The Impact of Generative AI on Creativity in Software Development
A Research Agenda

Jackson, Victoria; Vasilescu, Bogdan; Russo, Daniel; Ralph, Paul; Prikladnicki, Rafael; Izadi, Maliheh; D'Angelo, Sarah; Inman, Sarah; Andrade, Anielle; Van Der Hoek, André

DOI
10.1145/3708523

Publication date
2025

Document Version
Final published version

Published in
ACM Transactions on Software Engineering and Methodology

Citation (APA)
Jackson, V., Vasilescu, B., Russo, D., Ralph, P., Prikladnicki, R., Izadi, M., D'Angelo, S., Inman, S., Andrade, A., & Van Der Hoek, A. (2025). The Impact of Generative AI on Creativity in Software Development: A Research Agenda. *ACM Transactions on Software Engineering and Methodology*, *34*(5), Article 133. https://doi.org/10.1145/3708523

Check for updates

The Impact of Generative AI on Creativity in Software Development: A Research Agenda

VICTORIA JACKSON, University of California Irvine, Irvine, California, USA
BOGDAN VASILESCU, SCS, ISR, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
DANIEL RUSSO, Department of Computer Science, Aalborg University, Aalborg, Denmark
PAUL RALPH, Dalhousie University, Halifax, Nova Scotia, Canada
RAFAEL PRIKLADNICKI, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
MALIHEH IZADI, EWI, Delft University of Technology, Delft, Netherlands
SARAH D'ANGELO, Google Inc, Canterbury, New Zealand
SARAH INMAN, Google Inc, Seattle, California, USA
ANIELLE ANDRADE, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
ANDRÉ VAN DER HOEK, University of California Irvine, Irvine, California, USA

As GenAI becomes embedded in developer toolchains and practices, and routine code is increasingly generated, human creativity will be increasingly important for generating competitive advantage. This article uses the McLuhan tetrad alongside scenarios of how GenAI may disrupt software development more broadly, to identify potential impacts GenAI may have on creativity within software development. The impacts are discussed along with a future research agenda comprising five connected themes that consider how individual capabilities, team capabilities, the product, unintended consequences, and society can be affected.

CCS Concepts: • **Software and its engineering → Software development techniques**; • **Human-centered computing → Collaborative and social computing**;

This work is supported by the 1st Copenhagen Symposium on Human-Centered Software Engineering AI, through Grants G-2023-21020 and CF23-0208, respectively. R. Prikladnicki is partially funded by Fapergs and CNPq, Brazil. A. Andrade is financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil (CAPES)—Finance Code 001. P. Ralph's contributions were funded by NSERC Discovery Grant RGPIN-2020-05001. Vasilescu kindly acknowledges partial support from the NSF awards 1901311, 2107298, and 2317168, the Google Open Source Programs Office, and a Google Award for Inclusion Research. van der Hoek kindly acknowledges support by the National Science Foundation under Grants CCF-2210812 and 2326489.
Authors' Contact Information: Victoria Jackson (corresponding author), University of California Irvine, Irvine, California, USA; e-mail: vfjackso@uci.edu; Bogdan Vasilescu, SCS, ISR, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA; e-mail: vasilescu@cmu.edu; Daniel Russo, Department of Computer Science, Aalborg University, Aalborg, Denmark; e-mail: daniel.russo@cs.aau.dk; Paul Ralph, Dalhousie University, Halifax, Nova Scotia, Canada; e-mail: paulralph@dal.ca; Rafael Prikladnicki, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil; e-mail: rafaelp@pucrs.br; Maliheh Izadi, EWI, Delft University of Technology, Delft, Netherlands; e-mail: m.izadi@tudelft.nl; Sarah D'Angelo, Google Inc, Canterbury, New Zealand; e-mail: sdangelo@google.com; Sarah Inman, Google Inc, Seattle, California, USA; e-mail: icsarah@google.com; Anielle Andrade, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil; e-mail: anielle.lisboa@edu.pucrs.br; André van der Hoek, University of California Irvine, Irvine, California, USA; e-mail: andre@ics.uci.edu.

ACM 1557-7392/2025/5-ART133
https://doi.org/10.1145/3708523

Additional Key Words and Phrases: Creativity, Generative AI, Software Development, Foundational Models

ACM Reference format:
Victoria Jackson, Bogdan Vasilescu, Daniel Russo, Paul Ralph, Rafael Prikladnicki, Maliheh Izadi, Sarah D'Angelo, Sarah Inman, Anielle Andrade, and André van der Hoek. 2025. The Impact of Generative AI on Creativity in Software Development: A Research Agenda. *ACM Trans. Softw. Eng. Methodol.* 34, 5, Article 133 (May 2025), 28 pages.
https://doi.org/10.1145/3708523

1 Introduction

Creativity has always been important to software development [45]. Creativity helps teams address the more mundane software tasks that arise everyday as well as the more specialized but typically more infrequent tasks aimed at yielding major product advances. For example, everyday creativity [3], in the form of creative problem solving, aids in resolving the challenges that inevitably arise in the commonplace tasks needed to build software. Examples of such tasks include coding a new feature either individually [55] or collectively [57], designing an appropriate solution to some new requirement [90], or fixing a bug [131]. The more infrequent "big-bang" (Big-C [61]) creativity is needed when major innovations are required, perhaps in re-architecting an application to incorporate the latest Artificial Intelligence techniques or identifying new product features that will hopefully delight customers [80]. To flourish, both forms of creativity rely on the interplay of talented individuals [73], collaborative teams [133], and supportive environments [4]. Use of appropriate techniques (and concomitant tools) such as whiteboarding [51], ideation [102], and hackathons [47] aids creativity too.

Both organizations and individuals benefit from the outcomes arising from creativity in the workplace [107]. Creativity is considered an antecedent to the innovation [9] critical for companies to perform and grow [93]. Creativity helps companies produce novel products and improve their processes [79] as well as work towards meeting their sustainability efforts [89]. As such, companies rely on employees exhibiting creativity. Indeed, the World Economic Forum considered analytical and creative thinking the two most important skills for workers in 2023 [39].

Individuals also benefit from creative work. Being creative can improve emotional well-being [1], job satisfaction [42], and career growth [79]. Moreover, creativity is recognized as the trait of an exceptional developer [50]. It is therefore helpful for organizations to cultivate creativity in individuals by allowing them to express creativity at work [112].

Understudied [6], creativity in software engineering is at a watershed moment, as the emergence of **Generative AI (GenAI)** could have a major impact on individuals, employers, and the products they build. It has already been observed that GenAI can help software professionals be creative by generating ideas for a variety of different tasks (e.g., programming, designing a UI, identifying new potential requirements) [15], which can thereby influence the resulting products. Furthermore, if GenAI becomes as powerful as some people believe it will in terms of taking the "rote" out of software development [116], more or perhaps nearly all the work performed by software professionals may necessarily be creative work, and companies may increasingly view creativity as central to software professionals' skills and products' competitive advantage.

Therefore, although most research, studies, and discussions in the media have concentrated on GenAI and programmer *productivity* (e.g., [14, 140]), understanding the short-term, medium-term, and long-term impacts and connections between GenAI and *creativity* in software development warrants equal, if not greater, emphasis. This article takes the first step to drawing attention to this topic via two contributions:

(1) Driven by a set of potential future scenarios, we articulate the many different kinds of impacts GenAI may have on the intersection of creativity and software development by applying the Marshall McLuhan tetrad [85] (a construct consisting of four questions for analyzing effects of technology on society) to each of the P's in the "4P" framework of creativity [110] (Person, Product, Process, and Press).
(2) Derived from this analysis of potential impacts, we propose a research agenda comprised of five interdependent themes. Together, these themes cover three tangible, more immediate effects, namely individual capabilities, team capabilities, and products, as well as two indirect, longer-term effects, namely unintended consequences and society.

Overall, our work calls for future research addressing both the individual potential impacts in the tetrads as well as overarchingly each of the five themes in the context of the ongoing evolution of GenAI. This research is needed to both amplify benefits and mitigate harms of GenAI and related innovations.

The remainder of this article is organized as follows. In Section 2 we introduce key foundations upon which our perspective rests. Section 3 introduces related work and Section 4 presents our detailed exploration of the potential impacts of GenAI on creativity in software development. Section 5 then introduces the research agenda, while Section 6 discusses its implications. Section 7 concludes the article.

2 Foundations

Numerous definitions of creativity have been proposed [111], each with a somewhat different focus. Some address the human aspect of being creative (e.g., "Coming up with novel and useful ideas" [5]). Others emphasize creativity in the product (e.g., "A product or response will be judged as creative to the extent that (a) it is both a novel and appropriate, useful, correct or valuable response to the task at hand, and (b) the task is heuristic rather than algorithmic" [2]). Yet others characterize what the creative process (e.g., "A process of becoming sensitive to problems, deficiencies, gaps in knowledge, missing elements, disharmonies, and so on; identifying the difficulty; searching for solutions, making guesses, or formulating hypotheses about the deficiencies; testing and retesting these hypotheses and possibly modifying and retesting them; and finally communicating the results" [65]). For purposes of this article, we adopt Boden's definition: "Creativity is the ability to come up with ideas or artifacts that are new, surprising and valuable" [16]. This definition succinctly blends aspects of the prior three definitions, given that the human aspects, the product, and the process all are relevant to creativity in software development.

To explain creativity, various theoretical frameworks have been proposed over the years (e.g., the Systems Model of Creativity [26], the Componential Model of Creativity [4]), each reflecting the multi-faceted nature of creativity. In this article, we utilize the 4P framework [110] and the 4C model of creativity [61]. The 4P framework considers that creativity can be explained by four components: the *Person*, the *Product*, the *Process*, and the *Press* (environment). Person refers to the creative individual and consists of their personality, behaviors, and skills that contribute to their creativity. Product is the outcome of the creative process—in the case of software development, this could be an artifact such as code or a document or the actual end-user product. Process is the methods and techniques used to generate new ideas, such as brainstorming. Finally, Press acknowledges the impact of the environment (e.g., social, cultural, economic, physical) on creativity. This framework is especially well-suited to exploring creativity in software development as products are built by teams of individuals using processes and tools operating in an environment specific to their organization. Adopting this framework thus allows us to explore the potential impact of GenAI on creativity in software development from a set of comprehensive, relevant, and structured dimensions. Complementing the 4P model is the 4C model, which considers creativity to exist on a continuum ranging from mini-c that represents "novel and personally meaningful" creativity, through the everyday creativity of little-c that occurs in the course of daily tasks, to Pro-c creativity exhibited by professionals with years of experience, and ending with Big-C creativity that represents major breakthroughs and more "big bang" creativity. Utilizing this continuum enables us to consider a range of creative experiences exhibited by software professionals in their day-to-day work and also when creating major new products.

