International Journal of Human–Computer Interaction

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/hihc20

Ironies of Generative AI: Understanding and Mitigating Productivity Loss in Human-AI Interaction

Auste Simkute, Lev Tankelevitch, Viktor Kewenig, Ava Elizabeth Scott, Abigail Sellen & Sean Rintel

To cite this article: Auste Simkute, Lev Tankelevitch, Viktor Kewenig, Ava Elizabeth Scott, Abigail Sellen & Sean Rintel (15 Oct 2024): Ironies of Generative AI: Understanding and Mitigating Productivity Loss in Human-AI Interaction, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2024.2405782

To link to this article: https://doi.org/10.1080/10447318.2024.2405782

Published online: 15 Oct 2024.

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2024.2405782

Taylor & Francis
Taylor & Francis Group

SURVEY ARTICLE

Ironies of Generative AI: Understanding and Mitigating Productivity Loss in Human-AI Interaction

Auste Simkute[a]*, Lev Tankelevitch[b]*, Viktor Kewenig[c], Ava Elizabeth Scott[c], Abigail Sellen[b], and Sean Rintel[b]

[a]University of Edinburgh, Edinburgh, UK; [b]Microsoft Research, Cambridge, UK; [c]University College London, London, UK

ABSTRACT
Generative AI (GenAI) systems offer opportunities to increase user productivity in many tasks, such as programming and writing. However, while they boost productivity in some studies, many others show that users are working ineffectively with GenAI systems and losing productivity. Despite the apparent novelty of these usability challenges, these 'ironies of automation' have been observed for over three decades in Human Factors research on the introduction of automation in domains such as aviation, automated driving, and intelligence. We draw on this extensive research alongside recent GenAI user studies to outline four key reasons for productivity loss with GenAI systems: a shift in users' roles from production to evaluation, unhelpful restructuring of workflows, interruptions, and a tendency for automation to make easy tasks easier and hard tasks harder. We then suggest how Human Factors research can also inform GenAI system design to mitigate productivity loss by using approaches such as continuous feedback, system personalization, ecological interface design, task stabilization, and clear task allocation. Thus, we ground developments in GenAI system usability in decades of Human Factors research, ensuring that the design of human-AI interactions in this rapidly moving field learns from history instead of repeating it.

1. Introduction

Generative artificial intelligence (GenAI) systems, such as large language models (LLMs) that can generate novel content and perform many other tasks, present myriad opportunities and challenges to humans in knowledge-intensive domains. GenAI applications have emerged in domains such as healthcare (Nova, 2023), research (Lund & Wang, 2023), writing (Chen & Chan, 2023; Dang et al., 2023), creative work (Gmeiner et al., 2023; Kulkarni et al., 2023; Oppenlaender, 2022; Pennefather, 2023a), consulting (Dell'Acqua et al., 2023), and recruitment (Budhwar et al., 2023). Software engineering has been particularly impacted, with GenAI-assisted programming tools, such as GitHub Copilot (Friedman, 2021), being increasingly used to support software engineering practices and perform tasks such as auto-completing code, translating code across languages, and answering programming questions, among others (Ross et al., 2023; Sarkar et al., 2022).

GenAI's ability to solve domain-specific problems speaks to its potential to augment human performance and transform productivity. Recent research already suggests the enormous positive impact these systems could have on workers' performance in domains including programming (Peng et al., 2023), writing (Noy & Zhang, 2023), law (Choi & Schwarcz, 2023), and consulting (Dell'Acqua et al., 2023). Based on this research, the expectation is that new tools will often free up users' time and allow them to focus on higher-level tasks, increasing their productivity. However, when using the new tools in practice, many users, such as programmers, report increased cognitive load, frustration, and time spent on the tasks that GenAI is intended to support. Feedback from Copilot users, as well as usability studies of GenAI-driven programming tools, suggest that, in some cases, using GenAI support can, in fact, lead to productivity loss. For example, software engineers and novice programmers struggle to effectively prompt systems, debug generated code, lose their state of flow when interrupted by long code suggestions, and get stuck in ineffective practices, such as reviewing, editing and then ultimately deleting suggestions (Barke et al., 2023; Prather et al., 2023; Sarkar et al., 2022). Similar observations are emerging in creative domains, where graphic (Kulkarni et al., 2023; Oppenlaender, 2022) and manufacturing (Gmeiner et al., 2023) designers struggle with prompt engineering and other aspects of GenAI interaction. This suggests that the potential of GenAI systems to boost productivity may not be guaranteed, evenly distributed, or fully exploited.

These observations mirror the long line of Human Factors studies exploring human-automation interactions in safety-critical systems in aviation, industrial plants, and other areas (Endsley, 2017; Lee & Seppelt, 2009). Indeed, they reflect the "ironies of automation" (Bainbridge, 1983), which capture the idea that the more advanced an automated system is, the

CONTACT Sean Rintel serintel@microsoft.com Microsoft Research, Cambridge, UK

*Both authors contributed equally to this research.

more important the human operator may be.[1] Despite automation taking over human manual control in areas where it is expected to provide superior performance, humans are still left to supervise automation. However, operators might have insufficient support to supervise, and so instead of being supported by automation, they find themselves cognitively overburdened, trying to decipher systems' outputs and spot errors. Similarly, in the context of GenAI, users' roles have shifted from producing output to evaluating it, often with little contextual information and situational awareness. This is exacerbated by GenAI tools' ability to produce outputs at a capacity too demanding for adequate evaluation, with questionable reliability, and with poor explainability (Chen et al., 2023; Liao & Vaughan, 2023; Schellaert et al., 2023). Moreover, poor system and interface design can result in unhelpful restructuring of workflows, which increases cognitive load and undermines productivity gains (Bainbridge, 1983). This is echoed in programmers' experiences and feedback around Copilot features (Barke et al., 2023; Prather et al., 2023; Sarkar et al., 2022), with evidence of similar effects emerging in other domains (Dang et al., 2023; Gmeiner et al., 2023; Gu et al., 2023). Finally, as a result of which tasks get automated, as well as poor system design, automation often makes easy tasks easier while making hard tasks even harder. This same pattern is now being observed in usability studies of GenAI systems (Barke et al., 2023; Sarkar et al., 2022).

In this paper, we answer recent calls for bridging Human Factors and Human-Computer Interaction research to advance human augmentation by AI and human-AI interactions (Chignell et al., 2023). Extrapolating from over 30 years of Human Factors research on the "ironies" of human-automation and productivity loss, we synthesize an overview of the usability and productivity challenges observed in recent GenAI user studies. We demonstrate how these challenges emerging in GenAI systems mirror those experienced by operators when automation was introduced to their workflows decades ago. Based on these parallels, we highlight key areas of productivity loss and provide insights into the human factors leading to these issues, exploring aspects including feedback, situational awareness, cognitive workload, workflow disruptions and others. We focus primarily on programming due to the early adoption of tools like GitHub Copilot and the accompanying usability research, but we also reflect on emerging studies from other domains, such as healthcare, writing, and design, showing that these issues are not limited to a single domain. Moreover, we discuss potential design solutions, emphasizing the importance of following the Human Factors principles of feedback and flexibility when designing GenAI systems. We suggest that the fast-paced innovation of GenAI will benefit from the decades of Human Factors research in order to design GenAI systems that truly harness the full productivity potential of this technology. In summary, our paper makes the following contributions:

1. Based on Human Factors research and a synthesis of recent GenAI studies, we identify key challenges that can lead to productivity loss, grouped into four broad categories: (i) the production-to-evaluation shift, (ii) unhelpful workflow restructuring, (iii) task interruptions, and (iv) task-complexity polarization.
2. We provide potential design directions from Human Factors research that address each category of challenges: (i) continuous feedback, (ii) system personalization, (iii) ecological interface design, (iv) main task stabilization and timing, and (v) clear task allocation. Throughout, we also emphasize the importance of following the Human Factors principles of feedback and flexibility.
3. We motivate further research into the impact of GenAI systems on aspects such as situational awareness and cognitive workload to better understand systems' unintended effects on human performance. We also encourage future researchers to take advantage of the plethora of relevant Human Factors work to enrich their understanding of existing human-GenAI interaction issues and anticipate others.

2. Methodology

We used a narrative review approach to identify, analyze, and synthesize the relevant literature into the presented themes (Sukhera, 2022). This proceeded in two stages. First, we distilled the key challenges to effective human work with automation from Bainbridge's seminal work on the "Ironies of Automation" (Bainbridge, 1983). This work highlights that increased advancement of automation systems has also, ironically, increased the importance of human oversight and, in some cases, has complicated human work. We focused on the following challenges emphasized by Bainbridge:

1. the change in humans' role from production to monitoring
2. monitoring and overtaking challenges resulting from increased system complexity and reduced awareness of system states
3. poor design solutions failing to integrate other technologies and support human workflows
4. automation effects of increased cognitive workload in already difficult tasks

We did not focus on challenges relating to reduced opportunities for skill development and increased expertise demands for monitoring systems. These issues are undoubtedly important but were too early to be observed in research on GenAI at the time of this paper's writing, and should be explored in future work. The four identified challenges served as a starting point, and we used further Human Factors research to expand on these and their influencing factors. As our aim was only to illustrate the challenges identified in Human Factors research *in order to analyze their parallels in emerging GenAI research*, our search of Human Factors research was not exhaustive, and focused on highly cited work that was conceptually related to Bainbridge's work (Bainbridge, 1983). We also reviewed the references in identified articles in a snowballing approach. Based on this search, we refined the four conceptual themes and influencing factors presented here:

1. as the human role shifts from production to monitoring, the human ability to effectively monitor automation is challenged by reduced situational awareness, which is exacerbated by increased automation capacity, increased system complexity and opaqueness, and reliability issues
2. effective human work is disrupted by automation, which introduces workflow changes, such as the loss of task sequence or feedback
3. poor design solutions for automation systems, such as poor timing or suggestions, can interrupt human work
4. automation makes easy tasks easier, while making cognitively demanding tasks more difficult due to automation monitoring and output management demands, and other challenges

Finally, with these themes in mind, we conducted a search of the literature on Generative AI. We searched Google Scholar and the Association of Computing Machinery Digital Library (ACM DL) using the keywords: *generative AI*, *large language model*, *LLM*, *ChatGPT*. Articles had to be studies of GenAI system usage, reviews, or conceptual syntheses related to Generative AI usability (rather than technical evaluations of systems). Given the early stage of GenAI adoption, most articles focused on areas of fast adoption, including programming, writing, creativity and other aspects of knowledge work. As we focused specifically on productivity losses (rather than gains), we selected articles that indicated human-AI interaction challenges resulting in reported productivity losses or ineffective use of GenAI systems. Importantly, we acknowledge that productivity gains are plausible and have indeed been observed, as noted in Section 1 (e.g., Dell'Acqua et al., 2023; Noy & Zhang, 2023; Peng et al., 2023). Articles also had to be in English, and be available in their entirety. Given the rapid pace of progress in the field, we included pre-prints as well as peer-reviewed articles. The cut-off date for the search was 2024-01-20. As our approach was a narrative review, our search was not systematic or exhaustive, and there may be articles we have missed. Finally, we aligned our observations from the research on GenAI with the four conceptual themes that stemmed from the Human Factors literature, finalizing the four productivity challenges of GenAI automation: (1) the production-to-evaluation shift, (2) unhelpful workflow restructuring, (3) task interruptions, and (4) task-complexity polarization. Table 1 presents the articles we included in Section 3 on the productivity challenges of GenAI. For each article, it summarizes the domain, task or focus area, methodology, and relevant theme (i.e., sub-section in which it is cited).

3. Productivity challenges of Generative AI automation

Here, we outline the key productivity challenges that have been observed in human-automation interaction over decades of Human Factors research and are now becoming apparent in user studies of GenAI systems. Our focus is on GenAI *systems*, the integrated whole comprising GenAI models and interfaces. Some challenges pertain to *GenAI models* (e.g., issues around prompting), and some pertain to *interface design* (e.g., issues around task interruptions).

We begin with challenges related to the shift from manual control or production to a more passive supervisory role of the user, such as monitoring and evaluation of AI outputs (Section 3.1). We explore specific aspects related to this shift, such as reduced situational awareness, the contributory factors of automation's high capacity, complexity and opaqueness, reliability, and potential resultant complacency and over-reliance. We then outline how the introduction of automation such as GenAI can unhelpfully restructure users' workflows, stifling their productivity (Section 3.2). We focus on how the introduction of new tasks, such as prompting or output adaptation, can affect user performance and how workflow restructuring can lead to loss of task sequence and feedback. We also explore the influence that task interruptions from AI suggestions can have on users' productivity (Section 3.3). Finally, we explore how automation such as GenAI can paradoxically lead to easy tasks being made easier and hard tasks made harder, a phenomenon we refer to as "task-complexity polarization" (also known as "clumsy automation" in Human Factors research (Wiener & Curry (1980); Section 3.4)). Figure 1 outlines the four types of challenges.

3.1. The production-to-evaluation shift

Decades ago, the introduction of automation shifted many manual control tasks to monitoring tasks, leaving humans to supervise the automation (Sheridan, 2012). However, monitoring (or vigilance) is tedious and requires attention, and can, therefore, paradoxically impose a considerable workload on humans (Grubb et al., 1995; Warm et al., 2008). For example, when automation was introduced in the aviation context (e.g., detection of air traffic in an aircraft's vicinity), pilots' workload was not reduced but moved to supervising activity. Pilots reported spending more time interacting with automation and trying to understand it instead of concentrating their efforts on their primary task of flying the aircraft (Rudisill, 1995). In other domains, operators supervising automation also spent a significant amount of time and effort learning how to manage the new technology (Baxter et al., 2012) (see Section 3.2.2).

GenAI workflows have introduced a similar shift from manual control to monitoring—in this case, from the production of outputs to their evaluation—with Sarkar (2023) terming this new user role "critical integration" (see Figure 1a).[2] In AI-assisted coding, users spend extended periods reviewing and validating code suggestions (Barke et al., 2023; Vaithilingam et al., 2022), sometimes at the expense of other productive tasks like writing code or running tests (Vaithilingam et al., 2022; Weisz et al., 2022). Some programmers have said that working with Copilot felt like a "proofreading task" (Weisz et al., 2022). Accordingly, in

Table 1. Generative AI articles included in the review of productivity challenges.

