* (pp. 1–11), Pittsburgh, PA.
Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science, 381*, 187–192.
Parker, S. K., & Grote, G. (2022). Automation, algorithms, and beyond: Why work design matters more than ever in a digital world. *Applied Psychology, 71*(4), 1171–1204.
Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). Sage.
Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. *arXiv preprint arXiv:2302.06590*, 1–19.
Pothukuchi, A. S., Kota, L. V., & Mallikarjunaradhya, V. (2023). Impact of generative AI on the software development lifecycle (SDLC). *International Journal of Creative Research Thought, 11*(8), 287–291.
Raisch, S., & Krakowski, S. (2021). Artificial intelligence and management: The automation–augmentation paradox. *Academy of Management Review, 46*(1), 192–210.
Sætra, H. K. (2023). Generative AI: Here to stay, but for good? *Technology in Society, 75*, 102372.
Suchman, L. (2007). *Human-machine reconfigurations: Plans and situated actions* (2nd ed.). Cambridge University Press.
Tavory, I., & Timmermans, S. (2014). *Abductive analysis: Theorizing qualitative research*. University of Chicago Press.
Tiwana, A. (2010). Systems development ambidexterity: Explaining the complementary and substitutive roles of formal and informal controls. *Journal of Management Information Systems, 27*(2), 87–126.
Treude, C. (2023). Navigating complexity in software engineering: A prototype for comparing GPT-n solutions. *arXiv preprint arXiv:2301.12169*, 1–5.
Waardenburg, L., Huysman, M., & Sergeeva, A. V. (2022). In the land of the blind, the one-eyed man is king: Knowledge brokerage in the age of learning algorithms. *Organization Science, 33*(1), 59–82.
Wilson, H. J., & Daugherty, P. R. (2019). Creating the symbiotic AI workforce of the future. *MIT Sloan Management Review, 61*(1), 1–4.