3 Related Work

We briefly review relevant related work on creativity within software development, the use of GenAI in software engineering, and research into GenAI and creativity beyond the software engineering domain. The latter research is of interest as some of the findings may be applicable to software development: a consideration we will return to in Section 4.

3.1 Creativity within Software Development

Creativity within software development is an understudied topic [55, 91]. For Person creativity, only a handful of studies exist. One example finds that personality traits can predict a programmer's creativity [7] and another observes that having overly templated requirements can result in fixation [90] when designing software. Product creativity also has received little attention beyond a few studies on **Open Source Software (OSS)**, with some claiming it is more creative than commercial software [62, 97] and another study framing innovation in OSS projects as novel reuse of existing libraries [37]. Within Process creativity, much research has looked into requirements engineering [6], such as techniques to encourage creative thinking [80] in identifying requirements. Whiteboarding and brainstorming are popular creativity techniques used by developers [51]. These techniques, alongside others such as mob (AKA ensemble) programming [123], are also suited for fostering creativity in hybrid teams [57]. The environment (called "Press" in the 4P framework for historical reasons) can affect a developer's creativity. Working in a psychologically safe climate [34] can aid creativity [137], as can having a supportive manager who encourages new ideas [27]. Having high levels of team collaboration, however, can hinder creativity in software development [53]. Distance can lead to reduced creativity since distributed software designers spend less time exploring the problem space than co-located designers [58].

3.2 GenAI and Software Engineering

The adoption of **Large Language Models (LLMs)**, a form of GenAI, for software development has increased dramatically in the last few years, with a commensurate explosion in research exploring new opportunities for LLM use in support of different development tasks (e.g., coding [14], code review [28], testing [115], debugging [60]) and research studying how industry is grappling with adopting LLM driven tools (e.g., [104, 113]). Two recent literature reviews [36, 54] find the majority of research has focused on code-generation tasks, followed by testing. Neither review mentions studies that directly target creativity, although it is indirectly hinted at when discussing potential research into hallucinations and whether the hallucinations could be harnessed in some way, e.g., to identify new features [36]. Both papers also include research into the use of LLMs for tasks that developers typically consider as creative (e.g., bug fixing), showing the potential for LLMs to disrupt creative work. The papers, however, do not explicitly discuss creativity.

Despite the plethora of new publications that have continue to emerge on the topic of LLMs and GenAI in software development, then, there is no research that directly examines the impact of GenAI on creativity in software development. This is perhaps unsurprising given the lack of research on creativity in software development more generally. Today, however, it is more important than ever to truly engage in research on creativity, GenAI, and software development. Such is the purpose of this article: to raise awareness of the issue and identify important directions for future research.

3.3 GenAI and Creativity

In contrast to software engineering, other domains including **Human-Computer Interaction (HCI)**, organization and business management, and cognitive science have explored the impact of LLMs on creativity. One common research theme is investigating how humans can co-create with GenAI for a variety of tasks commonly considered as creative including interaction design [63, 117, 129], music composition [77], writing [64, 72, 136], and research ideation [75, 108]. Many of the studies consider how LLMs can aid idea generation (i.e., as acting as a brainstorming aid [94]) as well as subsequent evaluation and selection of ideas.

Empirical evidence on the creativity of LLM outputs is mixed. Various studies find LLMs performing at the same level as humans on creative thinking tasks. For instance, the creativity of human and GenAI produced stories was found to exhibit no significant differences [98]. Other studies show GenAI outperforming humans. Consultants using GenAI [30] produced more than 40% higher quality solutions in a controlled study, where quality was an aggregate score comprising creativity, analytical thinking, writing proficiency, and persuasiveness. Ideas generated by chatbots were considered more creative than those generated by humans [17]. LLMs, too, outperformed humans in generating uncommon and creative uses for everyday objects when completing standardized tests of divergent thinking such as the **Torrance Tests of Creative Thinking (TTCT)** [52] and the **Alternate Uses Task (AUT)** [68, 94, 124].

Conversely, other studies highlight LLMs under-performance in comparison to humans [118]. Koivisto and Grassini [68] noted that the most creative human ideas exceeded those of chatbots while another study comparing LLM-generated short stories to stories written by human experts found that human-written stories significantly surpass the LLM-generated stories [22] when measured using the TTCT creativity dimensions. Other studies have found evidence of a homogenization effect (i.e., less semantically distinct outputs) from LLM assistance across a variety of GenAI text and image generation tools [8, 12, 31, 99, 139]. Bias and the presence of "tropes" were also found in a study in a comedy setting [88].

Researchers have investigated reasons for and solutions to this under-performance. The temperature hyper-parameter[1] of LLMs, typically expected to act as a "creativity parameter" [29], is "[only] weakly correlated with novelty," and overall "the influence of temperature on creativity is far more nuanced and weak" than might be expected [103]. It could be that inadequate interaction parameters steer users towards quickly converging on ideas rather than exploring a broad design space [126]. Numerous studies have explored ways to increase the novelty and creativity of ideas contributed via a LLM. These include: encouraging the user exploration of design spaces [23, 108, 126]; prompt engineering techniques to generate diverse ideas [38, 132]; persona-based approaches where prompts define different personas [46, 75] for the LLM to adopt thereby introduce diversity into the responses; as well as multi-agent approaches where agents are asked to role-play [78].

Finally, concerns persist that the autoregressive nature of LLMs will "prevent them from reaching transformational creativity" [40], and that there continues to be a risk of generative monoculture, i.e., "a significant narrowing of model output diversity relative to available training data for a given task" despite countermeasures such as altering sampling or prompting strategies [134]. More recent work advocates the importance of viewing humans as co-creators and developing appropriate

[1]The temperature hyper-parameter controls the randomness of a GenAI model's output. A low temperature results in more deterministic outputs whereas a high value makes the response more random, diverse, and possibly nonsensical.

co-creation modes of interaction [19, 56, 83], and for better understanding the many tradeoffs involved in human-AI co-creation [20, 49].

4 Exploring the Potential Impact

Much has been said about how GenAI might eventually shape software development, both in the gray literature as opinion and reflection pieces (e.g., [21, 84, 130]) and the research literature as research agendas and vision papers (e.g., [35, 36, 96, 114, 122]). Complementing this emerging body, our article contributes a research agenda on a topic that has not yet received this kind of attention: creativity in software development. To shape the research agenda we present in Section 5, we took a two-phased approach to comprehensively and systematically explore the many potential impacts GenAI might have on creativity in software development. First, we speculated future scenarios [32] of how GenAI may impact creativity in software engineering. Second, using the scenarios as the driver, we applied parts of the Disruptive Research Playbook [125] (from here on referred to as "playbook"), which was specifically designed for identifying socially relevant software development research questions when studying disruptive technologies (such as GenAI).

Note that for purposes of this article we take a broad view as to what activities constitute software development. We include the traditional activities such as requirements, design, coding, and testing as well as complementary activities (e.g., product visioning/definition, **User Experience (UX)**, UI design). Considering software development more broadly enables us to identify potential impacts of GenAI on the act of software development as well as on its boundaries, i.e., the products being developed, thus providing a richer research agenda.

In terms of potential future scenarios, much has been said about how GenAI will impact software development, from mundane to far-fetched. Some predict, for instance, that GenAI will merely be another tool in software professionals' arsenals, amplifying their ability to perform work but not replacing their jobs [33]. Others compare the current disruptive impact of GenAI on many professions, including software development, to that of automated looms on 19th-century weavers [86], where automation significantly changed the nature of work and led to major job displacement [67]. A good research agenda should be based on the spectrum of possibilities rather than a single prophecy; therefore, we drew up a series of brief, high-level scenarios intended, collectively, to cover this spectrum. To identify the scenarios, the paper authors initially captured their various suggestions in a shared document. Subsequently, the authors discussed, expanded, refined, and eventually agreed on the scenarios through online and offline discussions.

The playbook builds upon the McLuhan tetrad [85] to frame the research landscape. The tetrad facilitates analyzing potential impacts of new technologies by asking what the technology: (i) *enhances*, (ii) *obsolesces*, (iii) *retrieves* (from obsolescence), and (iv) *reverses* into (i.e., how can its effects change direction when pushed to extremes?). The answers to these questions can then be used to choose specific phenomena to study, identify potential research questions, articulate higher-level research thrusts, and determine research strategies best suited for pursuing the questions and thrusts. Inspired by the spectrum of scenarios, we brainstormed an initial set of potential impacts to populate a tetrad for each P of the 4P framework of creativity. We equally sought to cover the 4C framework with the impacts incorporated in each of the tetrads, since creativity is not always (and in fact, generally not) about major innovation (Big-C). Everyday creativity (often mini-c or little-c) and the creativity exhibited by seasoned professionals when shaping a software product (Pro-C) are equally important. We subsequently iterated over these impacts to identify additional impacts and refine the initial set until we settled on a broad set of potential impacts across all of the tetrads.