Article	Domain	Task (or focus)	Methodology	Challenge theme*
Chen et al. (2023)	General	–	Conceptual review	1
Dell'Acqua et al. (2023)		Creating recipes	User study; quantitative (game activity, outcomes); n = 220 (students)	1
Drosos et al. (2020)	Programming	AI programming assistance by example	User study; mixed (formative interview [n = 7; professionals], activity, questionnaire [n = 12; professionals])	1
Kazemitabaar et al. (2023)	Programming	Code generation in self-paced learning	User study; mixed (longitudinal tasks, activity, knowledge retention tests); n = 69 (students)	1
Liao et al. (2023)	Design	UX design for AI systems	Formative study; mixed (think-aloud, interview); n = 23 (professionals)	1
Preiksaitis et al. (2023)	Healthcare	Physician assistance	Conceptual review	1
Sarkar (2023)	Knowledge work and creativity	–	Conceptual review	1
Schellaert et al. (2023)	General	–	Conceptual review	1
Srinivasa Ragavan et al. (2022)	Data science	Formula generation in spreadsheets (end-user programming)	User study; mixed (think-aloud, task measures, interview); n = 20 (professionals)	1
Sun et al. (2022)	Programming	Prompting, code translation, code auto-completion	Formative study; qualitative (scenario-based design workshops); n = 43 (professionals)	1
Weisz et al. (2022)	Programming	Code translation	User study; mixed (qualitative evaluations; questionnaire); n = 32 (professionals)	1
Woodruff et al. (2023)	Knowledge work	–	Formative study; qualitative (participatory research workshops); n = 54 (professionals)	1
Zamfirescu-Pereira et al. (2023)	General	Recreating an expert as a chatbot	User study; mixed (think-aloud); n = 10 (graduate students, professionals)	1
Noy and Zhang (2023)	Writing	Knowledge work tasks (reports, press releases, analysis plans, emails etc.)	User study; mixed (experimental longitudinal task, evaluations, questionnaire); n = 444 (professionals)	1
Arnold et al. (2021)	Writing	Writing reviews for books, film, and travel	User study; quantitative (descriptive prompt evaluation questionnaire); n = 30 (MTurk)	2
Calderwood et al. (2020)	Writing	Creative writing of novels	User study; qualitative (think-aloud, interview); n = 4 (novelists)	2
Jayagopal et al. (2022)	Programming	Learnability of code generation tools	User study; qualitative (think-aloud, interview); n = 22 (students)	2
Jiang et al. (2022)	Programming	Learnability of code generation tools	User study; qualitative (longitudinal take-home tasks, video observation, activity, interview); n = 14 (professionals with mixed experience)	2
Kulkarni et al. (2023)	Design	Text-to-image generation for visual design	User study; mixed (video observation, questionnaire, design evaluation); n = 16 (non-professional designers)	2
Oppenlaender (2022)	Creativity	Text-to-image generation for visual art	Field study; qualitative (online ethnography); n = unknown (text-to-image tool community)	2
Pennefather (2023a)	Creativity	–	Conceptual review	2
Pennefather (2023b)	Creativity	–	Conceptual review	2
Xu et al. (2022)	Programming and data analysis	Code generation and retrieval for file manipulation, machine learning, data visualisation etc.	User study; mixed (task performance, code logs, questionnaire); n = 31 (students, freelancers)	2
Bhat et al. (2023)	Writing	Writing film reviews	User study; qualitative (concurrent and retrospective think-aloud); n = 14 (students)	3
Clark et al. (2018)	Writing	Image-to-text generation for creative writing of stories	User study; mixed (questionnaire, interview); n = 36 (MTurk)	3
Frey and Osborne (2023)	General	–	Conceptual review	4
Liao and Vaughan (2023)	General	–	Conceptual review	4
Chen and Chan (2023)	Writing	Ad copywriting	User study; mixed (ad clicks, text analysis, questionnaire); n = 355 (Prolific)	1,2
Gmeiner et al. (2023)	Design	Manufacturing design generation	User study; qualitative (think-aloud); n = 14 (study 1; professionals) n = 6 (study 2; students)	1,2
Dang et al. (2023)	Writing	Creative and argumentative writing	User study; mixed (formative interview [n = 6; Prolific], experiment with activity [n = 129; Prolific], text analysis; open feedback themes)	1,2,3
Barke et al. (2023)	Programming	Code generation and other AI assistance	User study; qualitative (observation, interview); n = 20 (students, professionals)	1,2,3,4
Prather et al. (2023)	Programming	Code generation and other AI assistance for novices	User study; mixed (observation, think-aloud, interview); n = 19 (students)	1,2,3,4
Sarkar et al. (2022)	Programming	–	Conceptual review	1,2,3,4
Vaithilingam et al. (2022)	Programming	Code generation	User study; mixed (observation, task completion, questionnaire); n = 24 (students)	1,2,3,4
Ross et al. (2023)	Programming	Code generation	User study; mixed (longitudinal tasks; conversation logs; event logs, questionnaire); n = 42 (professionals)	1,2,4

(*continued*)

Table 1. Continued.

Article	Domain	Task (or focus)	Methodology	Challenge theme*
Gu et al. (2023a)	Data science	Understanding and verification of AI assistance for data analysis	User study; qualitative (observation, interview); n = 13 (students and professionals)	1,3
Mcnutt et al. (2023)	Data science	Coding assistance in notebooks	User study; qualitative (interview, design probe); n = 15 (professionals)	1,3
Weisz et al. (2021)	Programming	Code translation	User study; qualitative (interview, design probe); n = 11 (professionals)	1,3
Choi and Schwarcz (2023)	Law	AI assistance for legal reasoning (law exams)	User study; mixed (exam results, written response analysis); n = 48 (students)	1,3,4
Gu et al. (2023)	Data science	AI assistance for data analysis execution and planning	User study; qualitative (observation, interview); n = 22 (professionals)	1,3,4

*'Challenge theme' refers to (1) the production-to-evaluation shift, (2) unhelpful workflow restructuring, (3) task interruptions, and (4) task-complexity polarization (as per Section 3).

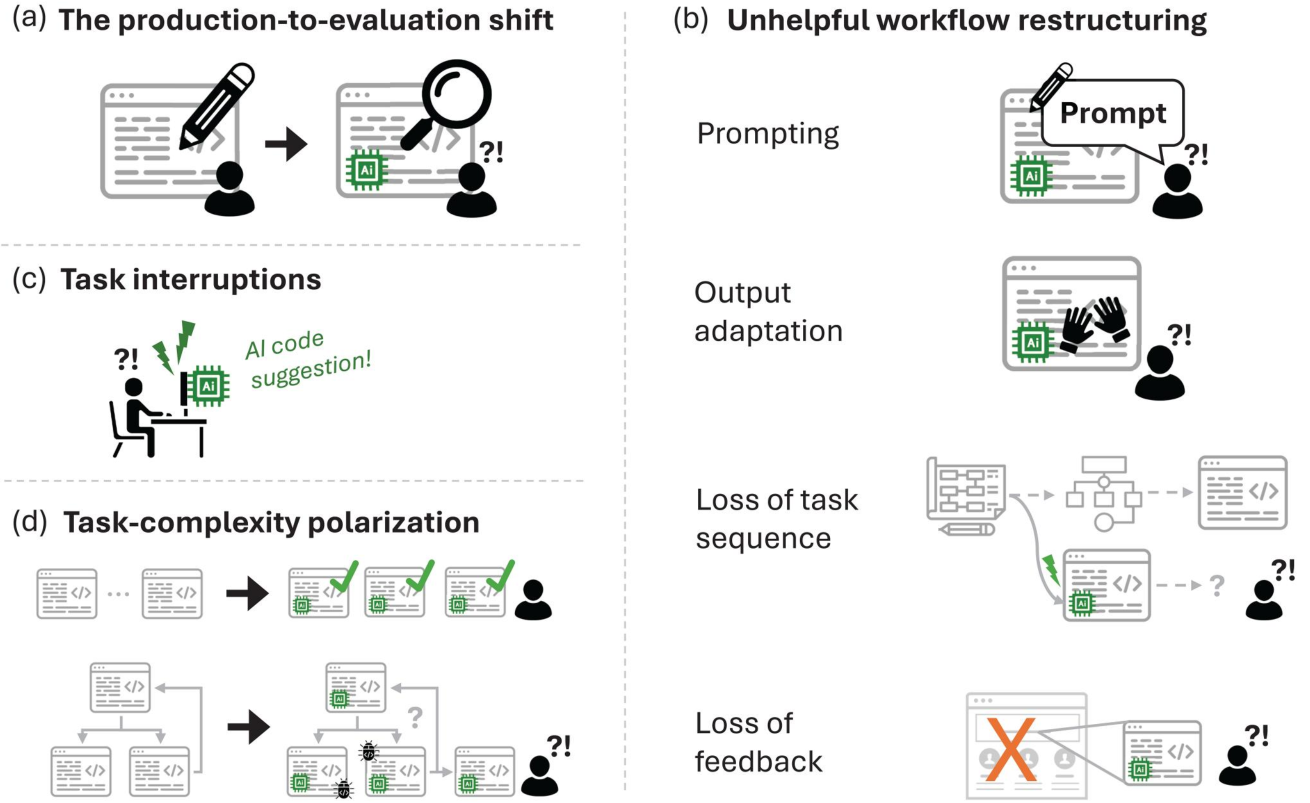

Figure 1. Productivity challenges of Generative AI automation: (a) the production-to-evaluation shift, in which users' situational awareness of their working environment is reduced, increasing the cognitive demand required to evaluate AI outputs; (b) unhelpful workflow restructuring, including the addition of new challenging tasks of prompting systems and adapting outputs, a loss of task sequence due to AI suggestions or other changes, and a loss of feedback when AI suggestions are presented without the relevant context; (c) task interruptions from automated AI suggestions; and (d) task-complexity polarization, in which automation tends to make easy tasks easier and hard tasks harder when implemented in practice.

some cases, working with current GenAI systems might not benefit users relative to a more manual approach. For example, when Vaithilingam et al. (2022) compared programmers' experience with Copilot versus traditional autocomplete, they found that Copilot participants failed to complete their tasks more often. When they did complete them, they were no faster than those who used autocomplete. Vaithilingam et al. (2022) suggest that assessing the correctness of generated code created an efficiency bottleneck, often leading participants down an unsuccessful path of debugging. This not only took time out of their main task, thereby decreasing productivity, but also required a significant amount of cognitive effort. A similar shift towards evaluation of outputs has been observed in consultancy (Dell'Acqua et al., 2023), and in creative writing, where most of the writing time is now being replaced by editing AI-generated text (Noy & Zhang, 2023). Overall, practitioners from various domains, such as advertising, education, business and law, overwhelmingly agree that GenAI outputs will require supervision (Woodruff et al., 2023).

3.1.1. Reduced situational awareness

A key reason why monitoring automation (like evaluating GenAI outputs) is so demanding is that, due to processing being relatively more passive, it reduces operators' situational awareness: their perception of data and elements of the situation, comprehension of the situation, and the projection of future status (Endsley, 1995). Passive processing resulting in decreased situation awareness has been observed with experienced air traffic controllers (Endsley et al., 1997; Metzger & Parasuraman, 2001) and in other automated tasks (Manzey et al., 2012). Low situation awareness significantly decreased operators' ability to effectively monitor and observe errors in the automation and to determine whether the given situation is outside the bounds of automation capabilities (Jones & Endsley, 1996).

Evidence suggests that users of GenAI systems similarly experience reduced situational awareness. For example, participants in Vaithilingam et al. (2022) reported that their debugging of AI-generated code was hampered because they could not use their intuition about where the bug might be and instead ended up refactoring or abandoning the code entirely. This is echoed by participants in Barke et al. (2023) who say, e.g., *"I don't see the error immediately, and unfortunately, because this is generated, I don't understand it as well as I feel like I would've if I had written it."* Participants in Weisz et al. (2022) noted a trade-off between writing and debugging code, citing a lack of comprehension for AI-generated code translation and *"spotting errors in 'foreign' code"* as challenges. Similarly, in data science, users report feeling out of control when unable to understand AI-generated suggestions (Mcnutt et al., 2023) and highlight readability "as being a critical feature of usable synthesized code" (Drosos et al., 2020). For novices in a domain, this reduced situational awareness can be particularly challenging, as noted in Prather et al. (2023). In the healthcare domain, AI-generated medical records may lead physicians to become detached from patients' medical history, and in turn spend additional time analysing GenAI outputs to compensate for the missing information (Preiksaitis et al., 2023). These findings indicate that gaining situational awareness of GenAI output is demanding and takes users' time and attention away from proceeding with the main task.

The next sections describe factors that can exacerbate already reduced situational awareness, as well as a potential outcome of the "monitoring" challenge of automation: complacency and over-reliance.

3.1.2. Factors exacerbating low situational awareness

Automation research shows that low situational awareness can be exacerbated by several factors, including automation's high output capacity and systems' complexity, opaqueness, and low reliability.[3]

3.1.2.1. High automation capacity. Monitoring automation—in this case, evaluating GenAI output—is, ironically, made more difficult by the high capacity of automation, which makes it challenging to understand and anticipate system behaviour. For example, when traders in the digital stock exchange changed roles from executing to monitoring trades, they underperformed as they were unable to effectively monitor the trades in real-time (Haldane & May, 2011). As such, they resorted to monitoring them at a higher level of abstraction and required additional resources to process that information, thereby missing more trades that were executed in the meantime.

Similarly, GenAI is notable for its high capacity in outputting content, such as entire documents or software programs, or multiple simultaneous suggestions (Barke et al., 2023; Chen et al., 2023; Sarkar et al., 2022; Schellaert et al., 2023). This makes evaluating these outputs challenging. In GenAI-assisted coding (Barke et al., 2023), found that users deal with the plethora of code suggestions by quickly assessing them using a "pattern matching" approach, where they search for the presence of certain keywords or control structures. The impact of high output capacity can be worsened by poor system design. For example, participants in Barke et al. (2023) noted that the separation of Copilot's multi-suggestion pane from their main code increased cognitive load due to the lack of relevant code context when reviewing and trying to differentiate the code suggestions.

3.1.2.2. Automation complexity and opaqueness. Evaluation is further challenged by the complexity and opaqueness (i.e., poor explainability) of automated systems, which can reduce situational awareness. More features and modes create more possible interactions among system components and a corresponding reduction in system predictability as the system increasingly considers multiple factors or component states (Endsley et al., 2003). This can lead to unfamiliar and infrequent system states, which add to the challenge of comprehending systems' workings. For example, even well-trained pilots were startled by unexpected flight automation system behaviours in complex systems (Wiener & Curry, 1980). System opaqueness similarly reduces situational awareness and affects monitoring, for example, in the use of automation aids in local government organizations (Lindgren, 2023). Put another way, system complexity and opaqueness make it more difficult for users to create an accurate mental model of the system needed for the correct interpretation of information, including situations where manual control will be needed (Baxter et al., 2012).