4.1 Future Scenarios

It is impossible to predict exactly how software professionals will integrate GenAI into their daily routines. Early observations suggest that GenAI will affect how software professionals practice creativity [15]. However, projections of what GenAI can and cannot achieve regarding creativity support differ significantly. It is particularly important in that regard to contextualize GenAI's achievements in other creative tasks as discussed in Section 3.3: Much research on GenAI and its promise for creativity tends to be about singular, relatively short tasks that are mostly unconstrained (e.g., write a poem on some topic and in a certain style, generate alternative uses, provide a list of ideas). While some individual software tasks can be like this (e.g., provide a snippet of code that does "X," generate a test case for this code, produce a set of personas), these results do not necessarily transform more broadly to all of the creative aspects of software development. Software development, after all, is a complex and continuous undertaking that involves numerous kinds of constraints that influence the role of creativity. Existing code bases significantly limit what can realistically be achieved. Customer or user requirements prevent certain choices because ultimately the software must support their work practices. Hardware and infrastructure limitations abound. In this much more challenging context, the ideas and contributions of GenAI may not be as helpful, given where they need to be sited: They must be realistic with respect to the constraints, useful towards the overall purpose and roadmap, and fit with the code that is already there. As such, the scenarios below on the one hand contain elements that seek to replicate studies and findings from other disciplines and on the other hand incorporate elements that tailor to the more unique needs of creativity in software development.

Each scenario is fundamentally about the following question: (1) how well does GenAI eventually work in the domain of software development and specifically so in supporting its creative aspects? In other words, the scenarios range in the level of creative output and competency that GenAI is assumed to be able to achieve when it comes to the software domain. While some people believe that GenAI will rapidly improve and have far reaching abilities, this is far from certain given the challenging space of software development sketched in the above. GenAI could well improve slower than anticipated or encounter fundamental limitations that cannot be overcome, rendering it unable to perform certain or perhaps even most creative tasks. Precisely where the limitations lie, or if they even exist, is unknown at this time, hence our scenarios are diverse in envisioned future GenAI creative capabilities.

Three additional questions play an important role in the scenarios we constructed: (2) to what extent do software professionals embrace GenAI for creative work, (3) who is in charge, and (4) does the software industry experience reduced or increased demand? As with the first question, the answers are unknown. Speaking to the second question (to what extent do they embrace GenAI for creative work), for the moment, we see software professionals rapidly adopting GenAI. That, however, is not universally true for all developers and certainly not for all tasks. Indeed, most tasks for which GenAI is being adopted are mundane [127]. As such, depending on their experiences with GenAI when software professionals turn to more creative work, this may continue, stall, or reverse. GenAI may turn out to be unreliable in delivering creative solutions, unsuitable for certain creative tasks, or developers might feel their livelihood is being threatened by GenAI overtaking their creative endeavors (witness the Screen Actors Guild and its strike [119]). As such, our scenarios adopt a set of different stances with respect to GenAI adoption for creative software tasks.

With respect to the third question (who is in charge), we note that autonomous agents have long been a topic in the artificial intelligence literature more broadly [41]. Current GenAI technologies in use in software development firmly have the user in charge: through prompt engineering [82], the software professional steers, and thus controls the outcomes. That said, studies have already examined ways in which autonomous GenAI agents can collaborate in creative endeavors (e.g., multiple autonomous agents each take on a different persona and collaborate in problem-solving [138]). Though these studies are outside of software development thus far, several of our scenarios consider the possibility of such autonomous agents in use and contributing to creative software development tasks.

Finally, the fourth question (reduced or increased demand for software) considers that many people seem to assume that the more work is automated by GenAI, the less work will be left for software professionals. While demand does often work that way, it does not always. By lowering the barriers to creating software and related products, GenAI may create whole new kinds of problems and work for humans. Software development, both creative and otherwise, may therefore experience increased demand such that all the work performed by GenAI just creates more work, more opportunities for creativity, and more demand for more and more different kinds of software, some of which we have yet to invent. Again, our scenarios touch upon different such possibilities.

These four parameters along which the future can significantly vary give rise to many different scenarios. As bookends, we use two scenarios that, in adopting the human-centric viewpoint on GenAI of both the Copenhagen manifesto [114] and Shneiderman's Human-Centered AI [121], we will call the "optimistic" versus "pessimistic" futures. In the optimistic scenario, software professionals have significantly more time for creative thinking because they can offload their mundane software development work to GenAI. Moreover, GenAI serves as a highly creative digital partner to help them brainstorm, elaborate, and assess ideas for products, features, architectures, UXs, bug fixes, refactorings, and more. Creative and exciting new products emerge all the time, helping society lead a more fulfilling life. Overall, the creative capacity of the software industry is dramatically expanded and humans remain an integral part.

Contrastingly, in the pessimistic scenario, software professionals are marginalized as GenAI not only performs all mundane software development tasks, but also has advanced to the point where it can do—and does with absolutely minimal direction—all of the creative work software professionals would normally perform. The role of humans in software development work is greatly diminished. "Software developer" is no longer a job title and there is rarely a need for creativity from humans, with only a few master creators left driving the GenAI to create new products almost at will and with simple, straightforward instructions. The products are nonetheless still amazing and definitely benefit society. The creative capacity of the software industry (or better: what is left of it) is essentially limitless, yet the human is no longer a meaningful part of the process.

As we said, these are extreme positions, so let us consider five potential scenarios that could occur between the two extremes. Please note that these are presented in no particular order, given that the many possible answers to each of the four questions create a multi-dimensional space of many possibilities. These five scenarios, therefore, focus on rather different choices in this space.

— *GenAI as idea generator*. GenAI as it presently is capable of assisting software professionals in generating ideas (e.g., identifying possible stakeholders, offering alternative code snippets that implement certain functionality, suggesting applicable design patterns) does not evolve much in future. Software professionals learn to effectively obtain these ideas through clever prompt engineering, but also know that the results are not fully reliable and must be filtered and augmented carefully, with human judgment and intuition. The ideas remain just that: ideas, both good and bad, upon which humans build further in their creative pursuit. GenAI in this case is simply another tool in the creativity tool belt of software professionals they can choose to use when they want to, just like techniques such as brainstorming and mob programming. Not much changes otherwise.

— *GenAI as creative partner*. GenAI fails to live up to the hype, but in select areas it actually performs extremely well—so well, in fact, that GenAI results can be relied upon by software professionals in the creative process. As one example, GenAI could propose novel features by automating market intelligence and ingesting user feedback. As another example, GenAI could serve as the critical voice in design meetings, examining proposed solutions and offering up potential shortcomings that the software professionals should consider. In these examples, and others like it, GenAI serves a specific role with a specific task, augmenting the otherwise human creative process. Software professionals would remain firmly in charge of making decisions and involving other humans' insights and feedback. Overall, the creative process proceeds more quickly and the results improve as well; GenAI is considered a useful, albeit virtual partner. Human roles are shifted, but no jobs are lost.
— *GenAI as dev team*. GenAI has replaced most members of the human development team, including the software architects who are normally responsible for making the main technical decisions and the programmers who push the designs forward into code and deployment. Specialist software professionals who interface with customers users are still valued and employed, but they provide their input to GenAI, which does the remainder of the work of designing new features as well as coding, testing, fixing bugs, and running AB tests [66] to determine the best feature and design for the product's market goals. After a successful pilot with one team, the company pushes GenAI into all product teams, fires or reassigns personnel, and thereby saves salary cost. Initially, the company gains a tremendous market, but after several years, it suddenly loses its advantage as competitors have come up with much more innovative solutions driven by human insight. GenAI provided a medium-term advantage, but its reliance on what already exists led to it creating mediocre features. Speed provided an advantage, but GenAI failed at providing the true innovation needed for long-term survival of the company.
— *GenAI as creative director*. GenAI is in charge of the creative efforts in the organization and it has at its disposal both human software professionals who it can tap for certain kinds of tasks as well as a variety of other GenAI agents that each provide specialized roles. Having been trained with data on numerous creative projects, the primary GenAI understands the parameters of successful creative exercises, what kinds of tasks are needed when, when to consider alternatives, under what conditions to push for more, what level of innovation and quality of product it can rely from different human workers and different GenAI agents, and more. It uses this knowledge to orchestrate the software professionals and other GenAIs agents in an intricate network of activities where each actor—human or agent—is provided with just the information it needs to get a task done. Overall visibility into what is transpiring is lost to the software professionals, who merely see their job as providing an income. They are busy all the time; as soon as they complete one task, another arrives at the behest of the primary GenAI. Resulting products are innovative, though exactly how they came to be will never quite be known.
— *GenAI canceled*. GenAI is capable of supporting software professionals in many different creative tasks, ranging from everyday creative problem solving when bugs are encountered or new UI features need to be integrated to helping software teams ideate the next innovative new features that will help the company retain or advance its market share. Despite having this powerful capability at hand, the software professionals in the organization have come together and decided they will not use GenAI for their creative work, instead relying on their own creativity and ingenuity. They are less concerned about job losses, as the company has a seemingly endless pipeline of customers. Rather, after carefully monitoring their productivity and creative output, they have come to the realization that, while GenAI helps them at times

to be more efficient and at other times think of something they had not thought about, it regularly also causes them too much rework and too much examination of suggestions that they do not fully understand. They are particularly concerned, and indeed have on a number of occasions, unearthed new kinds of security risks as a result of GenAI freely combining prior work. Overall, the net gain has been minimal, if that, and despite company leadership pushing for what it sees as an important way forward, the software professionals have put GenAI aside and are enjoying developing more innovative and secure products on their own.

We note that the set of scenarios is not exhaustive. Moreover, the outcomes that we sketched could well be different under each scenario, for instance by the professionals in the last scenario actually embracing GenAI because it generates more secure solutions than they are able to themselves or by the company in the third scenario actually maintaining its competitive advantage because the GenAI driven solutions are not mediocre but on par with human software professionals. This is precisely why research is needed to understand how GenAI might shape the future of creativity in software development. Indeed, if the research discussed in Section 3.3 is any indication, it is well likely that for the foreseeable future the precise impact of GenAI on creativity in software work will not be conclusive, and that the details of the precise settings where the studies take place serve as important parameters for the conclusions being drawn. We thus did not only use the above scenarios as input for populating the tetrads, but varied the diversity of settings, human choices, GenAI capabilities, and outcomes to generate variants of the scenarios that led to us identifying additional potential impacts that we incorporated in the tetrads.

We also observe that the scenarios are crafted from the perspective of the capabilities of GenAI, but clearly impact software professionals and their teams in many different ways: not only what work they perform and how, but also their psychosocial wellbeing, sense of agency, and ability to hone their craft. Indeed, many competing forces are exhibited by the scenarios, from capabilities of the GenAI versus those needed by software professionals, to company desires versus those of the professionals, to short term impacts versus medium and long term impacts. These human and other factors are equally considered by the tetrads that we present in the next section.

Finally, we expressly withhold judgment as to what outcomes are considered “good” or “bad” in these scenarios. Clearly, the scenarios implicitly hint at a range of deeply ethical considerations. Rather than positioning ourselves as “for” GenAI or “against” GenAI in software development, or for certain forms of GenAI to be acceptable in the software profession and others not, we feel a neutral research agenda as presented in Section 5 is more appropriate. We nonetheless return to the topic of ethics in the discussion presented in Section 6.

4.2 Tetrads

To use the McLuhan tetrad, we select GenAI as our disruptive technology and limit the phenomena under consideration to creativity in software development. We specifically combine GenAI, creativity in software development, and the 4P framework (Person, Product, Process, Press) using four tetrads; one for each of the Ps. The Person tetrad, for example, poses four questions:

(1) How does GenAI *enhance* an individual’s creativity?
(2) Which creative outcomes become *obsolete* due to GenAI?
(3) What previous creative outcomes could GenAI *bring back* to the foreground?
(4) If overly relied upon, how could GenAI *disrupt* creative outcomes?

The authors together brainstormed answers across each of the four tetrads. While brainstorming, they referred to the future scenarios discussed earlier and considered the impact on both creativity in the everyday work of software teams and the “big-bang” creativity common to major innovations,

Table 1. McLuhan's Tetrad Considering Potential Impacts of GenAI on Person (Software Professional) Creativity

ENHANCES *How does GenAI enhance an individual's creativity?*	**OBSOLESCES** *Which factors considered relevant to an individual's creativity become obsolete due to GenAI?*
Idea Generation. GenAI provides potential starting points for brainstorming, offers alternative creative directions, and helps avoid design fixation. *Cross-Domain Inspiration.* Exposing software professionals to ideas and solutions from other fields broadens creative stimulation. *Faster Debugging.* Bugs can sometimes be vexing. GenAI provides creative suggestions tailored to the local codebase for how to diagnose them. *Being Thorough.* GenAI provides checklists of things for software professionals to think about as they work. *Understanding Stakeholders.* By impersonating different stakeholders (e.g., a tester to identify edge cases), GenAI brings valuable knowledge to creative tasks. *Choosing Right.* An individual's alternative designs are assessed and compared per architectural fitness functions and other criteria by GenAI to assist software professionals in choosing the optimal solution for a given situation.	*Differences in Personality Traits.* Anyone can be creative irrespective of whether they have traits commonly associated with creativity (e.g., Openness). *Deep Thinking.* Software professionals offload (part of) their creative thought process so time for deep thinking is no longer required or at least reduced. *Reflection.* Long considered important to breaking through challenging problems, reflection on the problem or solutions is no longer required. *Expertise.* No longer required, as all relevant domain and technical knowledge is provided by GenAI. *Mentorship.* Other software professionals no longer need to serve as mentors, since GenAI provides personalized feedback and learning opportunities.
RETRIEVES *What factors relevant to an individual's creativity could GenAI bring back to the foreground?*	**REVERSES INTO** *What does GenAI do to an individual's creativity when GenAI is pushed to extremes or overused?*
Sketching. With more time and to balance interactions with GenAI, the practice of manually sketching designs and random doodling becomes ubiquitous among software professionals and helps drive creativity. *Verification.* Skills in reading and interpreting ideas and solutions, as well as applying formal verification, return to ensure GenAI's ideas and solutions are valid and appropriate. *Creative Work.* Software professionals, with more time to engage in fun, creative work, come up with what are actually unique and groundbreaking ideas, as opposed to ideas within the confines of being beholden to productivity demands and marketing promises.	*Rubber Stamping.* Instead of software professionals being experts who carefully judge and augment GenAI's output, they subjugate themselves to GenAI expertise and no longer push their own critical thinking. *Loss of Consideration of Alternatives.* A human tendency is to be biased towards the ideas near the top of a list, which may lead to insufficient exploration of the alternatives offered by GenAI. *Loss of Creative Skills.* As GenAI takes over more and more creative tasks, software professionals lose their critical thinking and problem-solving skills and are no longer able to address situations where GenAI fails.

so to cover both the day-to-day problem-solving that drives software development all the time and the more infrequent, high-level visioning of products and their features. Tables 1–4 show the four resulting tetrads. We encourage the reader to carefully study the content of each, and perhaps augment them with their own projections as answers to the questions.

Continuing with the example of the Person component of creativity, Table 1 contains various potential impacts of GenAI on the Person, out of which we highlight a few here that we consider particularly important. Note that we do not necessarily predict all of these will happen. Additionally, note that each of the stated impacts may only happen to a smaller or much smaller degree of effect than our brief, rather absolutely stated entries in the tetrads imply. Moreover, it could be that certain impacts only occur for some portion of the full population of software professionals and software teams, or only in certain settings. Nonetheless, by considering the breadth of potential impacts, a thoughtful research agenda can be shaped. While creativity could be enhanced by GenAI *assisting with idea generation* and providing *cross-domain inspiration*, there is a risk that at the extremes software professionals rely so much on GenAI for creativity that they *stop considering alternatives* and *lose the creative skills* critical to software development. The potential also exists for GenAI to *obsolete factors known to aid creativity such as personality traits and expertise*, as these are no longer relevant when GenAI helps the individual be creative. GenAI could indeed be a great leveler, bringing all software professionals to the same creative level. However, using GenAI will require software professionals *to continue to hone their verification skills* to ensure the generated ideas are valid.

Product creativity (Table 2) may be enhanced if the use of GenAI allows the *rate of delivery of new, innovative features to be increased* alongside *customized UXs*. On the flip side, there is a risk that *products become homogenized* due to an over-reliance on GenAI to determine new features as

Table 2. McLuhan's Tetrad That Considers Potential Impacts of GenAI on Product Creativity

ENHANCES *How does GenAI enhance creative outcomes?*	**OBSOLESCES** *Which creative outcomes become obsolete due to GenAI?*
Continuous Enhancement. With continuous deployment as the backdrop, the use of GenAI speeds up the stream of new, innovative, and valuable product features. *Customized UXs.* GenAI makes products more appealing and useful to users by helping create highly personalized and adaptive UXs. *Taking Stock.* GenAI analyzes market trends, user feedback, and product performance to help companies identify product opportunities and enhancements. *Making the Impossible Possible.* GenAI allows teams to build software systems that are intractable with today's technology. *New Product Ideation.* GenAI promotes radical innovation by identifying entirely new classes of products that people never realized were needed.	*Standard Solutions.* Solutions that are copycats will become a thing of the past because GenAI provides tailored solutions that uniquely fit each situation. *Frameworks.* Since every application is generated quickly and directly through prompt engineering, general-purpose frameworks are no longer required and replaced by custom-generated code. *Intermediate Design Artifacts.* With GenAI, typical intermediate design artifacts such as concept sketches, wireframes, UML diagrams, and the like are longer be needed. *Art.* The art that goes into software, such as iconography, color schemes, and GUI layouts, is all automatically generated, no longer needing to be designed manually. *Specialized Artifacts.* Because GenAI inherently knows about everything, artifacts that once were considered the domain of specialists (e.g., music and level design in computer games, database design for enterprise systems) are simply generated, and no longer require creative human engagement.
RETRIEVES *What previous creative outcomes could GenAI bring back to the foreground?*	**REVERSES INTO** *If overtly relied upon, how could GenAI disrupt creative outcomes?*
Patterns and Styles. While still considered important overall, thousands of patterns and styles exist that are much less frequently accessed and leveraged than the handful everyone knows. These are now readily brought to bear to any software design problem through GenAI. *Old Designs.* Design solutions embedded in old software that has long been forgotten work very well for newer situations, whether outright or indirectly in a new interface or language. *Analog Design Principles.* There is a renewed interest in analog design principles and their integration into digital products, driven by GenAI's ability to synthesize and apply wide-ranging and cross-domain design philosophies.	*Echo Chamber.* Ideas and software solutions created by GenAI infiltrate the training data of future GenAI, causing GenAI to become self-reinforcing and no longer creative. *Homogeneous Products.* Different software professionals all choose the same recommended ideas, thereby reducing the diversity of products in the market. *Overly Creative Products.* GenAI's suggestions are too complex or too different and thereby not effective for situations where routine products suffice. *Biased Products.* GenAI invariably contains various biases. As a result, products could be exclusionary to certain users or become too general when data and model scientists overcompensate for the biases. *Harmful Products.* GenAI could generate harmful products by replicating harmful features that are common in its dataset (e.g., UX dark patterns, addictive game features, encryption backdoors).

well as a risk that *harmful products incorporating dark patterns*[2] could proliferate. In using GenAI, *common frameworks may no longer be required* as all code is generated from the ground-up, but *interest in analog design principles*[3] *may be revived* due to GenAI incorporating wide-ranging design philosophies.

Since GenAI is a tool, it is not surprising that the creative process could be impacted in numerous ways (Table 3), such as a creative process that benefits from GenAI *automating more mundane tasks* or *GenAI acting as the moderator* in practices such as brainstorming, thereby leading to better outcomes. However, if GenAI is relied upon too much, *software professionals might lose their creative intuition* or *experience a loss of trust in their colleagues. Siloed specializations may become eliminated* as GenAI enables software professionals to contribute to creative work outside their own area. Additionally, *the need for design techniques* such as user studies with real people may become obsolete, as GenAI can simulate the people. However, using GenAI *may further promote tasks used today to stimulate creativity* such as pair programming and idea generation techniques in order to solve problems too complex for GenAI or to provoke solutions that differ from GenAI generated ones.

Ways in which the creative environment (Table 4) could be impacted include *contextual information being provided automatically* by GenAI listening in and offering suggestions, or, taken to extremes, *software professionals losing pride in their work* since GenAI does all of the interesting creative work. *Colleagues may no longer be needed* as GenAI can do all the support work needed to design, build, and deploy a new system. It may also *be more important for management to be supportive of developer creativity* as creativity will be a strategic business differentiator.