The opaqueness and complexity of GenAI systems are cited as key barriers to usability, including prompting and evaluating outputs (Liao et al., 2023; Sun et al., 2022). One issue, termed "fuzzy abstraction matching" (Sarkar et al., 2022), describes the opaque relationship between the content of prompts and the resultant output, driven by the flexibility of GenAI models to produce plausible but potentially incorrect outputs for prompts with a wide range of abstraction. Another issue is the sheer range of implicit and explicit parameters available to users, which increases systems' complexity (Schellaert et al., 2023). This not only makes prompting a challenge (e.g., Dang et al., 2023; Zamfirescu-Pereira et al., 2023) but also the evaluation of outputs (e.g., Barke et al., 2023; Liang et al., 2023; Weisz et al., 2021) as the two are inextricably intertwined in current systems. The top usability issue for AI programming assistants, as surveyed in Liang et al. (2023), is not knowing what part of users' code or comments the GenAI system is relying on to produce output. Likewise, one participant in Barke et al. (2023) laments the challenge of evaluating code suggestions, *"it might be nice if it could highlight what it's doing or which parts are different, just something that gives me clues as to why I should pick one over the other."*

3.1.2.3. Automation reliability. The challenge of monitoring automation is further exacerbated by systems' unreliability. For example Metzger and Parasuraman (2005), found that air traffic controllers who worked with unreliable automation to make aircraft-to-aircraft conflict decisions were unable to monitor the systems effectively and were ultimately better at detecting conflicts without automation. Similar impacts of reliability were found for target detection and decision-making tasks (Galster et al., 2001; Wickens et al., 2000). Evaluation of GenAI outputs is likewise exacerbated by the non-determinism of GenAI models (Schellaert et al., 2023), which can produce different outputs for the same input, resulting in lower reliability from the user's perspective. More than merely being non-deterministic, GenAI systems can introduce subtle or non-intuitive errors into outputs, particularly in long outputs such as multi-line code suggestions (Sarkar et al., 2022) (see also Section 3.4). Woodruff et al. (2023) found that knowledge workers across domains overwhelmingly cited a lack of reliability as a key reason for humans having to review GenAI outputs. Example concerns ranged from violation of brand standards and copyrights in generated content, to inaccuracies in legal documents (Woodruff et al., 2023).

3.1.3. Potential complacency and over-reliance

Ultimately, as Human Factors research shows, the shift from production to evaluation, the resultant reduced situational awareness, and additional workload can result in complacency, over-reliance on systems, and increased errors (Parasuraman & Riley, 1997). Trying to recover from these errors further increases the workload and, as workload affects monitoring ability, can create a vicious cycle. In high-workload situations, there are fewer attentional resources available for monitoring imperfect automation, resulting in a risk of errors (McBride et al., 2011) and significantly longer error detection time (Dixon et al., 2005). Complacency due to high-workload conditions has been observed in aviation, where pilots would fail to conduct sufficient checks of system state (Funk et al., 1999; Parasuraman et al., 1993). In a spacecraft simulator study, operators did not properly assess the recommendations and simply complied with them, which resulted in missed failures (Manzey et al., 2006).

An increase in complacency and over-reliance related to output evaluation has been observed in GenAI user studies. For example, when verifying the correctness of AI-generated code, some programmers reported skimming through the output rather than reading and evaluating the code rigorously (Sarkar et al., 2022; Vaithilingam et al., 2022). This is especially prevalent for those with less experience, such as end-user programmers (Sarkar et al., 2022) or novices (Kazemitabaar et al., 2023; Prather et al., 2023). In some cases, this has led to errors that users either missed (Ross et al., 2023) or had to later spend time debugging (Vaithilingam et al., 2022). Notably, in advertising, both expert and non-expert writers showed overconfidence in the quality of AI-generated drafts, failing to thoroughly revise them (Chen & Chan, 2023). Complacency and over-reliance have also been reported in the data science domain (Gu et al., 2023; 2023; Srinivasa Ragavan et al., 2022); in the legal domain, where "AI-assisted exams were more likely to miss hidden issues" (Choi & Schwarcz, 2023); and in the design domain, where one participant commented, *"I would never design it like that, but this [GenAI system] thinks it can do it like that […] But this is what it gave me, so I don't have a problem with that."* (Gmeiner et al., 2023). Over-reliance has been shown to lead to decreased performance; for example, management consultants showed overall poorer performance when they blindly adopted AI-generated outputs (Dell'Acqua et al., 2023).

3.2. Unhelpful workflow restructuring

Automation can restructure workflows in unhelpful ways by introducing new challenging tasks, disrupting familiar task sequences, and removing informative feedback (Figure 1b). This changes what strategies operators use, how they perceive information, and how they act in a specific context, potentially leading to ineffective use of freed-up time and cognitive resources. Thus, rather than reducing what they work on when all or part of tasks are automated, people instead rely on different strategies for working on that task (Bainbridge, 1983). For example, when automation introduces new tasks in operators' workflow, disrupting their familiar workflow, they struggle to adapt their strategies (Klein et al., 2006). Likewise, when automation unexpectedly increases the workload during peak times, operators tailor the system or the task to accommodate the automation needs (Cork et al., 1998). If tailoring the system is not possible, users are forced to tailor their tasks, often having to add new tasks to their workload (Cork et al., 1998). For example, physicians using automation aids learned how to manipulate monitors displaying physiological data to fit their work strategies. However, because this manipulation was an additional task physicians had to perform, they avoided using the system in high-workload situations (Cork et al., 1998). Moreover, when automation changes the familiar sequence of the task, for example, by removing a step, operators make errors and repeat their actions. For example, physicians might forget to record a dose of medication in a log and mistakenly repeat the procedure (Altmann & Gregory Trafton, 2015). Finally, when automation removes the critical feedback necessary to make an informed decision, operators succumb to errors. For example, in aviation, pilots were missing critical failures due to relevant information from vibration and smell being lost in the automation process (Moray et al., 1986).

3.2.1. Prompting as a new task

The central role of prompting in GenAI systems is one major way in which such systems are restructuring workflows. Studies show that users struggle with prompting, dedicating considerable time and effort to it. In Xu et al. (2022), programmers using a code generation plugin invested significant effort in experimenting with prompts to understand how their queries worked best. Likewise, in Jiang et al.

(2022), participants using an LLM-driven tool developed various strategies to deal with model failures, for example, rewording prompts by reducing the scope of the request or looking for alternative wording. Trying to adapt prompts is a cognitively demanding task, as participants must form a mental model of what the model can work with (the problem of "fuzzy abstraction matching" (Sarkar et al., 2022)). Beyond being demanding, prompting may interfere with other aspects of users' workflows. For example, Copilot users' code commenting workflows can change. Participants in Barke et al. (2023) wrote and re-wrote detailed comments intended for Copilot, hoping to increase the context available to the system, and then also spent time deleting comments for Copilot after the fact.

Similar workflow changes were observed in the design and writing domains. For example, one non-professional designer in Kulkarni et al. (2023) complains, *"it felt like I was fighting it… I felt like it was helpful, but I also felt like I had to massage every word and select every character very carefully not to upset it so that it could generate something I wanted"* (see also Oppenlaender (2022)). Dang et al. (2023) distinguish between *diegetic* prompts (instructions implicitly conveyed by inputted content to be acted on by the system) and *non-diegetic* prompts (instructions explicitly conveyed to the system). The latter is particularly disruptive to users' workflows in the writing domain, as they "[force] writers to shift from thinking about their narrative or argument to thinking about instructions to the system" (Dang et al., 2023; see also Yuan et al., 2022), a finding echoed in the coding domain (Jayagopal et al., 2022). More broadly, prompting seems to function as a new task that competes with other workflow tasks, adding to the workload and potentially increasing over-reliance on automation as users invest more time into it (Endsley & Rodgers, 2016). Indeed, this might explain why some users try to coerce AI output to be useful (see Section 3.2.2) or become complacent in reviewing it (see Section 3.1.3).

3.2.2. Output adaptation as a new task

Another workflow change with GenAI is the need to adapt generated output, effectively a new type of task. In Barke et al. (2023), several participants chose to adapt Copilot suggestions to use as a template for their code. Rather than accepting or rejecting code entirely, they deleted and edited parts so they would not have to write it from scratch. Others used the strategy of slowly breaking down large blocks of code and adapting them as needed or cherry-picking code from multiple suggestions. This suggests that the use of suggestions is not straightforward, and complex strategies are created by programmers for their workflows. The productivity gains of these workflow changes remain unknown, and although participants in Barke et al. (2023) found them helpful, they may ultimately decrease productivity. For example, if the adapted code has an error, the necessary debugging will add to the workload, as observed in, e.g. Barke et al. (2023) and Vaithilingam et al. (2022). In the design domain Gmeiner et al. (2023), found that manufacturing designers struggled with GenAI assistance. In this case, the GenAI system was found to be "dominating the design process," and "designers either gave up and accepted unsatisfying results, improvised 'hacky' strategies to work around the AI or abandoned the AI assistance altogether and proceeded to work manually."

The productivity gains or losses of output adaptation may depend on users' expertise. In Vaithilingam et al. (2022), participants of varying levels of expertise struggled to adapt the code suggestions, and many abandoned them entirely, thereby losing time. Among novices, code adaptation may particularly reduce productivity. Prather et al. (2023) studied novice programmers working with Copilot, identifying an unproductive interaction mode they termed "shepherding," in which participants spent considerable time trying to coerce Copilot to produce useful code. This included accepting suggestions, then deleting them without any changes, or spending considerable time adapting suggestions without writing any code of their own. More broadly, the assortment of code adaptation strategies reflects a new layer of complex tasks that programmers are introducing to their workflow to accommodate and effectively use GenAI. Ironically, the more complex the code, the more powerful the potential productivity benefits, yet the more intricate and time-consuming the process of reviewing and adaptation might become (e.g., Barke et al., 2023).

3.2.3. Loss of task sequence

Workflow changes can also lead to difficulty in following the familiar sequence of steps in a task. Many tasks have sequential constraints, a set of steps that have to be performed in a specific order. When one of the steps is skipped or repeated, errors can occur (Altmann & Gregory Trafton, 2015). To perform a task correctly under sequential constraints, the cognitive system has to keep track of where it is in the sequence and select the correct next step when one step is complete (Altmann & Gregory Trafton, 2015). Changes in the structure of the task can make it difficult for one to follow the natural sequence of the steps. Automation research showed that operators' reactions are slower and less integrated when they cannot generate the sequence of activity themselves (Janssen et al., 2015). Not having a task structure to follow also prevents users from monitoring their own progress. Under manual control, users obtain information about the results of their actions and then can correct themselves (Smith, 1979). Without this information, they are more likely to repeat the same type of errors (Wiener & Curry, 1980).

In GenAI workflows, auto-suggestions generated by the system or the requirement to prompt systems are examples of disruptions to the familiar sequence of steps, which could lead to productivity loss, as evidenced in recent studies. In the coding domain Barke et al. (2023), found that long code suggestions in Copilot disrupted users' task sequence by "forcing them to jump in to write code before coming up with a high-level architectural design." Analogously, in the design domain Gmeiner et al. (2023), found that the need for prompting meant that designers had to specify required parameters in advance instead of working step-by-step, thereby requiring designers "to think through the design problem in advance, which is challenging and different from the usual iterative design process." This loss of task sequence can be particularly disruptive among novices. For example Prather et al. (2023), identified an unproductive interaction pattern among novice programmers called "drifting," in which participants spent time adapting code suggestions, then deleting them, and repeating the cycle. Thus, they unproductively drifted from suggestion to suggestion without a direction. Moreover, this was exacerbated if the generated output contained an error, which sent users down a "debugging rabbithole," in which they spent time trying to adapt incorrect code rather than focusing on the correct solution (Prather et al., 2023). In film production, Pennefather (2023a) observed a filmmaker working with GenAI that had to shift between multiple software, struggling to identify which was the most suitable for which part of their creative process. The creative described the process as *"an exercise in randomization and an attempt to control chaos"* (Pennefather, 2023b) (see Oppenlaender (2022) for similar observations with creative text-to-image generation workflows).

Task sequence can also be obscured when a large part of the workflow is automated. For example, both expert and non-expert copywriters were anchored to GenAI suggestions and produced lower-quality results when GenAI generated the majority of the text versus when it only provided feedback to users (Chen & Chan, 2023). Similarly, professional novel writers (Calderwood et al., 2020) and inexperienced writers working with GenAI (Arnold et al., 2021) found guidance more useful than the injection of generated text. In these examples, users' familiar task sequences in a given domain are disrupted by aspects of GenAI systems.

3.2.4. Loss of feedback

Automation can deprive users of key feedback needed to assess the state of automation and its ability to perform tasks. For example, automation can cause users to change from processing raw data to processing integrated information. Introducing automation into paper-making plants moved operators away from the information associated with informal feedback (e.g., smells, sounds) and put them in control rooms (Lee & Seppelt, 2009). This change not only required operators to learn the task of plant control but also deprived them of contextual information that could help them diagnose automation failures and intervene appropriately. Similarly, in aviation, relevant information from vibration and smell was lost in the automation of process control operations (Moray et al., 1986), and the automation of auto-feathering systems in commercial aircraft removed the signal telling pilots about engine shut-downs (Billings, 1991). The lack of transparency or supporting contextual feedback often only becomes an issue under system failures when operators lack the relevant detail for detecting or addressing them (Endsley et al., 1997).

An analogous loss of feedback has also been observed in GenAI-assisted coding. Participants in Vaithilingam et al. (2022) noted that, in comparison to internet search tools like Stack Overflow, Copilot lacked additional information, such as discussions, explanations, and comparisons of code solutions. This sentiment was echoed by participants in Ross et al. (2023), who noted that their AI code assistant "lacked the 'multiple answers' … and 'rich social commentary' … that accompanies answers on Q&A sites." Thus, programmers using these tools see the code, comments, and data but miss out on the rich feedback that is usually available when programming with access to various media sources.

3.3. Task interruptions

Another aspect stifling productivity gains from GenAI is task interruption (Figure 1c). There are various cognitive costs related to interruptions (Altmann & Gregory Trafton, 2002; Janssen et al., 2011; Salvucci & Taatgen, 2011). Interruptions can disrupt the user's thought processes (Altmann et al., 2014) and initiate a switch between tasks that requires time and cognitive resources, which negatively affects performance (Janssen et al., 2015). Particularly long and complex interruptions significantly disrupt people's ability to resume their original tasks (Mark et al., 2008; 2012; Monk et al., 2008). Moreover, interruptions can also break the user's flow state (Taekman & Shelley, 2010).

Copilot auto-suggestions have been shown to interrupt users' main tasks, with programmers referring to Copilot auto-suggestions as *"interrupting their thoughts"* (Sarkar et al., 2022), *"intrusive,"* and *"messing up thought process"* (Prather et al., 2023). Accordingly, some programmers decide to switch the suggestions off to avoid distractions (Sarkar et al., 2022) or chose to disable the tool completely (Barke et al., 2023), while others admitted being *"tempted to follow what it's saying instead of just thinking about it"* (Prather et al., 2023). Beyond programming, similar interruptions are reported in the writing domain (Bhat et al., 2023; Clark et al., 2018; Dang et al., 2023) and in data science (Gu et al., 2023; Mcnutt et al., 2023).

Particularly distracting are the long, multi-line code suggestions. For example, these have been observed to break programmers' flow when in "acceleration mode," a state in which programmers work with well-formed intent, relative to an "exploration mode," in which programmers start a novel task or debug (Barke et al., 2023). Programmers were distracted from their flow as they felt compelled to read the code. If they chose to consider it, they then had to review it for errors. Thus, long code suggestions force users to switch back and forth between writing and reviewing code, and if the code has errors, they must then switch to debugging (Vaithilingam et al., 2022). This may be particularly disruptive if the errors are unrelated to the current task focus, as found in Weisz et al. (2021). Interruptions may be particularly impactful for novice programmers, who are tempted to read the large blocks of code despite their perception as a nuisance (Prather et al., 2023). Accordingly, their attention is shifted from thinking and problem-solving to deciphering. Ironically, the feature that should accelerate productivity significantly increases participants' cognitive load due to the associated task-switching.

Programmers, particularly experienced ones, eventually learned to dismiss long, multi-line suggestions (Barke et al., 2023; Sarkar et al., 2022). Nevertheless, even when ultimately rejecting these, their thought processes were already disrupted. This was the case not only for novice programmers who reported *"[wasting] time reading instead of thinking"* (Prather et al., 2023), but also for experienced programmers: *"I was about to write the code, and I knew what I wanted to write. But now I'm sitting here, seeing if somehow Copilot came up with something better than the person who's been writing Haskell for five years…"* (Barke et al., 2023). Similarly, in the writing domain, some users learned to ignore suggestions in certain contexts, whereas others deliberately sped up their writing to avoid getting distracted by a suggestion (Bhat et al., 2023).

That complex code suggestions are the most distracting during "acceleration" and are more helpful during "exploration" (Barke et al., 2023) suggests that their timing is a key factor. Indeed, automation research speaks to this. People respond faster to interrupting tasks if the interruption was scheduled as a breakpoint between main task chunks (Iqbal & Bailey, 2008) or when they occur at subtask boundaries (Bailey & Iqbal, 2008; Iqbal & Bailey, 2005; Janssen & Brumby, 2010). Similarly Cutrell et al. (2000), found that users interrupted earlier in a task were more likely to request a reminder after being interrupted, and Cutrell and Guan (2007) showed that the later in the main task an interruption occurs, the less recovery time is needed when subsequently returning to it. Indeed, in the data science domain Gu et al. (2023), found that when AI suggestions were out of sync with users' current analysis plans, participants were either distracted or ignored them.

3.4. Task-complexity polarization

Automation often makes easy tasks easier but fails to reduce the workload during cognitively demanding tasks, and in fact, often makes them harder (Lee & Seppelt, 2009). This has been termed "clumsy automation" in Human Factors research (Cook et al., 1991), but we introduce the more precise term *task-complexity polarization* (Figure 1d). One explanation is that easy tasks are easier to automate, and so the more difficult tasks tend to remain under manual control, albeit alongside the additional task of monitoring automation, and within a now more fragmented workflow (Lee & Seppelt, 2009). For example, automation has been shown to reduce pilots' mental workload when it is already low during easy tasks, as when the plane is on autopilot during a straight flight. However, automation increased the mental workload of pilots when the flight-related workload was already high, e.g., during landing, as they then had to simultaneously reprogram the system managing autopilot, activate landing procedures, and manage communication (Wiener & Curry, 1980). Humans are also ineffective in shifting cognitive resources saved by automation to support more difficult tasks. In the study by Metzger and Parasuraman (2005), air traffic controllers used automation designed to aid conflict detection and resolution tasks. This was expected to free up enough mental resources that controllers could allocate to performing more complex tasks. However, automation did not reduce the mental workload in routine tasks that were demanding, such as communication and accepting and handing off aircraft. Either the aid did not free enough resources, or the controllers could not allocate them to improve communication performance. Studies on automated decision-making used to support government tasks showed that the new technology often only reduced the easy assignments but left the difficult ones to the government workers, making their work more difficult and fragmented (Lindgren, 2023).

GenAI studies show that a similar pattern is emerging in current users of GenAI systems. First, there is evidence that GenAI systems are most helpful at making easy tasks even easier. For example, GenAI has been shown to be the most effective in supporting novice writers performing easy assignments and low-skilled customer service agents in entry-level tasks (Frey & Osborne, 2023). In AI-assisted programming, users across studies were most confident in using GenAI for simpler tasks, such as "writing boilerplate, repetitive code" (Barke et al., 2023), "short chunks of code" (Ross et al., 2023), or "coding in narrow contexts" (Sarkar et al., 2022). Barke et al. (2023) found that the most successful Copilot users were able to decompose the coding task into "microtasks," which Copilot was effective at completing (see also Vaithilingam et al. (2022)). However, it is precisely the task decomposition process itself that is the more cognitively demanding task, and for which Copilot was not able to provide support. Indeed, Copilot's limitations with larger coding problems meant that "[it] led to more task failures in medium and hard tasks" (Vaithilingam et al., 2022) (see also Ross et al. (2023); Sarkar et al. (2022)). In the data science domain, some users similarly reported feeling most confident in relying on GenAI for "peripheral tasks such as error-checking or report generation, rather than the central analysis process" (Gu et al., 2023). Likewise, in a study of AI-assisted legal analysis using GPT-4, Choi and Schwarcz (Choi & Schwarcz, 2023) conclude that" AI helps with simple legal analysis but stumbles over complex legal reasoning." Thus, whereas GenAI succeeds at making easy tasks even easier, current systems are less effective at supporting harder tasks.

There is also evidence that GenAI can make hard tasks even harder. First, as discussed throughout, GenAI systems can shift users' roles to one of cognitively demanding output evaluation (Section 3.1), restructure workflows in unhelpful ways (Section 3.2), and interrupt workflows (Section 3.3), all of which can interfere with users as they work on demanding tasks, for example by depriving them of relevant context or disrupting their task sequence. This can be particularly disruptive for novices, as one participant noted about long code suggestions, *"if you do not know what you're doing, it can confuse you more"* (Prather et al., 2023).

Secondly, GenAI systems can introduce errors into outputs that users must deal with. In AI-assisted coding, GenAI systems can "introduce subtle, difficult-to-detect bugs, which are not the kind that would be introduced by a human programmer writing code manually" (Sarkar et al., 2022). Errors are particularly likely in longer code suggestions (Barke et al., 2023; Sarkar et al., 2022), precisely the ones that might help users address complex tasks. This makes the already demanding task of debugging even more difficult, not only because of the inherent challenge of debugging "foreign" code (as discussed in Section 3.1), but also because of errors' subtlety and the difficulty in discerning whether an error is the user's or the system's fault (Barke et al., 2023; Sarkar et al., 2022; Vaithilingam et al., 2022). A similar concern about GenAI systems introducing errors has been raised in the data science domain (Gu et al., 2023).

Thirdly, when users are stuck on a demanding task, although GenAI systems can provide multiple suggestions to help, this ends up overwhelming some users. Weisz et al. (2022) found that users' frustration and mental demand were significantly heightened when multiple AI-generated code translations were shown to participants. Users similarly found the multi-suggestion pane in Copilot to be overwhelming when they accessed it during a state of coding "exploration" (i.e., starting a novel task or stuck on a task Barke et al. (2023)). Thus, ironically, GenAI systems can make hard tasks even harder in various ways that may ultimately leave users with the same or increased cognitive workload.

4. Human factors solutions

Beyond diagnosing the usability challenges of automation, Human Factors research has spent decades studying approaches to mitigate these challenges (e.g., Endsley, 2017; Parasuraman et al., 1997; 2000; Sheridan & Parasuraman, 2005). Here, we outline some key potential design solutions that could reduce the productivity loss in human-GenAI interaction. These include providing continuous feedback to users and enabling explainability (Section 4.1), enabling system personalization (Section 4.2), applying ecological interface design (Section 4.3), using task stabilization and interruption timing techniques (Section 4.4), and enabling clear task allocation between users and systems (Section 4.5). Besides targeting individual productivity loss challenges, these solutions share the underlying Human Factors principles of providing feedback and enabling system flexibility (Carayon & Hoonakker, 2019).

More broadly, we argue that these proposed approaches aim to (i) increase user agency in how they adapt the GenAI support to users' preferred ways of working, reducing the cognitive load stemming from disrupted workflows; (ii) increase users' situational awareness of system changes and potential errors, reducing the cognitive load associated with the monitoring and evaluation of AI outputs; and (iii) increase user flexibility through the more granular application of AI support to their tasks, freeing users from having to make a binary decision of either using GenAI tools potentially ineffectively or not using them at all (Chen et al., 2023; Sarkar et al., 2022). Throughout, we focus on the programming domain as an example of how these approaches can be applied to GenAI systems. Critically, although these approaches are grounded in Human Factors research, their use in GenAI systems and the wider range of relevant application domains (e.g., many types of knowledge and creative work) needs further testing and evaluation. Ongoing research is necessary to validate the effectiveness of the proposed solutions in this context.

4.1. Continuous feedback and explainability

When GenAI is introduced to users' workflows, their role can shift from active involvement in performing the task (i.e., production) to more passively reviewing the AI-generated outputs for errors (i.e., output evaluation). The latter is a cognitively demanding task due to the lack of supporting contextual information and the resultant loss of situational awareness. We propose that feedback about system behavior is a key strategy to keep users engaged and in the loop of GenAI system performance.

During the monitoring stages, receiving continuous feedback is crucial for the operator to remain in the loop and recognize moments when interruption and input are needed (Lee & Seppelt, 2009; Loft et al., 2007). Feedback is essential to help operators know if their requests have been received if the actions of the automation system are being performed properly, and if any errors are occurring (Norman, 1990). With GenAI systems, this includes knowing which aspects of the input are serving as prompts, how they are being interpreted by the system, how the output matches them, and whether there are any errors. Thus, feedback is tied to carefully designed explainability features (Liao & Vaughan, 2023; Sun et al., 2022). It should help users know why the system is responding in a certain way and allow them to build mental models of the system's behaviour, how it interacts with them, and where they can expect failures (i.e., cause-and-effect relationships). Other explainability features such as knowing the confidence of the information provided, seeing alternate solutions and usage examples, and having access to familiar information sources can help to support users' ability to notice system mistakes and reduce the cognitive burden of monitoring (Liao & Vaughan, 2023; Sun et al., 2022; Tankelevitch et al., 2024). Figure 2 illustrates example approaches to providing feedback and increasing explainability. Notably, by helping users develop a more accurate mental model of the system, feedback and explainability features can also serve to support users in better prompting and output adaptation (Chen et al., 2023; Liao & Vaughan, 2023), thereby helping them structure their workflows more effectively (i.e., also addressing the challenge of unhelpful workflow restructuring as per Section 3.2).

We suggest that GenAI tools should continuously provide relevant feedback to users, updating them on the system's state, particularly during the monitoring stages. Feedback should be informative but non-intrusive, where the amount and form of feedback adapts to the interactive style of the participants and the nature of the problem (Norman, 1990).

In the context of GenAI systems, feedback is important for understanding system inputs and outputs and the cause-and-effect relationship between them. In the case of automated suggestions, users expressed a need for more information on specifically which code and comments Copilot relies on as inputs (Barke et al., 2023) and explanations on why certain code and documentation suggestions were made (Chen & Zacharias, 2024). In the case of conversational interfaces,

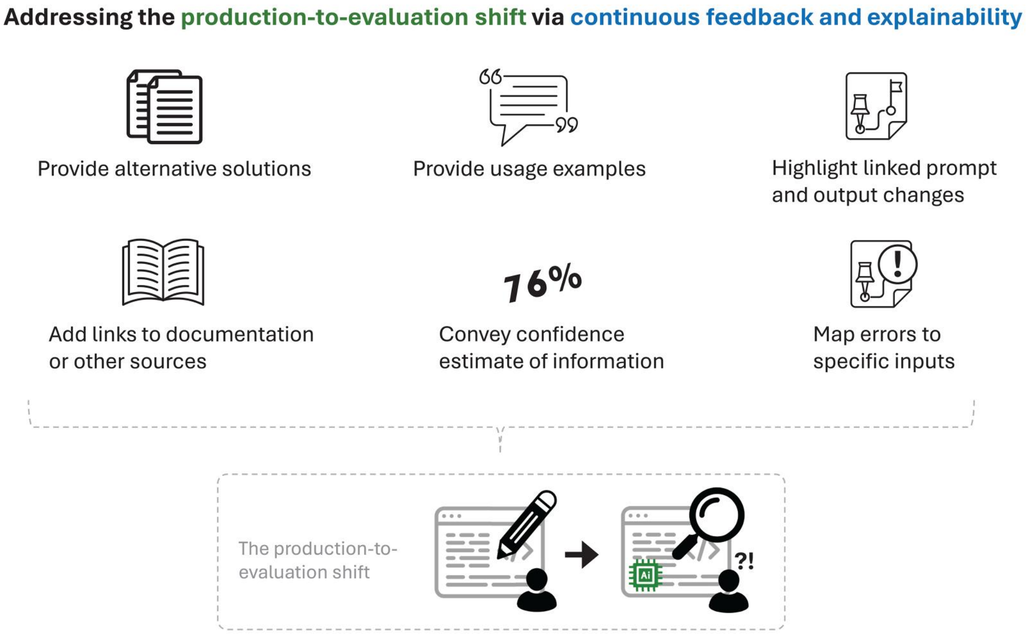
Addressing the production-to-evaluation shift via continuous feedback and explainability
Provide alternative solutions
Provide usage examples
Highlight linked prompt and output changes
Add links to documentation or other sources
76%
Convey confidence estimate of information
Map errors to specific inputs
The production-to-evaluation shift
AI
?!

Figure 2. Examples of approaches to providing continuous feedback and increasing explainability which can help address the challenges of the production-to-evaluation shift (Section 3.1). Examples include providing alternative solutions, adding links to relevant documentation or other information sources, providing relevant usage examples, conveying information based on confidence, highlighting prompt changes and resulting output changes, and providing information linking errors to inputs.

feedback could highlight prompt changes and the resulting output changes (Zamfirescu-Pereira et al., 2023), use explainability features indicating factors that influenced the output (Bauer et al., 2023), or use an analytic dashboard that shows statistics of AI tools' performance changes (Wang et al., 2024). Feedback could also be used to support pattern-matching between the AI suggestions and users' task goals. For example, the output could have keywords highlighted, such as function calls or variable names, that would be a meaningful indication of a code fit (Barke et al., 2023). It could also include more context and documentation with the output, e.g., links to Stack Overflow or official documentation pages (Xu et al., 2022), real-time information about libraries (Feuerriegel et al., 2024) or provide relevant usage examples (Moreno et al., 2015). To understand outputs, Vaithilingam et al. (2022) suggested using inline comments or highlighting different parts of the code based on confidence to help users understand the code generated by Copilot (see also Vasconcelos et al. (2023); Weisz et al. (2021)). The authors also suggested supporting debugging by automatically generating test cases and test data for users to validate and identify corner cases (Vaithilingam et al., 2022). Weisz et al. (2022) proposed using alternate translations, where the system showed users the alternative it had considered to help them identify errors. In the writing domain (Yuan et al., 2022), proposed that systems should give prompt suggestions to users. Chen and Zacharias (2024) suggested highlighting outputs with low certainty, indicating areas that should be inspected.

Feedback can be overwhelming if it is poorly presented or excessive. It can also be incomprehensible without proper context, abstraction, and integration (Lee & Seppelt, 2009). As such, feedback should be provided by applying methods of ecological interface design (Rasmussen & Vicente, 1989) (see Section 4.3) and notification design (Paul et al., 2015) (see Section 4.4), which are effective approaches for improving situational awareness and error detection.

4.2. System personalization

Human Factors studies have shown that when system personalization is constrained, the cognitive demands on operators and the associated productivity loss both increase (Cook & Woods, 1997). Indeed, as described in Section 3.2, increased cognitive demand and productivity loss have been observed in studies of GenAI-assisted programming as users try to understand and accommodate systems by changing their ways of working. This could be mitigated by allowing users to flexibly personalize systems to fit their tasks and ways of working (Lee & Seppelt, 2009).