[2]So called dark patterns trick users into taking actions they may not have taken otherwise, often to the benefit of companies [48].
[3]Analog design makes use of tactile materials such as pen and paper as opposed to digital design software and hardware inputs [18].

Table 3. McLuhan's Tetrad That Considers Potential Impacts of GenAI on Process Creativity

ENHANCES *How does GenAI enhance creative processes?*	**OBSOLESCES** *Which creative processes become obsolete due to GenAI?*
Mundane Activities. Important activities that feed into creativity (e.g., competitor analysis) are performed by GenAI, relieving the software professional from having to engage in these. *Rapid Prototyping.* GenAI automates the process of going from rough ideas to prototypes, experimentation, reflection, and iteration. *Collaboration.* GenAI helps bridge diverse team members and diverse teams by synthesizing inputs, explaining different perspectives, helping overcome terminology barriers, and suggesting non-obvious, integrative solutions. *On-Demand Creativity.* GenAI understands when and what kind of creativity is needed at what points in the development process; it suggests the best hybrid team of software professionals and GenAI agents. *Moderation.* Brainstorming and other creative activities are no longer led by a human, but GenAI orchestrates these activities entirely, continuously steering participants in the right direction.	*Brainstorming.* Structured or unstructured idea generation is no longer needed since GenAI takes the place of these kinds of human sessions. *Siloed Specializations.* Divisions between specialized roles in organizations and creative processes blur as GenAI enables any software professional to engage in and meaningfully contribute to many different tasks of interest. *Design Techniques.* Techniques such as user studies, focus groups, cognitive walkthroughs, and others are no longer needed since GenAI understands user demands and desires. *Hiring Specialist Design and Innovation Companies (e.g., IDEO).* With GenAI, specialist design and visioning companies are no longer needed to develop innovative approaches to complex problems.
RETRIEVES *What previous creative processes could GenAI bring back to the foreground?*	**REVERSES INTO** *If overtly relied upon, how could GenAI disrupt creative processes?*
Pair Programming. Programming in pairs or even mobs is a vehicle for creative problem-solving. Pair or mob programming happens today, but becomes much more important to address the remaining challenging problems GenAI cannot solve. *Idea Generation Techniques.* To counter GenAI providing mundane solutions, organizations prioritize and amplify events and techniques for out-of-the-box thinking, including brainstorming, hackathons, design thinking sprints, and so on, making them part-and-parcel of the everyday experience of software professionals (rather than infrequent, special encounters). *Analog Creativity Techniques.* There is a resurgence in the use of analog creativity techniques (e.g., paper prototyping), as teams seek to balance GenAI's capabilities with tangible, hands-on methods that foster deep thinking and innovation. *Manual User Research.* To avoid becoming entirely disconnected from the customers, manual user research makes a resurgence to shape how the company or team drives GenAI to generate their products. *Studying Other Domains.* A source of inspiration is to study other systems, other domains, and even other disciplines. With extra time, and the pressure to come up with novel ideas all the time, software professionals engage continuously and deliberately with adjacent and other systems and fields.	*Loss of Intuition.* The nuanced, intuitive aspects of decision making that often pervade how teams ultimately make the right choices is undermined as teams grow accustomed to deferring to GenAI's data-driven suggestions that "must be right." *Loss of Trust with One Another.* Since interacting with GenAI causes software professionals to rarely get to work with colleagues anymore, they lose their ability to constructively disagree and resolve such disagreements, which is a key ingredient for the emergence of creative solutions. *Job Losses.* Multiple GenAIs, under the leadership of an overarching GenAI, successfully participate in creative idea generation and synthesis, obviating the need for human creative software teams and their practices. *Roteness of the Software Professional's Role.* The GenAI has taken over all creative duties and all that remains for the software professional is to fill in rote work that GenAI somehow cannot complete (e.g., documenting features, minimal programming to glue GenAI generated components together). *Kafkaesque Unaccountability.* The more creative processes rely on GenAI, the more anything GenAI cannot do (or cannot do well) appears impossible, in the same way today's administrators often say "the system won't let us do that." The world is populated with unknown systems that seemingly cannot be changed and for which no organization can be held accountable.

Taken collectively, the tetrads reveal many potential impacts for using GenAI for creativity in software development. Several impacts are contradictory, reflecting the differences among the scenarios, uncertainty as to which GenAI capabilities will emerge over time, and variability in how software professionals might respond to these capabilities. For example, within the creative process (Table 3), GenAI may negate the need for traditional design techniques such as focus groups as GenAI can determine users' needs and feedback through simulation. Software professionals could embrace this capability, since user research is difficult and time-consuming. Contrarily, there may be a resurgence in human-centered design techniques to complement the support afforded by GenAI, precisely because software professionals do not trust the results from GenAI and feel the need to be more closely connected with their user base for a more true understanding of their needs (and thus an improved ability to guide product development as compared to indiscriminately following GenAIs suggestions). Other potential impacts appear related, with similar impacts appearing in more than one tetrad. For example, concerns about the impacts of reduced human contributions to creativity are noted in the "reverses into" dimension within the Person, Process, and Press tetrads. While only some of the many impacts noted here may come to pass and, as mentioned previously, the extent to which some of them become reality might vary significantly depending on individual, team and setting, collectively they reveal the breadth of considerations that future research must consider.

5 Research Agenda

The four tetrads (Tables 1–4) articulate numerous potential impacts of using GenAI for creativity in software development. Each of them can lead to research questions (e.g., "How does GenAI

Table 4. McLuhan's Tetrad That Considers Potential Impacts of GenAI on Press (Environment) Creativity

ENHANCES *How does GenAI enhance the environmental conditions that promote creativity?*	**OBSOLESCES** *Which environmental conditions become obsolete due to GenAI?*
Discoverability of Contextual Information. GenAI helps surface contextual information by listening to conversations and continuously sharing relevant information (e.g., projecting potential design options on a wall). *Environmental Inspiration.* GenAI curates environmental settings (e.g., light, sound, visuals) to inspire creativity based on the task at hand (whether for an individual or a team working together), even dynamically adapting settings based on progress. *Team Assembly.* GenAI knows who has what skills, experiences, and knowledge, and thereby suggests the ideal teams for different creative tasks, making sure the right people are in the room. *Virtual Effectiveness.* Technology for working together remotely (e.g., Zoom, Virtual Reality meeting spaces) is enhanced with GenAI features that approach or even surpass the benefits of physical co-location, fostering creativity among distributed and/or hybrid teams. *Psychological Safety.* By creatively sparring with GenAI rather than colleagues or team leads, software professionals brainstorm more freely and toss out many more creative (and not-so-creative) ideas than they otherwise would. *Voice of Reason.* GenAI logically and rationally considers the wider impact of what is being asked of it and prevents software that brings harm to humans and other species from being developed and deployed. *Increased Risk Taking.* GenAI encourages organizations to take creative risks by decreasing the effort associated with releasing software changes and enabling instant pivots should issues arise.	*Whiteboards sketches, sticky notes, etc.* Traditional tools used in creative activities may no longer be needed, because software professionals simply tell GenAI their thoughts and it produces their virtual equivalents instantly. *Cubicles.* Traditional cubicles are finally abolished, as software development has become all about creativity, with no more rote tasks that require individual concentration and focus. *Stagnant Spaces.* Fixed layouts and functions of certain spaces (e.g., meeting rooms, lounge spaces) disappear in favor of spaces that dynamically adjust to facilitate different modes of creative work, particularly involving hybrid software professional and GenAI agent teams. *Offices.* With creativity the sole surviving skill, work can be performed anywhere, with no need for an office, whether at work or at home. The park is a great place to think. *Colleagues.* With GenAI support, individual software professionals design, create, and deploy any system the users ask for, alleviating the need for colleagues in the creative process, whether informally via typical chance encounters or formally through explicitly working as a team.
RETRIEVES *What environmental conditions could GenAI bring back to the foreground?*	**REVERSES INTO** *If overly relied upon, how could GenAI disrupt environmental conditions?*
Physical Movement. As GenAI takes over tasks that traditionally required being seated at a desk, there may be a revival of designing spaces that encourage physical movement and its relationship to creative thinking. *Supportive Management.* Often, creativity—especially everyday creativity—is not recognized as part of day-to-day business. By reinforcing creativity's role in achieving a competitive advantage, GenAI gives individuals more freedom to express their creativity and be recognized for it by management. Indeed, management may even proactively encourage individuals to build slack time in their schedules to promote creative thinking.	*Sabotage.* Software professionals are so enraged with their jobs being taken, they sabotage GenAI so that it under-performs and underwhelms, eventually becoming obsolete. *Isolation.* Individuals interact more with GenAI than with each other, stifling spontaneous human interactions that spark creativity and leading to isolation more generally. *Reduced Pride.* Software professionals feel they have less agency to be creative, as GenAI does it all for them. They lose motivation, pride, and enjoyment from their work as they can no longer express themselves creatively. They may quit or suffer poor mental well-being. *Management Pressure.* By making creativity the sole source of competitive advantage, GenAI increases pressure on software professionals to be creative, unintentionally undercutting performance as people tend to be less creative when under too much pressure. *Inability to Speak Up.* Management places so much trust in GenAI that individuals feel unable to question the GenAI's outputs. *Creativity–Whatever.* Because anyone can be creative with GenAI, software professionals are interchangeable and creativity is no longer appreciated as anything special, thus paradoxically becoming an undervalued skill. *Degradation of Nature.* Over-relying on GenAI produces enormous amounts of pollution (via electricity usage), further degrading the natural environment.

affect prototyping effort?" or hypotheses (e.g., adopting GenAI tool "X" decreases prototyping effort) that can be tested with an appropriate research method (e.g., quantitative data obtained from industry)—Steps 3 and 4 in the Disruptive Research Playbook [125]. Similarly, a laboratory experiment could help determine whether cognitive biases (e.g., design fixation) persist when using GenAI; an interpretivist case study of a software team could investigate the impact of GenAI on the creative process when they work on identifying new features for an existing product; or a longitudinal questionnaire could investigate changes in psychological safety and perceived management support for creativity after adopting GenAI, as posited in the press tetrad. Many more such studies could be designed for each of the identified impacts. All of them, too, will need to be replicated for different individuals, teams, products, and settings to understand more generally applicable impacts versus more contextualized impacts.