Users should be able to personalize the inputs to the system. For example Barke et al. (2023), proposed that users should be able to control the context they provide to Copilot, enable comments that make code invisible to the tool, or decide that the tool will rely on Stack Overflow-style prompts rather than in-context code. Similarly, users should be able to choose the type of assistance to receive (or the output format of that assistance) to support their preferred ways of working. For example, in creative writing, choosing to receive feedback from GenAI, rather than a chunk of AI-generated text, preserved writers' creativity and alleviated anchoring effects and over-reliance (Chen & Chan, 2023). Users could also personalize the system to provide help when needed rather than having suggestions generated automatically (Wang et al., 2024) (see also Section 4.4.3 for more on timing). An important aspect may be the ability to inform the system about users' state of work (e.g., "acceleration" or "exploration," as per Barke et al. (2023)), such that suggestions better match the users' task characteristics and goals in terms of complexity, variety, length, and frequency (Gu et al., 2023). Systems could automatically detect users' states (Barke et al., 2023; Gu et al., 2023), guided by user-adjustable parameters, and respond according to user-provided preferences, feedback, or through the use of prompts (Tankelevitch et al., 2024).

In summary, GenAI system personalization could include allowing users to choose what to input into the system, the type of GenAI assistance to receive (and its output format), and how the system behaves in relation to users' fluctuating states of work, including the timing of assistance (Figure 3). Personalization is particularly important as users might have varying levels of task and domain expertise, which has been shown to affect their preferences and needs regarding the amount and kind of information provided (Paris, 1988). For example, novice programmers might want to spend some time working on the problem themselves and only ask Copilot for support when they are stuck (Prather et al., 2023), whereas experts might want to simply complete their lines (Barke et al., 2023). Notably, personalization is a broad concept and aspects of it relate to solutions targeting interruption timing (Section 4.4) and task-complexity polarization (Section 4.5).

4.3. Ecological interface design

The introduction of GenAI to users' workflows can disrupt them, leaving workers looking to adjust their ways of working or their familiar task structure. These processes increase cognitive load and result in productivity losses. Moreover, these disruptive changes can prevent users from being able to exercise their expertise and from benefiting from AI support. To align GenAI systems with users' workflows effectively, we suggest that GenAI systems be designed according to an ecological interface design (EID) approach. EID emphasizes designing interfaces that reflect users' perceptual constraints within a work environment in a highly domain- and context-specific manner (Rasmussen & Vicente, 1989). Specifically, it emphasizes (i) combining what users control and what they see in the system so that they can interact using clear, real-time signals; (ii) providing a consistent mapping between work domain constraints and interface cues; and (iii) showing the system's key relationships directly on the screen, making it easier for users to form a mental model of the system (McIlroy & Stanton, 2015; Rasmussen & Vicente, 1989). EID has been shown to reduce workload and improve performance in aviation risk management (King et al., 2022), medical domains (Effken et al., 1997), and automation-assisted driving (Stoner et al., 2003).

In practice, this approach suggests that an automation aid or AI system should be designed to perform consistently with operators' mental models, preferences, and expectations in a given work domain (Goodrich & Olsen, 2003). For example, GenAI systems should consider a broader domain context for their inputs by including information from interactions with external sources within the work domain (e.g., with Copilot, the consideration of code beyond the current file Bird et al. (2023)). Which sources and when they are considered should be clearly specified to users to support real-time control.

Systems should also consider work domain constraints. For example, Copilot should consider the natural task sequence of certain programming tasks by providing support

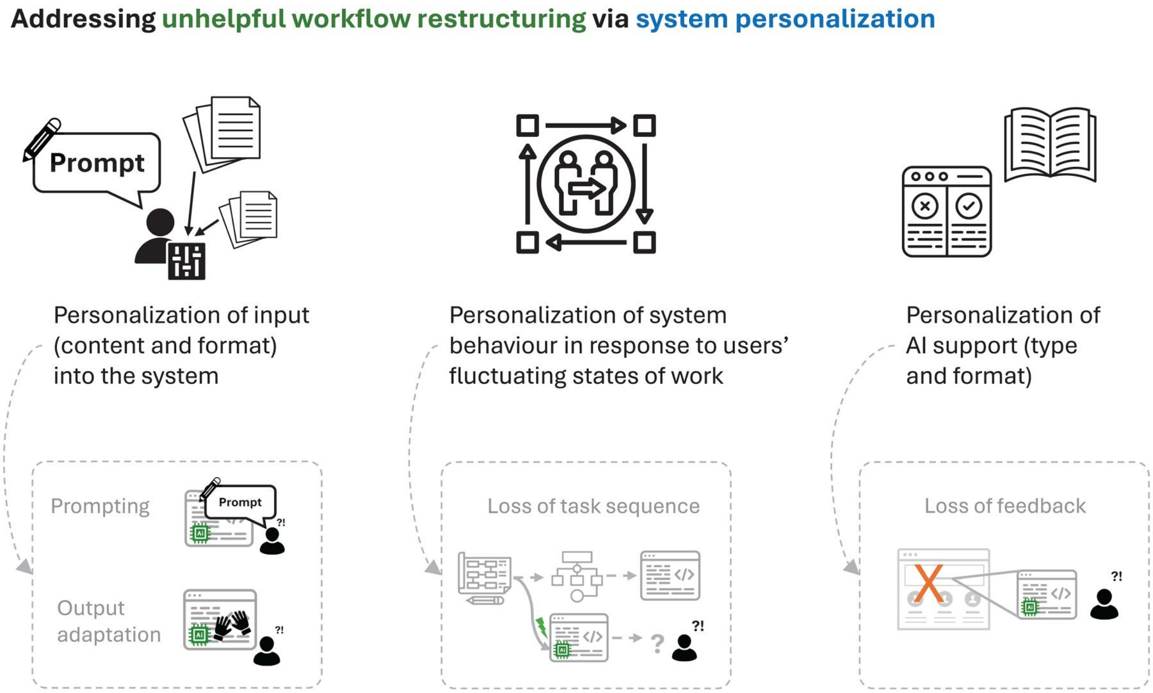
Addressing unhelpful workflow restructuring via system personalization
Prompt
Personalization of input (content and format) into the system
Personalization of system behaviour in response to users' fluctuating states of work
Personalization of AI support (type and format)
Prompting
Prompt
?!
Output adaptation
?!
Loss of task sequence
?
?!
Loss of feedback
?!

Figure 3. Examples of system personalization that can address the challenges of unhelpful workflow restructuring. This includes (i) providing personalized control over the input into GenAI systems, which can support prompting and output adaptation; (ii) personalization of system behaviour (e.g., format and frequency of outputs) in response to user's changing states of work, which helps prevent a loss of task sequence; (iii) personalization of the type of AI support (e.g., providing feedback on user's work instead of generating novel outputs), which can help mitigate the loss of feedback.

for high-level architectural design (or planning) when it is needed and avoiding code suggestions that might interfere with this process (Gu et al., 2023) (see also Section 4.4 for more on managing interruptions). Likewise, interfaces should adapt to support debugging when long code suggestions are provided, as outlined in Section 4.1. Systems should also help users understand how code suggestions map to and affect other aspects of the code beyond the local insertion point. Likewise, when helping physicians with administrative tasks, GenAI system outputs should include records of patients' unique medical histories and physicians' clinical reasoning (Preiksaitis et al., 2023).

EID also aims to support users' ways of perceiving information in a specific domain. For example, it encourages using a hierarchical visual structure to display relevant information to allow multiple levels of information to be (meaningfully) visible simultaneously in the interface. This way, users can guide their attention to the level of interest, depending on their level of expertise and current task demands (Rasmussen & Vicente, 1989). This also supports flexibility, as users do not have to attend to a specific description level. For example, depending on where users are in their workflow, GenAI systems can provide programmers with suggestions at different levels of abstraction (Gu et al., 2023), from high-level pseudo-code to low-level implementations, organized in a visual hierarchy, which would be particularly helpful for novices (Prather et al., 2023). Similarly Gu et al. (2023), suggest that interactive visualizations, linked to users' code and other parts of the interface, can be used to support decision-making. Also in line with EID principles, Feuerriegel et al. (2024) argued that GenAI tools should be embedded in workflows via user interfaces that are tailored to users' domain-specific needs and challenges.

Finally, as discussed in Section 4.1, explainability features are essential to help users form an accurate mental model of AI systems. These features should be integrated directly into the interface (e.g., as in AI Chains Wu et al. (2022)), taking into consideration the work domain context and domain expert knowledge (DeGrave et al., 2023; Pasquale & Malgieri, 2023). In the healthcare domain, explainability has been shown to be most effective when combined with insights from medical experts. Without considering domain specifics, explanations (e.g., in dermatology) lacked important context and included unnecessary information (e.g., background skin texture) that confused expert dermatologists DeGrave et al. (2023) (see also Huang et al., (2023) for similar results in radiology). Indeed, a review of research on decision-support system implementation in medicine shows that misaligned user interfaces and explainability of AI outputs are key issues for successful clinician-AI interaction at work (Zając et al., 2023). For example, clinicians were ineffective working with decision-support systems when explanatory information was incomprehensible, or if interfaces presented multiple risk scores simultaneously (Zając et al., 2023). In summary, EID features could be used in the context of GenAI by considering interface design that (i) supports real-time control by mapping user input and wider context into feedback provided by the interface, (ii) supports users' specific ways of perceiving information in a given domain and task, (iii) uses domain-specific explainability features (Figure 4).

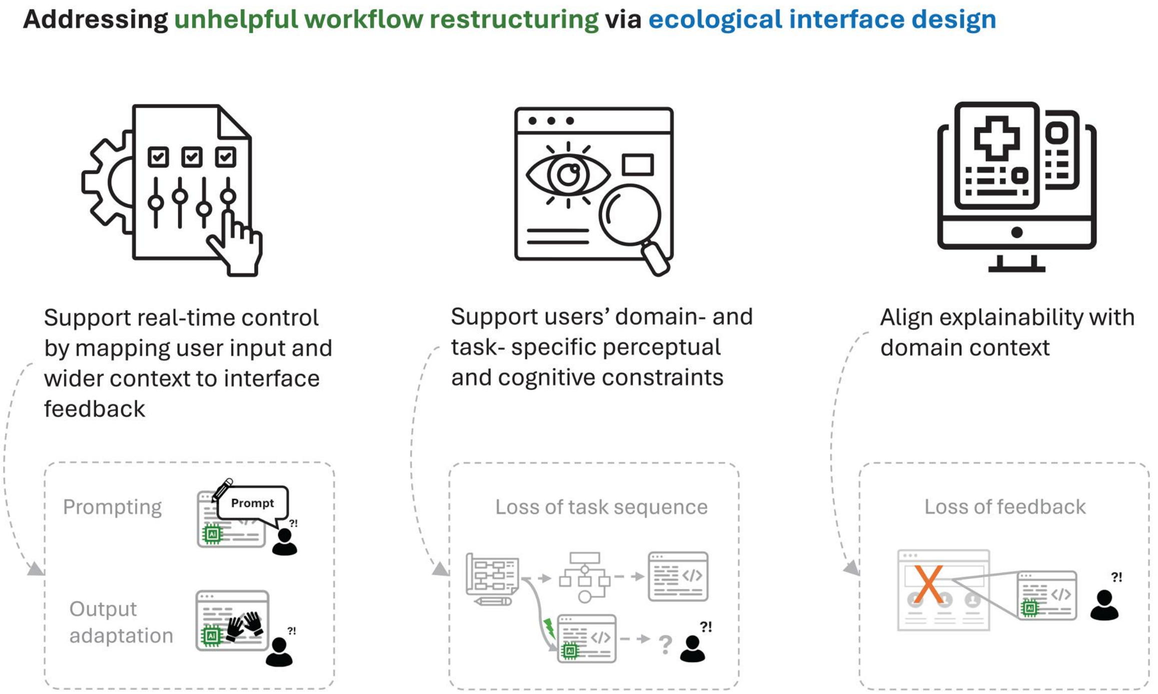
Addressing unhelpful workflow restructuring via ecological interface design
Support real-time control by mapping user input and wider context to interface feedback
Support users' domain- and task- specific perceptual and cognitive constraints
Align explainability with domain context
Prompting
Prompt
?!
Output adaptation
?!
Loss of task sequence
?
?!
Loss of feedback
?!

Figure 4. Examples of how ecological interface design can help address the challenges of unhelpful workflow restructuring. This includes (i) supporting real-time control by mapping user input and wider context into feedback provided by the interface, which can support prompting and output adaptation, (ii) supporting users' domain- and task- specific perceptual and cognitive constraints, which can help mitigate the loss of task sequence, and (iii) aligning explainability with the domain context, which can help mitigate the loss of feedback.

4.4. Main task stabilization and interruption timing

As discussed in Section 3.3, GenAI system suggestions (e.g., Copilot code suggestions) interrupt users, especially during their flow states, distracting them and potentially leading to productivity loss. Accordingly, some users disable auto-suggestion features or GenAI systems entirely because of their distracting nature (Barke et al., 2023; Liang et al., 2023; Sarkar et al., 2022). Writers similarly prefer not to be interrupted by AI-generated snippets of text (Chen & Chan, 2023). Instead of forcing users to avoid interruptions by disabling tools, systems should preserve users' flow states by incorporating task stabilization techniques or by carefully timing interruptions around their flow states. We discuss three solutions to mitigate the negative impact of task interruptions: (i) task stabilization with attention guidance, (ii) task stabilization via pre-interruption alerts, and (iii) timing of interruptions (Figure 5).

4.4.1. Task stabilization via attention guidance

Interruptions can be designed to support task stabilization, i.e., to help users prepare their current (main) task for the temporary switch in focus (Czerwinski et al., 2004; Parnin & DeLine, 2010). For example, among software users and developers (Paul et al., 2015), found that interruptions were helpful when they directed users to the parts of the current task (or a new task) they needed to attend to. Interruptive notifications were also useful as progress indicators, helping users plan and resume their next task after interruption. In the case of GenAI systems such as Copilot, this could manifest in long code suggestions being divided (e.g., via colour) into small logical units for programmers to easily parse during the acceleration (flow) mode (Barke et al., 2023). Alternatively, systems could direct users' attention to certain keywords (e.g., via highlighting) that could help them identify the applicability of the suggestion by using "pattern matching" (Barke et al., 2023). In line with Human Factors principles, interface design should provide cues to guide users' attention to the next appropriate action. Otherwise, users may fall into "procedural traps" (Rasmussen & Vicente, 1989; Reason, 1990), novel situations where they rely on their normal rule set but without the usual success. Indeed, this has been observed in Copilot studies, where programmers end up in "debugging rabbitholes" (Prather et al., 2023; Vaithilingam et al., 2022).