Space is insufficient to present the flood of research questions that would result from systematically generating hypotheses and research questions from the scenarios and tetrads. We instead organize our research agenda for creativity, GenAI, and software development along five high-level themes that we identified by grouping and refining broadly similar impacts across the four tetrads.

The first three themes cover three tangible, more immediate effects, namely: (1) individual capabilities, (2) team capabilities, and (3) products. The next two themes concern indirect, longer-term effects, namely: (4) unintended consequences and (5) societal impacts. For each theme, we offer several indicative and we feel particularly important research directions (not a comprehensive list) that cover a multitude of research questions. The latter two themes begin to converge on themes concerning many scholars from other domains—the broader impact of GenAI generally, and not just creativity, on all of society.

5.1 Theme 1: Individual Capabilities

Many of the identified impacts across the tetrads imply that creativity of the individual will increase, for two overarching predicted effects: (1) GenAI provides creativity support through idea generation, including cross-domain inspiration, as well as constructive critique on human-generated ideas, and (2) GenAI absorbs the mundane aspects of software development and creative work, thereby providing more time for software professionals to ponder, doodle, and come up with novel solutions. However, it is unclear if the individual will truly be more creative and in what aspects of their work (if at all). Various natural human responses may instead lead to less creative exploration (e.g., rubber stamping and loss of consideration of alternatives as identified in Table 1, "reverses into"), and necessary skills may deteriorate over time. At the most fundamental level, then, research is needed to study whether GenAI helps software professionals be more creative, and if so, for what tasks, how it helps, and whether creativity persists over time. Given that this can depend on how GenAI is made available to them and how it is integrated into their day-to-day tool usage, experimentation with new kinds of tools and studying emerging working practices will be equally important. Thus three important research directions are:

- *Understanding how software professionals incorporate GenAI into their creative work practices.* This almost goes without saying as the first area on which to focus, but building an understanding of how software professionals use GenAI in their creative work, for which tasks, and how individual differences such as personality may change their engagement provides an important baseline, identifying both what works well and does not work well. Strengths and weaknesses of GenAI when it comes to supporting, integrating with, and enhancing creative software development tasks can be identified. Studies should be repeated over time to understand changing patterns of adoption.
- *Designing new tools that help software professionals be creative when, where, and how so desired.* We already have seen integration of GenAI capabilities into commercial tools used in creative work (e.g., Miro [87]). Many creative tasks are not supported yet, however, and most tool support to date focuses on offering up solutions. More comprehensive tools that not only suggest solutions, but also offer critique, act as a creative sparring partner, and embed best practices of how to engage in creative work without inadvertently exhibiting bias are needed. Such tools can, and should, build upon the rich research into Creativity Support Tools [43].
- *Assessing whether individual software professionals actually engage in more creative behaviors with GenAI.* Simply adopting GenAI tools is not a guarantee that the creative behavior of software professionals is enhanced. Just as recent studies cast some doubt on the productivity effects of GenAI (e.g., [70]), and existing studies of GenAI for other types of creative work document mixed results (e.g., [22]), studies need to be performed that carefully assess the impact of GenAI on creativity in software development, perhaps using standard tests such as TTCT and AUT (see Section 3.3). The results may be surprising and potentially quite context dependent. Laboratory studies should not be entirely trusted, and complemented with in-the-field research.

5.2 Theme 2: Team Capabilities

While “team” does not appear as a tetrad, there is an implied impact on teams and their ability to be creative in a number of the tetrads, especially in the Process tetrad (Table 3) but also the Press tetrad (Table 4) as the environment surrounding a team can influence the creativity exhibited. Our concern here goes beyond multiple team members using GenAI individually; rather, the core question is whether GenAI interjected in teamwork enhances the team’s overall ability to be creative, and if so, how. This is a more complex research challenge, involving new tool development that leverages GenAI for innovative purposes, with functionality that can be invoked (e.g., mundane activities, rapid prototyping) or could be more autonomous in nature (e.g., moderation, contextual information). How teams respond to such functionality, whether they are comfortable working with or under the direction of GenAI, and if it makes a difference will equally need to be studied. Three key research directions in this space include:

- *Understanding how software teams incorporate GenAI into their creative work practices.* An individual adopting GenAI for their own work is different from a team that intensely works together, perhaps in a 24-hour hackathon to identify new product possibilities or maybe in a meeting diagnosing a major security risk in the code. Having participants use GenAI as part of such team settings may change team dynamics drastically, as well as how meetings and discussions proceed. Research is needed to study how individuals succeed and do not succeed bringing GenAI into team settings. Is it socially acceptable? Do the results of GenAI actually help the meeting and discussion along, or are they largely set aside? What are some best practices for team members to incorporate GenAI usage in different team activities?
- *Supporting hybrid human-GenAI team creativity.* One role for GenAI in creative teamwork is to remain passive: providing its results in response to human prompts, whether in being asked to contribute ideas, verify a design under construction, or help a programmer pair tackle a gnarly hardware constraint by making some creative suggestions to work around it. A different role would be for GenAI to be a proactive participant in the creative work, intervening when it sees fit. As we mentioned, such GenAI agents are already being explored in other creative field (e.g., [78]). Studies are needed to understand the potential of such agents or even teams of agents in software development creative work, how such agents must be scaffolded in meeting tools that are accepted by software professionals, and whether their use would lead to a higher level of creativity in the conversation than humans only.
- *Amplifying human team effectiveness.* Much is known about what distinguishes effective, high-performing teams from those that are less effective. Factors such as the diversity of perspectives brought by different team members [81], psychological safety to contribute nonsense ideas or make mistakes [137], and enabling everyone to meaningful participate [101] are just a few factors that have been shown to play an important role in improving the creativity of teams. Yet, few teams have the wherewithal and ability to consistently perform at a high creative level. GenAI that, rather than focusing on contributing thoughts tied to creative results, focus on understanding and steering creative activities instead represents an interesting opportunity. Exploring whether GenAI can help moderate and direct teams (or hybrid human-GenAI teams as in the previous point) towards especially creative workflows, practices, and conversations is an important albeit likely challenging endeavor.

5.3 Theme 3: The Product

Software professionals build products and, as the Product tetrad (Table 2) reveals, many potential impacts exist on the product and they are not all positive. On the one hand, by virtue of

GenAI identifying best practices (e.g., UI patterns, architectural models), retrieving old designs that have been forgotten but are applicable to the problem at hand, and offering a stream of novel ideas, the quality of the code underlying a product could increase and the resulting product could provide tailored UXs, leading to delighted customers. Additionally, because mundane tasks are offloaded to GenAI, software professionals can spend more time being creative and come up with more and more novel ideas. On the other hand, the professionals may well trust results from GenAI too much and not inspect them as they should, leading to poorer quality and products that are unsuccessful because they miss the mark. Longer term, GenAI usage may lead to homogenized products. Three primary potential research directions concerning creativity, GenAI, and the product are:

— *Understanding customer satisfaction with GenAI created software products.* Customers are the ultimate recipients of the product, with users and other stakeholders in the organization needing to be satisfied with their experiences with it. As such, changes in how products are conceived, be it in how entirely new products or new features are identified or in how they are designed and implemented, can significantly impact customer satisfaction. Assessing whether features designed with the help of GenAI, or even exclusively by GenAI, lead to a similar or different level of satisfaction are needed. This is perhaps especially true when products and features are heavily customized towards each individual user: Does that lead to better or worse UXs? Similarly, if product homogeneity occurs, does this lead to customer dissatisfaction because a lack of innovation, or does it not as maybe customers do not care or actually appreciate familiarity. It is important for these kinds of research questions to be answered with real users, since products that are too different to the norm due to rampant creativity may actually lead to dissatisfaction.
— *Assessing impact on long-term product sustainability.* Increased creativity in the functionality of a software system does not necessarily correlate to an underlying software architecture that can easily absorb a continuous stream of ever-evolving requests for innovative (and thus likely more invasive) features. As such, studies are needed that assess the longer-term impact of continuously adding creative new features on software's technical debt, whether a difference exists if those features are identified by GenAI but programmed by software professionals or if those features are automatically programmed by GenAI, and what it all means for the pace of development as it may ultimately become slower because of inextricably mounting technical debt.
— *Tracking the ecosystem.* Software is not developed in a vacuum. Indeed, a known phenomenon is copycats: When a unique new product or feature enters the ecosystem, copies of it in somewhat altered form follow (e.g., when Flappy Bird was released, many clones appeared in the appstore in the following months [74]). So it is in the commercial world of software: Software companies often clone their competitors outright or borrow features. GenAI relying on other software for inspiration is not necessarily a new phenomenon.[4] Its ability to do so near instantly, however, could have a serious impact on the ecosystem, especially if human creativity makes way for GenAI driven creativity, which as we discussed may lead to homogeneity. Understanding the evolving entire ecosystem of all software being produced and whether it as a whole becomes more creative (diverse, innovative) is therefore of importance. This, perhaps, could be studied on a domain-by-domain basis (e.g., games, finance).

[4]We speak to the ethics of GenAI relying on the actual source code written by others in Section 6.