4.4.2. Task stabilization via pre-interruption alerts

Task stabilization can also be achieved by using pre-interruption alerts, which function as progress indicators, helping users plan and resume their next task after interruption (Paul et al., 2015). Andrews et al. (2003) found that those who received a pre-interruption alert could resume the main task faster than participants who did not. This aligns with studies showing that adding a brief lag period before interruption helps users set place-keepers at their current task point, making it easier for them to return to it after being interrupted (Altmann & Gregory Trafton, 2015; Brumby et al., 2013). Similar pre-interruption alerts may be helpful for GenAI systems. For example, when Copilot is about to suggest a long code chunk, an alerting notification could create a brief pause period necessary to lock the users' main task state. Even better, AI systems should set place-keepers automatically together with auto-suggestions, along with any other context-relevant information that could help users return to their train of thought. This would begin to address the challenge of helping users regain their prior context post-interruption, as has been raised in GenAI-assisted coding (Ross et al., 2023) and data science (Gu et al., 2023; 2023).

4.4.3. Timing of interruptions

Timing interruptions thoughtfully is another way to reduce their associated productivity loss. Interruptions are valuable for user productivity when they provide valuable awareness about things outside the user's attention, such as new or background tasks (Paul et al., 2015). However, interruptions can be disruptive when related to a task currently in focus.

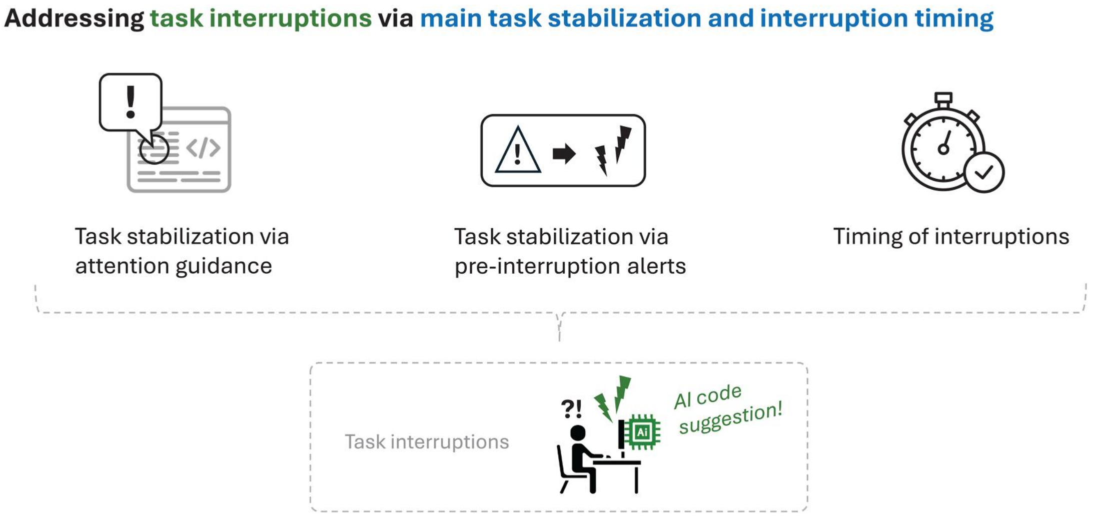
Addressing task interruptions via main task stabilization and interruption timing
Task stabilization via attention guidance
Task stabilization via pre-interruption alerts
Timing of interruptions
Task interruptions
?!
AI code suggestion!

Figure 5. Three types of approaches to mitigate the negative impacts of task interruptions: (i) main task stabilization via attention guidance, (ii) main task stabilization via pre-interruption alerts, and (iii) interruption timing.

We propose that systems such as Copilot should be able to recognize when the user is in focus (Barke et al., 2023; Gu et al., 2023). Then, interruptions should be limited to supporting contextual alerts or providing information about ongoing tasks in the background (e.g., providing explainability information). Otherwise, during this stage, suggestions should carefully align with users' flow (Gu et al., 2023), in line with ecological interface design. The system should recognize the strategies that users use during the flow state and support them by completing their thought processes, for example, auto-completing the end of the code line (Barke et al., 2023), providing only short code suggestions (Prather et al., 2023). Recognizing when users are not in a flow state, systems could give users prompt examples and suggestions (Yuan et al., 2022), provide feedback (Chen & Chan, 2023), or goal-orientated guidance (Arnold et al., 2021). This could be supported further by user personalization as per Section 4.2. This would enable GenAI support to be used more narrowly (e.g., to provide warning messages and supporting contextual information or short snippets of code) rather than users having to use the GenAI ineffectively or turn it off completely.

4.5. Clear task allocation

GenAI user studies suggest that current systems make easy tasks easier and hard tasks harder for users, a phenomenon we have termed task-complexity polarization (and referred to as "clumsy automation" in the Human Factors literature Wiener and Curry (1980)). Thus, it appears that these systems are not applied effectively to reduce overall workload. Human Factors research shows that one of the ways to address this is by clearly specifying how tasks are allocated between the human and system, particularly during high workload periods (Enstrom & Rouse, 1977; Sinaiko, 1972). This not only better distributes the workload according to the respective strengths and weaknesses of humans and automated systems but also reduces the cognitive demand on users stemming from trying to discern the relative responsibilities on a moment-by-moment basis. For example, in aviation, reducing pilots' workflow to a single loop (eliminating the need for the operator to interact with the automation through the high workload tasks) resulted in better performance in a cockpit simulator. Similarly, allocating tasks to the computer and allowing the operator to deal with the queue items manually have also been shown to reduce workload (Chu & Rouse, 1979). We suggest that the allocation of tasks between the user and GenAI system should be clearly defined and supported by GenAI systems. The user should know which tasks the GenAI system deals with at a given moment (Cook & Woods, 1997). Moreover, allocating tasks to automation systems and allowing the operator to deal with the queue items manually has also been shown to reduce cognitive workload (Chu & Rouse, 1979). Thus, users should also be able to allocate tasks to the GenAI system and themselves (Figure 6).

As discussed in Sections 3.1 and 3.4, for simple tasks or in low workload conditions, users often let the GenAI system operate continuously. However, when complex tasks needed to be performed, they often stepped in and overrode the system and, in some cases, engaged in ineffective practices (e.g., reviewing code suggestions, editing, and then deleting them Prather et al. (2023)). Instead of having to do this, users should be able to proactively allocate responsibilities to the GenAI system. For example, according to their experience with the system, personal preferences, or expertise, they could identify tasks or parts of the tasks that they are confident that AI will perform successfully without their oversight or ones that they found AI to be most helpful with. For example, users might prefer manually translating certain types of code (Weisz et al., 2022), allowing the tool to be responsible for generating control structures while the user fills out the body (Barke et al., 2023), or using prompt

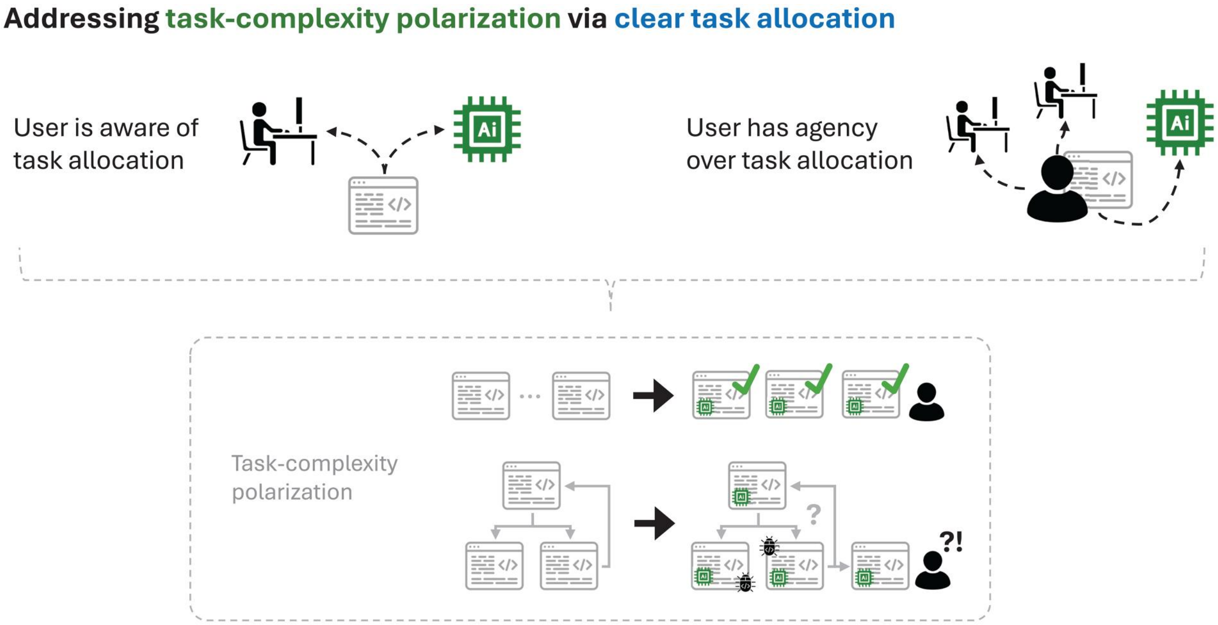
Addressing task-complexity polarization via clear task allocation
User is aware of task allocation
Ai
User has agency over task allocation
Ai
Task-complexity polarization
?
?!

Figure 6. User awareness of task allocation and agency over task allocation can help address the challenge of task-complexity polarization.

engineering strategies to assign a certain role (e.g., critical code reviewer) to the GenAI tool (Ulfsnes et al., 2024). Likewise, users could allocate only repetitive "boilerplate" code for the system to complete autonomously while requesting its high-level planning support (rather than entire code completion) during more complex or exploratory tasks. In creative domains, this might mean that GenAI tools provide ideas in an open-ended form (e.g., probing questions), rather than as explicit suggestions (Arnold et al., 2021), an approach that was found to be particularly helpful in copywriting (Chen & Chan, 2023). Making this initial allocation of responsibility and clearly understanding how tasks are divided would reduce the cognitive load of interacting with the GenAI system throughout demanding tasks. Moreover, it would help users better manage their demanding role as evaluators of AI output (as per Section 3.1).

Supporting effective task allocation depends on GenAI systems having a clear understanding of the work domain context, which is enabled by ecological interface design (see Section 4.3). As such, the described Human Factors approaches work in synergy to support human-GenAI interaction and productivity.

5 Conclusions

We have synthesized and analyzed the productivity challenges emerging during human-GenAI interactions, focusing on the much-studied domain of software development and noting similarities in areas such as data science, design, and writing. We have demonstrated the parallels between productivity challenges in older Human Factors automation studies and recent GenAI studies. Drawing on the human automation studies, we have categorized these challenges and the underlying reasons related to Human Factors, such as workload, feedback, and situational awareness. We show how aspects like the shift from active production (e.g., writing code) to passive evaluation (e.g., reviewing code), unhelpful workflow restructuring, task interruptions, and task-complexity polarization can stifle human performance and effective implementation of GenAI.

Further extrapolating from human-automation studies, we have provided a set of design solutions that could help avoid productivity losses in human-GenAI interaction. More broadly, we argue for more consideration of users' workflows, unique ways of working, and domain specificities when designing GenAI tools. To achieve this, we propose that systems be designed in accordance with ecological interface design, the principle of continuous feedback, support for flexibility via task allocation between users and systems, and user-guided system personalization. We also provide concrete design solutions for effectively guiding user attention during interruptions. Human Factors research provides a fertile starting ground to explore these solutions for GenAI. However, because the technology is still novel and is being applied in a wider set of domains than previous forms of automation, we suggest that future research is critical to test the validity and effectiveness of these solutions in the GenAI context.

Our paper is an initial bridge between Human Factors and Human-Computer Interaction issues of human-GenAI interaction. There is, of course, far more nuanced Human Factors research that can help understand and address the key productivity challenges in this fast-paced area. Reciprocally, we also expect that future Human-Computer Interaction research may open up new domains of exploration for Human Factors.

Notes

1. Endsley (2023) makes a similar parallel between the ironies of automation and the challenges of modern AI systems; however, whereas they cover both generative and non-generative AI and take a high-level view of AI, the current paper focuses specifically on GenAI and examines concrete usability challenges documented in recent user studies of GenAI systems.
2. This shift from production to evaluation is relative rather than absolute, as, for example, crafting prompts still constitutes a form of production (see Section 3.2.1).
3. Situational awareness can also be reduced due to automation-related unhelpful structuring of workflows (Section 3.2), including changes in the task sequence (Section 3.2.3) and the loss of feedback (see Section 3.2.4).

Disclosure statement

Several authors are employed by Microsoft which produces Generative AI tools.

Funding

The authors report there are no funding bodies associated with this research.

ORCID

Auste Simkute http://orcid.org/0000-0002-2996-4618
Lev Tankelevitch http://orcid.org/0000-0003-1286-5194
Viktor Kewenig http://orcid.org/0009-0009-5912-0676
Ava Elizabeth Scott http://orcid.org/0000-0002-4469-4556
Abigail Sellen http://orcid.org/0000-0001-9065-3061
Sean Rintel http://orcid.org/0000-0003-0840-0546

References

Altmann, E. M., & Gregory Trafton, J. (2002). Memory for goals: An activation-based model. *Cognitive Science*, *26*(1), 39–83. https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog2601_2.

Altmann, E. M., & Gregory Trafton, J. (2015). Brief lags in interrupted sequential performance: Evaluating a model and model evaluation method. *International Journal of Human-Computer Studies*, *79*, 51–65. https://doi.org/10.1016/j.ijhcs.2014.12.007

Altmann, E. M., Gregory Trafton, J., & Hambrick, D. Z. (2014). Momentary interruptions can derail the train of thought. *Journal of Experimental Psychology. General*, *143*(1), 215–226. https://doi.org/10.1037/a0030986

Andrew, A. M. (2003). Humans and automation: System design and research issues, by Thomas B. Sheridan, Wiley, in cooperation with the Human Factors and Ergonomics Society, Santa Barbara, California, 2002, pp. xii, 264. ISBN 0-471-23428-1. Wiley Series in System Engineering and Management HFES Issues in Human Factors and Ergonomics Series, Vol. 3 (Hardback, £37.50. *Robotica*, *21*(3), 345–345. https://doi.org/10.1017/S0263574702274858

Arnold, K. C., Volzer, A. M., & Madrid, N. G. (2021). Generative models can help writers without writing for them. *Joint Proceedings of the ACM IUI 2021 Workshops* (2021).

Bailey, B. P., & Iqbal, S. T. (2008). Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interaction*, *14*(4), 1–28. https://doi.org/10.1145/1314683.1314689

Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19*(6), 775–779. https://doi.org/10.1016/0005-1098(83)90046-8

Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, *7*(OOPSLA1), 85–111. https://doi.org/10.1145/3586030

Bauer, K., von Zahn, M., & Hinz, O. (2023). Expl(AI)ned: The impact of explainable artificial intelligence on users' information processing. *Information Systems Research*, *34*(4), 1582–1602. https://doi.org/10.1287/isre.2023.1199

Baxter, G., Rooksby, J., Wang, Y., & Khajeh-Hosseini, A. (2012). The ironies of automation: still going strong at 30?. In *Proceedings of the 30th European Conference on Cognitive Ergonomics*. ACM, Edinburgh United Kingdom (pp. 65–71). https://doi.org/10.1145/2448136.2448149

Bhat, A., Agashe, S., Oberoi, P., Mohile, N., Jangir, R., & Joshi, A. (2023). Interacting with next-phrase suggestions: How suggestion systems aid and influence the cognitive processes of writing. In *Proceedings of the 28th International Conference on Intelligent User Interfaces (IUI '23)* (pp. 436–452). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3581641.3584060

Billings, C. E. (1991). Toward a human-centered aircraft automation philosophy. *The International Journal of Aviation Psychology*, *1*(4), 261–270. https://doi.org/10.1207/s15327108ijap0104_1

Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowdermilk, T., & Gazit, I. (2023). Taking Flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. *Queue*, *20*(6), 35–57. https://doi.org/10.1145/3582083

Brumby, D. P., Cox, A. L., Back, J., & Gould, S. J. J. (2013). Recovering from an interruption: Investigating speed-accuracy trade-offs in task resumption behavior. *Journal of Experimental Psychology. Applied*, *19*(2), 95–107. https://doi.org/10.1037/a0032696

Budhwar, P., Chowdhury, S., Wood, G., Aguinis, H., Bamber, G. J., Beltran, J. R., Boselie, P., Lee Cooke, F., Decker, S., DeNisi, A., Dey, P. K., Guest, D., Knoblich, A. J., Malik, A., Paauwe, J., Papagiannidis, S., Patel, C., Pereira, V., Ren, S., … Varma, A. (2023). Human resource management in the age of generative artificial intelligence: Perspectives and research directions on ChatGPT. *Human Resource Management Journal*, *33*(3), 606–659. https://doi.org/10.1111/1748-8583.12524

Calderwood, A., Qiu, V., Gero, K., Chilton, L. B. (2020). How Novelists Use Generative Language Models: An Exploratory User Study. In *IUI 2020 Workshops*. Retrieved from https://www.semanticscholar.org/paper/How-Novelists-Use-Generative-Language-Models/%3A-An-Calderwood-Qiu/8cf1fc0b87dfda2a11bfaaaa3a0bf9f9e069bb0f

Carayon, P., & Hoonakker, P. (2019). Human factors and usability for health information technology: Old and new challenges. *Yearbook of Medical Informatics*, *28*(1), 71–77. https://doi.org/10.1055/s-0039-1677907

Chen, J., & Zacharias, J. (2024). Design principles for collaborative generative AI systems in software development. In M. Mandviwalla, M. Söllner, and T. Tuunanen (Eds.), *Design science research for a resilient future* (pp. 341–354). Springer Nature. https://doi.org/10.1007/978-3-031-61175-9_23

Chen, X. A., Burke, J., Du, R., Hong, M. K., Jacobs, J., Laban, P., Li, D., Peng, N., Willis, K. D. D., Wu, C.-S., & Zhou, B. (2023). Next steps for human-centered generative AI: A technical perspective. https://doi.org/10.48550/arXiv.2306.15774 arXiv:2306.15774 [cs].

Chen, Z., & Chan, J. (2023). Large language model in creative work: The role of collaboration modality and user expertise. https://doi.org/10.2139/ssrn.4575598

Chignell, M., Wang, L., Zare, A., & Li, J. (2023). The evolution of HCI and human factors: Integrating human and artificial intelligence. *ACM Transactions on Computer-Human Interaction*, *30*(2), 1–30. https://doi.org/10.1145/3557891

Choi, J. H., & Schwarcz, D. (2023). AI assistance in legal analysis: An empirical study. https://doi.org/10.2139/ssrn.4539836

Chu, Y.-Y., & Rouse, W. B. (1979). Adaptive allocation of decisionmaking responsibility between human and computer in multitask situations. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(12), 769–778. https://doi.org/10.1109/TSMC.1979.4310128

Clark, E., Ross, A. S., Tan, C., Ji, Y., & Smith, N. A. (2018). Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces (IUI '18)* (pp. 329–340). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3172944.3172983

Cook, R. I., Woods, D. D., Mccolligan, E., & Howie, M. B. (1991). Cognitive consequences of clumsy automation on high workload, high consequence human performance. In *NASA, Lyndon B. Johnson Space Center, Fourth Annual Workshop on Space Operations Applications and Research (SOAR 90)*. NTRS Author Affiliations: Ohio State Univ. NTRS Document ID: 19910011398 NTRS Research Center: Legacy CDMS (CDMS). Retrieved from https://ntrs.nasa.gov/citations/19910011398

Cook, R., & Woods, D. (1997). Adapting to new technology in the operating room. *Human Factors*, *38*(4), 593–613. https://doi.org/10.1518/001872096778827224

Cork, R. D., Detmer, W. M., & Friedman, C. P. (1998). Development and initial validation of an instrument to measure physicians' use of, knowledge about, and attitudes toward computers. *Journal of the American Medical Informatics Association: JAMIA*, *5*(2), 164–176. https://doi.org/10.1136/jamia.1998.0050164

Cutrell, E. B., Czerwinski, M., & Horvitz, E. (2000). Effects of instant messaging interruptions on computing tasks. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems* (pp. 99–100). ACM, The Hague The Netherlands. https://doi.org/10.1145/633292.633351

Cutrell, E., & Guan, Z. (2007). What are you looking for?: An eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, (pp. 407–416). https://doi.org/10.1145/1240624.1240690

Czerwinski, M., Horvitz, E., & Wilhite, S. (2004). A diary study of task switching and interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 175–182). ACM, Vienna Austria. https://doi.org/10.1145/985692.985715

Dang, H., Goller, S., Lehmann, F., & Buschek, D. (2023). Choice over control: How users write with large language models using diegetic and non-diegetic prompting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA (pp. 1–17). https://doi.org/10.1145/3544548.3580969

DeGrave, A. J., Cai, Z. R., Janizek, J. D., Daneshjou, R., & Lee, S.-I. (2023). Auditing the inference processes of medical-image classifiers by leveraging generative AI and the expertise of physicians. *Nature Biomedical Engineering*, 1–13. https://doi.org/10.1038/s41551-023-01160-9

Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. https://doi.org/10.2139/ssrn.4573321

Dixon, S. R., Wickens, C. D., & Chang, D. (2005). Mission control of multiple unmanned aerial vehicles: A workload analysis. *Human Factors*, *47*(3), 479–487. https://doi.org/10.1518/001872005774860005

Drosos, I., Barik, T., Guo, P. J., DeLine, R., & Gulwani, S. (2020). Wrex: A unified programming-by-example interaction for synthesizing readable code for data scientists. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)* (pp. 1–12). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3313831.3376442

Dr. Rudisill, M. (1995). Line Pilots' Attitudes About And Experience With Flight Deck Automation: Results Of An International Survey And Proposed Guidelines. *Proceedings of the eighth international symposium on aviation psychology*.

Effken, J. A., Kim, N.-G., & Shaw, R. E. (1997). Making the constraints visible: Testing the ecological approach to interface design. *Ergonomics*, *40*(1), 1–27. https://doi.org/10.1080/001401397188341

Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 65–84. https://doi.org/10.1518/001872095779049499

Endsley, M. R. (2017). From here to autonomy: lessons learned from human–automation research. *Human Factors*, *59*(1), 5–27. https://doi.org/10.1177/0018720816681350

Endsley, M. R. (2023). Ironies of artificial intelligence. *Ergonomics*, *66*(11), 1656–1668. https://doi.org/10.1080/00140139.2023.2243404

Endsley, M. R., & Rodgers, M. D. (2016). Distribution of Attention, Situation Awareness and Workload in a Passive Air Traffic Control Task: Implications for Operational Errors and Automation. *Air Traffic Control Quarterly*, *6*(1), 21–44. https://doi.org/10.2514/atcq.6.1.21

Endsley, M. R., Bolstad, C. A., Jones, D. G., & Riley, J. M. (2003). Situation awareness oriented design: From user's cognitive requirements to creating effective supporting technologies. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *47*(3), 268–272. https://doi.org/10.1177/154193120304700304

Endsley, M. R., Mogford, R. H., Allendoerfer, K. R., & Snyder, M. D. (1997). *Effect of free flight conditions on controller performance, workload, and situation awareness*. Technical Report. Federal Aviation Administration.

Enstrom, K. D., & Rouse, W. B. (1977). Real-time determination of how a human has allocated his attention between control and monitoring tasks. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*(3), 153–161. https://doi.org/10.1109/TSMC.1977.4309679

Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, *66*(1), 111–126. https://doi.org/10.1007/s12599-023-00834-7

Frey, C. B., & Osborne, M. (2023). Generative AI and the future of work: A reappraisal. *Brown Journal of World Affairs*, *30*(1).

Friedman, N. (2021). Introducing GitHub Copilot: your AI pair programmer. Retrieved from https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/

Funk, K., Lyall, B., Wilson, J., Vint, R., Niemczyk, M., Suroteguh, C., & Owen, G. (1999). Flight deck automation issues. *The International Journal of Aviation Psychology*, *9* (2), 109–123. https://doi.org/10.1207/s15327108ijap0902_2

Galster, S. M., Bolia, R. S., Roe, M. M., & Parasuraman, R. (2001). Effects of automated cueing on decision implementation in a visual search task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *45*(4), 321–325. https://doi.org/10.1177/154193120104500412

Gmeiner, F., Yang, H., Yao, L., Holstein, K., & Martelaro, N. (2023). Exploring challenges and opportunities to support designers in learning to co-create with AI-based manufacturing design tools. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)* (pp. 1–20). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3580999

Goodrich, M. A., & Olsen, D. R. (2003). Seven principles of efficient human robot interaction. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)* (Vol. 4, pp. 3942–3948). IEEE, Washington, DC, USA. https://doi.org/10.1109/ICSMC.2003.1244504

Grubb, P. L., Warm, J. S., Dember, W. N., & Berch, D. B. (1995). Effects of multiple-signal discrimination on vigilance performance and perceived workload. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *39*(21), 1360–1364. https://doi.org/10.1177/154193129503902101

Gu, K., Grunde-McLaughlin, M., McNutt, A. M., Heer, J., & Althoff, T. (2023). How do data analysts respond to AI assistance? A wizard-of-oz study. https://doi.org/10.48550/arXiv.2309.10108 arXiv:2309.10108 [cs].

Gu, K., Shang, R., Althoff, T., Wang, C., & Drucker, S. M. (2023). How do analysts understand and verify AI-assisted data analyses?. https://doi.org/10.48550/arXiv.2309.10947 arXiv:2309.10947 [cs].

Haldane, A. G., & May, R. M. (2011). Systemic risk in banking ecosystems. *Nature*, *469*(7330), 351–355. https://doi.org/10.1038/nature09659

Huang, J., Neill, L., Wittbrodt, M., Melnick, D., Klug, M., Thompson, M., Bailitz, J., Loftus, T., Malik, S., Phull, A., Weston, V., Heller, J. A., & Etemadi, M. (2023). Generative artificial intelligence for chest radiograph interpretation in the emergency department. *JAMA Network Open*, *6*(10), e2336100. https://doi.org/10.1001/jamanetworkopen.2023.36100

Iqbal, S. T., & Bailey, B. P. (2005). Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. Association for Computing Machinery, New York, NY, USA (pp. 1489–1492). https://doi.org/10.1145/1056808.1056948

Iqbal, S. T., & Bailey, B. P. (2008). Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)* (pp. 93–102). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1357054.1357070

Janssen, C. P., & Brumby, D. P. (2010). Strategic adaptation to performance objectives in a dual-task setting. *Cognitive Science*, *34*(8), 1548–1560. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1551-6709.2010.01124.x.

Janssen, C. P., Brumby, D. P., Dowell, J., Chater, N., & Howes, A. (2011). Identifying optimum performance trade-offs using a cognitively bounded rational analysis model of discretionary task interleaving. *Topics in Cognitive Science*, *3*(1), 123–139. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1756-8765.2010.01125.x.

Janssen, C. P., Gould, S. J. J., Li, S. Y. W., Brumby, D. P., & Cox, A. L. (2015). Integrating knowledge of multitasking and interruptions across different perspectives and research methods. *International Journal of Human-Computer Studies*, *79*, 1–5. https://doi.org/10.1016/j.ijhcs.2015.03.002

Jayagopal, D., Lubin, J., & Chasins, S. E. (2022). Exploring the learnability of program synthesizers by novice programmers. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)* (pp. 1–15). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3526113.3545659

Jiang, E., Toh, E., Molina, A., Olson, K., Kayacik, C., Donsbach, A., Cai, C. J., & Terry, M. (2022). Discovering the syntax and strategies of natural language programming with generative language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)* (pp. 1–19). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491102.3501870

Jones, D. G., & Endsley, M. R. (1996). Sources of situation awareness errors in aviation. *Aviation, Space, and Environmental Medicine*, *67*(6), 507–512.

Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment. https://doi.org/10.48550/arXiv.2309.14049 arXiv:2309.14049 [cs].

King, B. J., Read, G. J. M., & Salmon, P. M. (2022). Clear and present danger? Applying ecological interface design to develop an aviation risk management interface. *Applied Ergonomics*, *99*, 103643. https://doi.org/10.1016/j.apergo.2021.103643

Klein, G., Moon, B., & Hoffman, R. R. (2006). Making sense of sensemaking 1: Alternative perspectives. *IEEE Intelligent Systems*, *21*(4), 70–73. https://doi.org/10.1109/MIS.2006.75

Kulkarni, C., Druga, S., Chang, M., Fiannaca, A., Cai, C., & Terry, M. (2023). A word is worth a thousand pictures: Prompts as AI design material. https://doi.org/10.48550/arXiv.2303.12647 arXiv:2303.12647 [cs].