5.4 Theme 4: Unintended Consequences

In focusing on the "obsolesces" and "reverses into" impacts across all four tetrads (Tables 1–4), there are unintended consequences to the Person, Product, Process, and Press from adopting GenAI for software development. Some examples include the potential for a reduction in mentorship from experienced colleagues when GenAI is the sparring partner, biases introduced more readily due to the reliance on GenAI being the creative, and a loss of office space (whether one's personal cubicle, creativity inducing spaces, or entire office buildings) where software professionals work as colleagues and develop important working relationships with one another. These unintended consequences may take time to play out, but we should begin thinking about them now because it will take time to create the measurement models and instruments needed to assess these effects [109]. Three important directions include:

- *Understanding GenAI models and their potential for propagating issues in software.* Much has already been said (e.g., [59]) about GenAI models only being as good as their training data and, thus, inheriting the flaws in the training data. While population bias has clearly been identified as a very significant issue more generally [95], an important issue for GenAI and creativity concerns the flaws that exist in the software that is collectively used as training data, as well as how that training data is ingested and processed before it turns into a model. Security concerns [135], accessibility biases [44], broken UIs, nefarious UIs, privacy-invading tracking code, and more may all be propagated when a software professional merely prompts GenAI to engineer a creative new application for them. With the code simply being generated, novelty "wins" at the expense of flawed systems. Detecting such flaws is difficult. Perhaps explainable AI might be an avenue of support, but if its vision does not come to fruition, an important line of research will be detecting problems in how models ingest and process systems that have known problems, so to avoid it becoming part of future applications. If, in turn, that research does not yield results, GenAI-induced creativity hits a ceiling, as it should then not be used to automatically generate software directly.
- *Evolving social networks.* Across the tetrads, a common theme is that of GenAI having an impact on human relations and particularly in removing opportunities for them to build such relationships, since software professionals can simply rely on GenAI to do the work for them and answer any questions they may have. They just do not have to talk with others anymore. While this represents a rather dark scenario, it is difficult to imagine a future in which GenAI infiltrating the creative tasks of software professionals does not impact their social network. Because human relations are ultimately the foundation for creative work today [105], an important research direction is to study software professionals' evolving social connections as GenAI gets adopted inside organizations.
- *Countering unintended consequences.* At times, unintended consequences that emerge, especially when they are negative, become a source of innovation. So it may be with GenAI, creativity, and software development. Consider the potential loss of consideration of alternatives when GenAI is used, because human nature favors the first presented example. This behavior could be countered through innovative wrappers around GenAI, where the wrapper presents the design alternatives or suggested code snippets non-linearly, with an analysis of strengths and weaknesses of each suggestion, and explicitly guides the software professional in making their choice or creatively remixing some of the alternatives. Indeed, such a tool may even be useful outside of the software profession, perhaps in GenAI-driven writing, or art, or product design. We thus advocate that the "obsolesces" and "reverses into" impacts are carefully studied by the community and that it invests time to understand how these phenomena could be countered.

5.5 Theme 5: Societal Impact

Products and people sit in society so it is important to understand how society is impacted by the use of GenAI for creativity in software development. While some of the impacts in the tetrads allude to potential societal impact (e.g., making the impossible possible in Table 2), the 4Ps framework itself does not necessarily consider society at large explicitly. One area to explore is the feelings of society on adopting new products created through the use of GenAI rather than human creativity, while another could look at the impact on the software profession. Maybe software professionals go the way of fellow artisans such as bakers and craft beer makers with a premium attached to unique, hand-made products or they go the way of gas-lighters and no longer relevant to society. Such research is likely to require multi-disciplinary experts such as psychologists, sociologists, and economists alongside ethnographic specialists. We feel that it is particularly important to monitor this aspect of the current wave of GenAI. Software is as likely as other disciplines to undergo significant transformations—at its own hand. Whether or not this is preferable from a societal point of view is the domain of policy and governance; to do so effectively requires a thorough understanding of what is transpiring, which in turn requires carefully constructed longitudinal studies. Indeed, governments are already enacting policies which effect the use of GenAI within society. For example, the European Artificial Intelligence Act [25] is designed to protect human rights and fundamental values. At this stage, it is unclear how such regulation will impact the use of GenAI, including for creative purposes in software development, but it will need to be considered as we move forward. Three pressing directions to pursue are:

— *Adopting GenAI-driven software*. Humans have historically exhibited a mixed collective response when it comes to adopting industrial and technological advances, as cheaper prices, instant availability, newness, and other properties of the "more advanced" products do not necessarily equate a desire to adopt [128]. Indeed, the world is full of inventions that did not go far (e.g., The Segway [24]). GenAI potentially disrupting the software industry by creatively generating all kinds of new software may lead to similar mixed reactions. People may not trust applications entirely envisioned and created by GenAI (c.f., lack of trust in self-driving cars [71]) or have privacy concerns [13] about the underlying GenAI models learning from their personal data stored in the GenAI created application. Others may just not care and adopt GenAI designed applications without consideration of whom created it. Insight into the reaction of society to GenAI created software can result in feedback loops as to how far GenAI should and should not extend in product development.
— *Changing software profession*. Software is built by humans and even the most fantastical scenarios still recognize that humans will be required [116]. That said, as we highlighted in the tetrads, the jobs of software professionals may change some, quite a bit, dramatically, or even completely in response to how GenAI evolves. It is a question of whether software professionals will be open to these changes. Some are likely to embrace getting rid of mundane work and engaging in creative work all the time. Others may see themselves not as creatives, and even with the help of GenAI in their creative tasks feel threatened. Alternatively, perhaps the profession will seem more accessible to people without the traditional technical skills as GenAI automates the technical work. It is important to set up studies that monitor how individual software professionals feel about the use of GenAI for creativity and how it impacts their sense of value and purpose, job satisfaction, emotional well-being, and other such human factors—now and in the years to come. Examining software teams' composition, including roles, the skills needed, and nature of software work, to see how all this changes with GenAI is also key to understanding the educational needs for students and work training required. Furthermore, such research can inform the development of educational programs for retraining

current software professionals who may need to shift from being a traditional programmer to becoming a "creativity prompt specialist" using GenAI for a variety of creative purposes in the newly minted GenAI driven software development process.

— *Changing software industry.* While this article is about the impact of GenAI on creativity in software development, we would be remiss if we did not also consider the broader impact of GenAI and how it is shaping the software industry. If aspects of the most far-reaching scenarios become reality, alongside other predicted changes to software work (e.g., increased productivity) it might mean that the industry will need far fewer software professionals and that successful companies may effectively be one-person shops. At the extreme, end-user programming [11] may become "end user prompting" with instant apps on demand created by anyone at any time, ruling out a software industry altogether, except for the few companies controlling the GenAI infrastructure. An alternative future, however, may see the industry continuing to grow. Despite teams being somewhat smaller than before (savings are not as expected because creatively prompting GenAI to produce precisely the desired software features turns out to be a challenging task), more teams are needed to address surging demand. Overall, the industry will reshape itself in some way, shape, or form, voluntarily, driven by market forces, or spurred by developer rebellion (sabotage is not an uncommon response to workplace automation [92]). In this context, it is crucial to anticipate what kinds of changes may occur and to identify forces and measurement instruments for tracking them.

6 Discussion

GenAI is rapidly disrupting software development. Unsurprisingly, then, many researchers in the domain of **Artificial Intelligence for Software Engineering (AI4SE)** [76] are considering the impacts of AI on different aspects of software development, such as requirements gathering, coding, testing [36, 54], and wider considerations such as individual productivity [14]. We believe creativity is integral to software development and also warrants inclusion within the AI4SE remit. Through the use of diverse and intentionally thought provoking scenarios and the McLuhan tetrad as applied to the 4Ps of creativity, we have contributed a wide-ranging set of potential impacts GenAI may have on creativity within software development, leading to the identification of five research themes: individuals, teams, product, unintended consequences, and society. We feel all are equally important to study but note some are more complex than others to study, usually due to the data and timelines required. While studies involving individuals, teams, products, and early insights into potential unintended consequences can begin now, and we encourage researchers to do so, impact on society and also additional, longer-term unintended consequences (the unknown unknowns) are likely to be identified in retrospect so must wait.

The outcomes from the numerous research studies will no doubt include insights into both the social and the technical. The ways people work, including new ones, how people feel, and the way they use tools (or need new ones) are just some likely findings to emerge from future studies. Social implications will be keenly understood particularly as the unintended consequences and societal impact are studied. In some ways, these social implications are the most important to understand as they inform researchers and practitioners who design and build GenAI of the potential harms caused by the use of GenAI for creative software work, in much the same way software professionals are only now grappling with the issues caused by biases in LLMs [69].

We believe addressing our research agenda can help in both understanding the phenomenon of GenAI-driven creativity in software development as it plays out over time and shaping it. We feel the latter is particularly important, because with the many potential impacts captured in the tetrads, it becomes possible to design a future with eyes fully open to the potential positive and negative impacts that GenAI may have on many facets of creativity in software development, and ultimately society. Consider support tools leveraging GenAI for a variety of creative software tasks. While these may augment software professionals in their creative problem solving, at the extreme they may also replace human creative problem solving, leading to demoralized, demotivated, and depressed software professionals. These potential harms should be considered when designing such tools and should be mitigated by consistently adopting Human-Centered AI design approaches (as advocated by HCI researchers who emphasize AI should augment humans and not replace them [121]). This does not mean creative software development tasks cannot evolve in how they are being performed; it just means that the potential upsides *and* downsides must be considered by researchers and developers of GenAI-driven software creativity tools. It behooves all researchers to adopt such a stance as GenAI usage continues to grow within the software industry and if we are to contribute to a society in which human endeavor and creativity is valued. Ultimately, this starts with the software industry itself, since in many ways it is at the leading edge in leveraging GenAI for its own purpose—designing and developing creative and useful software.

In identifying the potential impacts and an associated research agenda, we note two limitations related to using potential future scenarios and the McLuhan tetrad to identify potential impacts of the use of GenAI on creativity in software development. First, the scenarios and tetrad merely provide a framework for identifying future possibilities, with a fair degree of imagination and brainstorming required to articulate the scenarios and populate the tetrad (or in our case, tetrads). Different groups, thus, may produce scenarios, questions, and impacts different from the authors of this article. Indeed, we do not consider our tetrads exhaustive and encourage others to further augment our ideas with additional research questions within the five research themes. Moreover, given the breakneck speed at which GenAI is progressing, some of the identified potential impacts may already come to pass and may well need to be studied after the event, rather than during.