Lee, J., & Seppelt, B. (2009). Human factors in automation design. In *Springer handbook of automation* (pp. 417–436). Springer. https://doi.org/10.1007/978-3-540-78831-7_25

Liang, J. T., Yang, C., & Myers, B. A. (2023). Understanding the usability of AI programming assistants. arXiv:2303.17125 [cs]. https://doi.org/10.48550/arXiv.2303.17125

Liao, Q. V., & Vaughan, J. W. (2023). AI transparency in the age of LLMs: A human-centered research roadmap. arXiv:2306.01941 [cs]. https://doi.org/10.48550/arXiv.2306.01941

Liao, Q. V., Subramonyam, H., Wang, J., & Wortman Vaughan, J. (2023). Designerly understanding: Information needs for model transparency to support design ideation for AI-powered user experience. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)* (pp. 1–21). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3580652

Lindgren, I. (2023). Ironies of public service automation – Bainbridge revisited. In *Proceedings of the 24th Annual International Conference on Digital Government Research* (pp. 395–404). ACM, Gda?sk Poland. https://doi.org/10.1145/3598469.3598514

Loft, S., Sanderson, P., Neal, A., & Mooij, M. (2007). Modeling and predicting mental workload in en route air traffic control: Critical review and broader implications. *Human Factors*, *49*(3), 376–399. https://doi.org/10.1518/001872007X197017

Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: How may AI and GPT impact academia and libraries? *Library Hi Tech News*, *40*(3), 26–29. https://doi.org/10.1108/LHTN-01-2023-0009

Manzey, D., Bahner, J. E., & Hueper, A.-D. (2006). Misuse of automated aids in process control: Complacency, automation bias and possible training interventions. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(3), 220–224. https://doi.org/10.1177/154193120605000303

Manzey, D., Reichenbach, J., & Onnasch, L. (2012). Human performance consequences of automated decision aids: The impact of degree of automation and system experience. *Journal of Cognitive Engineering and Decision Making*, *6*(1), 57–87. https://doi.org/10.1177/1555343411433844

Mark, G., Gudith, D., & Klocke, U. (2008). The cost of interrupted work: more speed and stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 107–110). ACM, Florence Italy. https://doi.org/10.1145/1357054.1357072

Mark, G., Voida, S., & Cardello, A. (2012). "A pace not dictated by electrons": an empirical study of work without email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 555–564). ACM, Austin, Texas, USA. https://doi.org/10.1145/2207676.2207754

McBride, S. E., Rogers, W. A., & Fisk, A. D. (2011). Understanding the effect of workload on automation use for younger and older adults. *Human Factors*, *53*(6), 672–686. https://doi.org/10.1177/0018720811421909

McIlroy, R. C., & Stanton, N. A. (2015). Ecological interface design two decades on: Whatever happened to the SRK taxonomy? *IEEE Transactions on Human-Machine Systems*, *45*(2), 145–163. https://doi.org/10.1109/THMS.2014.2369372

Mcnutt, A. M., Wang, C., Deline, R. A., & Drucker, S. M. (2023). On the design of AI-powered code assistants for notebooks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)* (pp. 1–16). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3580940

Metzger, U., & Parasuraman, R. (2001). The role of the air traffic controller in future air traffic management: An empirical study of active control versus passive monitoring. *Human Factors*, *43*(4), 519–528. https://doi.org/10.1518/001872001775870421

Metzger, U., & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human Factors*, *47*(1), 35–49. https://doi.org/10.1518/0018720053653802

Monk, C. A., Gregory Trafton, J., & Boehm-Davis, D. A. (2008). The effect of interruption duration and demand on resuming suspended goals. *Journal of Experimental Psychology. Applied*, *14*(4), 299–313. https://doi.org/10.1037/a0014402

Moray, N., Lootsteen, P., & Pajak, J. (1986). Acquisition of process control skills. *IEEE Transactions on Systems, Man, and Cybernetics*, *16*(4), 497–504. https://doi.org/10.1109/TSMC.1986.289252

Moreno, L., Bavota, G., Di Penta, M., Oliveto, R., & Marcus, A. (2015). How Can I Use This Method?. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. (pp. 880–890). ISSN: 1558-1225. https://doi.org/10.1109/ICSE.2015.98

Norman, D. A. (1990). The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *327*(1241), 585–593. https://doi.org/10.1098/rstb.1990.0101

Nova, K. (2023). Generative AI in healthcare: Advancements in electronic health records, facilitating medical languages, and personalized patient care. *Journal of Advanced Analytics in Healthcare Management*, *7*(1), 115–131.

Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. https://doi.org/10.2139/ssrn.4375283

Oppenlaender, J. (2022). The creativity of text-to-image generation. In *Proceedings of the 25th International Academic Mindtrek Conference* (pp. 192–202). arXiv:2206.02904 [cs]. https://doi.org/10.1145/3569219.3569352

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *39*(2), 230–253. https://doi.org/10.1518/001872097778543886

Parasuraman, R., Molloy, R., & Singh, I. L. (1993). Performance consequences of automation-induced 'complacency. *The International Journal of Aviation Psychology*, *3* (1), 1–23. https://doi.org/10.1207/s15327108ijap0301_1

Parasuraman, R., Mouloua, M., & Molloy, R. (1997). Effects of adaptive task allocation on monitoring of automated systems. *Human Factors*, *38*(4), 665–679. https://doi.org/10.1518/001872096778827279

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans: a Publication of the IEEE Systems, Man, and Cybernetics Society*, *30*(3), 286–297. https://doi.org/10.1109/3468.844354

Paris, C. L. (1988). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, *14*(3), 64–78. https://doi.org/10.5555/58914.58918

Parnin, C., & DeLine, R. (2010). Evaluating cues for resuming interrupted programming tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Atlanta Georgia USA (pp. 93–102). https://doi.org/10.1145/1753326.1753342

Pasquale, F., Malgieri, G. (2023). Generative AI, explainability, and score-based natural language processing in benefits administration. Retrieved from https://papers.ssrn.com/abstract=4826707

Paul, C. L., Komlodi, A., & Lutters, W. (2015). Interruptive notifications in support of task management. *International Journal of Human-Computer Studies*, *79*, 20–34. https://doi.org/10.1016/j.ijhcs.2015.02.001

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv:2302.06590 [cs]. Retrieved from http://arxiv.org/abs/2302.06590

Pennefather, P. P. (2023a). AI and the future of creative work. In P. P. Pennefather (Ed.), *Creative prototyping with generative AI: Augmenting creative workflows with generative AI* (pp. 387–410). Apress. https://doi.org/10.1007/978-1-4842-9579-3_13

Pennefather, P. P. (2023b). Use cases. In P. P. Pennefather (Ed.), *Creative prototyping with generative AI: Augmenting creative workflows with generative AI* (pp. 339–385). Apress. https://doi.org/10.1007/978-1-4842-9579-3_12

Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2023). "It's weird that it knows what I want": Usability and interactions with copilot for novice programmers. https://doi.org/10.48550/arXiv.2304.02491 arXiv:2304.02491 [cs].

Preiksaitis, C., Sinsky, C. A., & Rose, C. (2023). ChatGPT is not the solution to physicians' documentation burden. *Nature Medicine*, *29*(6), 1296–1297. https://doi.org/10.1038/s41591-023-02341-4

Rasmussen, J., & Vicente, K. J. (1989). Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies*, *31*(5), 517–534. https://doi.org/10.1016/0020-7373(89)90014-X

Reason, J. (1990). The contribution of latent human failures to the breakdown of complex systems. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *327*(1241), 475–484. https://doi.org/10.1098/rstb.1990.0090

Ross, S. I., Martinez, F., Houde, S., Muller, M., & Weisz, J. D. (2023). The programmer's assistant: Conversational interaction with a large language model for software development. In *Proceedings of the 28th International Conference on Intelligent User Interfaces (IUI '23)* (pp. 491–514). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3581641.3584037

Salvucci, D. D., & Taatgen, N. A. (2011). Toward a Unified View of Cognitive Control. *Topics in Cognitive Science*, *3*(2), 227–230. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1756-8765.2011.01134.x. https://doi.org/10.1111/j.1756-8765.2011.01134.x

Sarkar, A. (2023). Exploring perspectives on the impact of artificial intelligence on the creativity of knowledge work: Beyond mechanised plagiarism and stochastic parrots. In *Proceedings of the 2nd Annual Meeting of the Symposium on Human-Computer Interaction for Work (CHIWORK '23)* (pp. 1–17). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3596671.3597650

Sarkar, A., Gordon, A. D., Negreanu, C., Poelitz, C., Ragavan, S. S., & Zorn, B. (2022). What is it like to program with artificial intelligence?. https://doi.org/10.48550/arXiv.2208.06213 arXiv:2208.06213 [cs].

Schellaert, W., Martínez-Plumed, F., Vold, K., Burden, J., A. M. Casares, P., Sheng Loe, B., Reichart, R., Ó hÉigeartaigh, S., Korhonen, A., & Hernández-Orallo, J. (2023). Your prompt is my command: On assessing the human-centred generality of multimodal models. *Journal of Artificial Intelligence Research*, *77*(June 2023), 377–394. https://doi.org/10.1613/jair.1.14157

Sheridan, T. B. (2012). Human supervisory control. In *Handbook of human factors and ergonomics* (pp. 990–1015). John Wiley & Sons, Ltd. Section: 34 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118131350.ch34. https://doi.org/10.1002/9781118131350.ch34

Sheridan, T. B., & Parasuraman, R. (2005). Human-automation interaction. *Reviews of Human Factors and Ergonomics*, *1*(1), 89–129. https://doi.org/10.1518/155723405783703082

Sinaiko, H. W. (1972). Human intervention and full automation in control systems. *Applied Ergonomics*, *3*(1), 3–7. https://doi.org/10.1016/0003-6870(72)90003-8

Smith, H. P. R. (1979). *A simulator study of the interaction of pilot workload with errors, vigilance, and decisions*. Technical Report NASA-TM-78482. NASA. NTRS Author Affiliations: NASA Ames Research Center NTRS Document ID: 19790006598 NTRS Research Center: Legacy CDMS (CDMS). Retrieved from https://ntrs.nasa.gov/citations/19790006598

Srinivasa Ragavan, S., Hou, Z., Wang, Y., Gordon, A. D., Zhang, H., & Zhang, D. (2022). GridBook: Natural language formulas for the spreadsheet grid. In *27th International Conference on Intelligent User Interfaces (IUI '22)* (pp. 345–368). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3490099.3511161

Stoner, H. A., Wiese, E. E., & Lee, J. D. (2003). Applying ecological interface design to the driving domain: The results of an abstraction hierarchy analysis. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *47*(3), 444–448. https://doi.org/10.1177/154193120304700341

Sukhera, J. (2022). Narrative reviews: Flexible, rigorous, and practical. *Journal of Graduate Medical Education*, *14*(4), 414–417. https://doi.org/10.4300/JGME-D-22-00480.1

Sun, J., Liao, Q. V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., & Weisz, J. D. (2022). Investigating explainability of generative AI for code through scenario-based design. In *27th International Conference on Intelligent User Interfaces (IUI '22)* (pp. 212–228). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3490099.3511119

Taekman, J. M., & Shelley, K. (2010). Virtual environments in healthcare: Immersion, disruption, and flow. *International Anesthesiology Clinics*, *48*(3), 101–121. https://doi.org/10.1097/AIA.0b013e3181eace73

Tankelevitch, L., Kewenig, V., Simkute, A., Scott, A. E., Sarkar, A., Sellen, A., & Rintel, S. (2024). The metacognitive demands and opportunities of generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 1–24). https://doi.org/10.1145/3613904.3642902

Ulfsnes, R., Moe, N. B., Stray, V., & Skarpen, M. (2024). Transforming software development with generative AI: Empirical insights on collaboration and workflow. In A. Nguyen-Duc, P. Abrahamsson, and F. Khomh (Eds.), *Generative AI for effective software development* (pp. 219–234). Springer Nature. https://doi.org/10.1007/978-3-031-55642-5_10

Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022). Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)* (pp. 1–7). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491101.3519665

Vasconcelos, H., Bansal, G., Fourney, A., Liao, Q. V., & Vaughan, J. W. (2023). Generation probabilities are not enough: Exploring the effectiveness of uncertainty highlighting in AI-powered code completions. arXiv:2302.07248 [cs]. https://doi.org/10.48550/arXiv.2302.07248

Wang, R., Cheng, R., Ford, D., & Zimmermann, T. (2024). Investigating and designing for trust in AI-powered code generation tools. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency* (pp. 1475–1493). ACM, Rio de Janeiro, Brazil. https://doi.org/10.1145/3630106.3658984

Warm, J. S., Parasuraman, R., & Matthews, G. (2008). Vigilance requires hard mental work and is stressful. *Human Factors*, *50*(3), 433–441. https://doi.org/10.1518/001872008X312152

Weisz, J. D., Muller, M., Houde, S., Richards, J., Ross, S. I., Martinez, F., Agarwal, M., & Talamadupula, K. (2021). Perfection not required? human-AI partnerships in code translation. In *26th International Conference on Intelligent User Interfaces (IUI '21)* (pp. 402–412). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3397481.3450656

Weisz, J. D., Muller, M., Ross, S. I., Martinez, F., Houde, S., Agarwal, M., Talamadupula, K., & Richards, J. T. (2022). Better together? An evaluation of AI-supported code translation. In *27th International Conference on Intelligent User Interfaces (IUI '22)* (pp. 369–391). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3490099.3511157

Wickens, C. D., Gempler, K., & Morphew, M. E. (2000). Workload and reliability of predictor displays in aircraft traffic avoidance. *Transportation Human Factors*, *2*(2), 99–126. https://doi.org/10.1207/STHF0202_01

Wiener, E. L., & Curry, R. E. (1980). Flight-deck automation: Promises and problems. *Ergonomics*, *23* (10), 995–1011. https://doi.org/10.1080/00140138008924809

Woodruff, A., Shelby, R., Kelley, P. G., Rousso-Schindler, S., Smith-Loud, J., & Wilcox, L. (2023). How knowledge workers think generative AI will (not) transform their industries. arXiv:2310.06778 [cs] version: 1. Retrieved from http://arxiv.org/abs/2310.06778

Wu, T., Terry, M., & Cai, C. J. (2022). AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)* (pp. 1–22). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491102.3517582

Xu, F. F., Vasilescu, B., & Neubig, G. (2022). In-IDE code generation from natural language: Promise and challenges. *ACM Transactions* *on Software Engineering and Methodology*, *31*(2), 1–47. https://doi.org/10.1145/3487569

Yuan, A., Coenen, A., Reif, E., & Ippolito, D. (2022). Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces* (pp. 841–852). ACM, Helsinki Finland. https://doi.org/10.1145/3490099.3511105

Zając, H. D., Li, D., Dai, X., Carlsen, J. F., Kensing, F., & Andersen, T. O. (2023). Clinician-facing AI in the wild: Taking stock of the sociotechnical challenges and opportunities for HCI. *ACM Transactions on Computer-Human Interaction*, *30*(2), 1–39. https://doi.org/10.1145/3582430

Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023). Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1–21). ACM, Hamburg Germany. https://doi.org/10.1145/3544548.3581388

About the authors

Auste Simkute has recently completed her PhD in Design at the University of Edinburgh. Her research explores explainability and human-AI collaboration in expert domains. Auste worked with Microsoft researching generative AI in knowledge work and education. She also contributed to the DCMS policy report concerning generative AI regulations in the UK.

Lev Tankelevitch is a senior researcher at Microsoft Research where he explores ways to augment human agency in collaborative work and productivity using generative AI. His research approach is mixed-methods and reflects the intersection of cognitive science, behavioral science, and human-computer interaction.

Viktor Kewenig is a PhD candidate at University College London, specialising in cognitive neuroscience and AI, with a particular focus on the intersection of natural language comprehension and multimodal understanding in both computational models and the human brain. Collaborating with Microsoft Research Cambridge, he has recently been studying the impact of generative AI on cognition in education and knowledge work.

Ava Elizabeth Scott's research explores the relationship between metacognition, intentionality, and technology-use. Prioritizing ecologically-valid methods, she collects and analyses qualitative and quantitative data on people's use of reminders and collaborative technologies. She is supervised by Yvonne Rogers and Sam Gilbert at UCL, and has undertaken multiple projects with Microsoft Research.

Abigail Sellen is Distinguished Scientist and Lab Director at Microsoft Research Cambridge in the UK, with a research team in Nairobi, Kenya. At Microsoft, she oversees a portfolio of industrial research which takes an interdisciplinary approach to designing and developing new AI-infused technologies.

Sean Rintel studies the intersection of communication, technology, and work as a Senior Principal Research Manager at Microsoft Research. He is currently exploring Generative AI, metacognition, remote and hybrid meetings, and workflows. He is an active author, reviewer, and editor in academic communities, and collaborates on external academic projects.