Secondly, using the 4P creativity framework and the tetrad with its focus on how a technology can enhance, obsolesce, retrieve, and reverse into phenomena does not directly encourage consideration of the ethical, legal, and sustainability issues increasingly associated with GenAI (e.g., [10, 100, 106, 120]). We believe such important issues are applicable to each of the five research themes and should be incorporated whenever researching one of the themes. In considering ethics, GenAI lacks ethical judgment and contextual understanding, so without oversight GenAI can produce biased, inappropriate, or unethical results due to flawed training data or algorithms. Thus oversight over AI is required [10]. This oversight could conflict with the use of autonomous agents and their role in identifying new features or classes of products. One legal issue concerns ownership [10]. GenAI can blur the line between human-generated and machine-assisted work. This gives rise to the questions of authorship. If a developer uses GenAI to create a novel design, how much of the product is truly theirs? This ambiguity extends to intellectual property: Does ownership rest with the GenAI user, the model creators, or should it be in the public domain? This legal point should be considered whenever researching the use of GenAI to generate novel ideas, solutions, products, or services. Sustainability too needs to be considered. GenAI requires huge amounts of environmental resources leading to depletion of the natural environment [120], which clearly impacts society negatively for all GenAI use, including the subject of this article—creativity in software development. Considering this excessive energy consumption could result in tradeoffs in deciding whether to use GenAI for individual creativity or products.

7 Conclusion

No matter how GenAI evolves, software professionals are already using it to help in their creative design work, in their problem-solving when facing coding issues, and more. They look for it to get them started rather than staring at a blank sheet of paper, to nudge them out of fixations, to generate potential root causes for a bug, and so on. In this article, we examined this relationship between creativity and GenAI by (1) speculating on potential future scenarios of how GenAI may impact creativity in software engineering, alongside (2) McLuhan's tetrad to hypothesize how GenAI might impact four different components of creativity (the Person, the Product, the Process, and the Press (environment)). We used these potential impacts to derive a research agenda comprising five connected themes that consider how individual capabilities, team capabilities, the product, unintended consequences, and society can be affected.

We encourage researchers to consider the potential impacts and research themes outlined in this article, to augment them further by adding their own thoughts and projections, and to join us in researching creativity, GenAI, and software development so we can have a constructive collective voice in shaping the phenomenon.

Acknowledgments

We express our gratitude to the Alfred P. Sloan Foundation and the Carlsberg Foundation for their essential support of the 1st Copenhagen Symposium on Human-Centered Software Engineering AI. It was during this symposium that the foundational discussions for this article were initiated. Moreover, we are grateful to the organizers and fellow participants of the *SE 2030* workshops held at *FSE 2024* for fruitful and enjoyable discussions on the future of software engineering with various insights being incorporated into this article.

References

[1] Selcuk Acar, Harun Tadik, Danielle Myers, Carian Van der Sman, and Recep Uysal. 2021. Creativity and well-being: A meta-analysis. *The Journal of Creative Behavior* 55, 3 (2021), 738–751. DOI: https://doi.org/10.1002/jocb.485

[2] Teresa M. Amabile. 1996. *Creativity in Context* (2nd. ed.). Westview Press, Inc., Boulder, Colorado.

[3] Teresa M. Amabile. 2017. In pursuit of everyday creativity. *The Journal of Creative Behavior* 51, 4 (2017), 335–337. DOI: https://doi.org/10.1002/jocb.200

[4] Teresa M. Amabile. 1988. A model of creativity and innovation in organizations. *Research in Organizational Behavior* 10, 1 (1988), 123–167.

[5] Teresa M. Amabile and Steven Kramer. 2011. *The Progress Principle: Using Small Wins to Ignite Joy, Engagement, and Creativity at Work*. Harvard Business Review Press.

[6] Aamir Amin, Shuib Basri, Mohd Fadzil Hassan, and Mobashar Rehman. 2018. A snapshot of 26 years of research on creativity in software engineering-A systematic literature review. In *International Conference on Mobile and Wireless Technologies (ICMWT)*, 430–438.

[7] Aamir Amin, Shuib Basri, Mobashar Rehman, Luiz Fernando Capretz, Rehan Akbar, Abdul Rehman Gilal, and Muhammad Farooq Shabbir. 2020. The impact of personality traits and knowledge collection behavior on programmer creativity. *Information and Software Technology* 128 (2020), 106405.

[8] Barrett R. Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization effects of large language models on human creative ideation. In *16th Conference on Creativity & Cognition*, 413–425.

[9] Neil Anderson, Kristina Potočnik, and Jing Zhou. 2014. Innovation and creativity in organizations: A state-of-the-science review, prospective commentary, and guiding framework. *Journal of Management* 40, 5 (2014), 1297–1333. DOI: https://doi.org/10.1177/0149206314527128

[10] Marcellin Atemkeng, Sisipho Hamlomo, Brian Welman, Nicole Oyentunji, Pouya Ataei, and Jean Louis K. E. Fendji. 2024. Ethics of software programming with generative AI: Is programming without generative AI always radical? arXiv:2408.10554. Retrieved from https://arxiv.org/abs/2408.10554

[11] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software* 149 (2019), 101–137. DOI: https://doi.org/10.1016/j.jss.2018.11.041

[12] Nina Begus. 2023. Experimental narratives: A comparison of human crowdsourced storytelling and AI storytelling. arXiv:2310.12902. Retrieved from https://arxiv.org/abs/2310.12902

[13] Marco Antonio Beltran, Marina Ivette Ruiz Mondragon, and Seung Hun Han. 2024. Comparative analysis of generative AI risks in the public sector. In *25th Annual International Conference on Digital Government Research (dg.o '24)*. ACM, New York, NY, 610–617. DOI: https://doi.org/10.1145/3657054.3657125

[14] Christian Bird, Denae Ford, Thomas Zimmermann, Nicole Forsgren, Eirini Kalliamvakou, Travis Lowdermilk, and Idan Gazit. 2023. Taking flight with copilot: Early insights and opportunities of AI-powered pair-programming tools. *Queue* 20, 6 (Jan. 2023), 35–57. DOI: https://doi.org/10.1145/3582083

[15] Birgitta Bockeler and Ryan Murray. 2023. Generative AI and the Software Development Lifecycle: Much More Than Coding Assistance. Retrieved from https://www.thoughtworks.com/en-us/insights/articles/generative-ai-software-development-lifecycle-more-than-coding-assistance

[16] Margaret A. Boden. 2004. *The Creative Mind: Myths and Mechanisms*. Psychology Press, London, UK.

[17] Noah Bohren, Rustamdjan Hakimov, and Rafael Lalive. 2024. *Creative and Strategic Capacities of Generative AI: Evidence from Large-Scale Experiments*. Technical Report. Institute of Labor Economics (IZA).

[18] Nanna Borum, Eva Petersson Brooks, and Søren R. Frimodt-Møller. 2014. The resilience of analog tools in creative work practices: A case study of LEGO future Lab's team in Billund. In *Human-Computer Interaction. Theories, Methods, and Tools.* Masaaki Kurosu (Ed.), Springer International Publishing, Cham, 23–34.

[19] Sebastian G. Bouschery, Vera Blazevic, and Frank T. Piller. 2023. Augmenting human innovation teams with artificial intelligence: Exploring transformer-based language models. *Journal of Product Innovation Management* 40, 2 (2023), 139–153.

[20] Léonard Boussioux, Jacqueline N. Lane, Miaomiao Zhang, Vladimir Jacimovic, and Karim R. Lakhani. 2024. The crowdless future? Generative AI and creative problem-solving. *Organization Science* 5 (2024), 1589–1607.

[21] D. Brady. 2024. *How Generative AI Is Changing the Way Developers Work - The GitHub Blog*. GitHub. Retrieved from https://github.blog/ai-and-ml/generative-ai/how-generative-ai-is-changing-the-way-developers-work/

[22] Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024. Art or artifice? Large language models and the false promise of Creativity. In *CHI Conference on Human Factors in Computing Systems.* ACM, 1–34.

[23] DaEun Choi, Sumin Hong, Jeongeon Park, John Joon Young Chung, and Juho Kim. 2024. CreativeConnect: Supporting reference recombination for graphic design ideation with generative AI. In *CHI Conference on Human Factors in Computing Systems.* ACM, 1–25.

[24] Andrew V. Clark, Carol Atkinson-Palombo, and Norman W. Garrick. 2019. The rise and fall of the segway: Lessons for the social adoption of future transportation. *Transfers* 9, 2 (2019), 27–44.

[25] European Commission. 2024. AI Act — Shaping Europe's Digital Future. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

[26] Mihaly Csikszentmihalyi. 2014. Society, culture, and person: A systems view of creativity. In *The Systems Model of Creativity: The Collected Works of Mihaly Csikszentmihalyi*. Mihaly Csikszentmihalyi (Ed.), Springer, Netherlands, Dordrecht, 47–61. DOI: https://doi.org/10.1007/978-94-017-9085-7_4

[27] Fabio Q. B. da Silva, Cleviton V. F. Monteiro, Igor Ebrahim dos Santos, and Luiz Fernando Capretz. 2016. How software development group leaders influence team members' innovative behavior. *IEEE Software* 33, 5 (Sept. 2016), 106–109. DOI: https://doi.org/10.1109/MS.2016.120

[28] Nicole Davila, Jorge Melegati, and Igor Wiese. 2024. Tales from the trenches: Expectations and challenges from practice for code review in the generative AI era. *IEEE Software* 6 (2024), 1–8. DOI: https://doi.org/10.1109/MS.2024.3428439

[29] Joshua Davis, Liesbet Van Bulck, Brigitte N. Durieux, and Charlotta Lindvall. 2024. The temperature feature of ChatGPT: Modifying creativity for clinical research. *JMIR Human Factors* 11, 1 (2024), e53559.

[30] Fabrizio Dell'Acqua, Edward McFowland III, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. 2023. Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper* 24–013 (2023).

[31] Anil R. Doshi and Oliver Hauser. 2023. Generative artificial intelligence enhances creativity. *Science Advances* 10, 28 (2023), eadn5290.

[32] Anthony Dunne and Fiona Raby. 2024. *Speculative Everything*. MIT Press. Retrieved from https://mitpress.mit.edu/9780262548687/speculative-everything/

[33] C. Ebert and P. Louridas. 2023. Generative AI for software practitioners. *IEEE Software* 40, 04 (Jul. 2023), 30–38. DOI: https://doi.org/10.1109/MS.2023.3265877

[34] Amy C. Edmondson and Zhike Lei. 2014. Psychological safety: The history, renaissance, and future of an interpersonal construct. *Annual Review of Organizational Psychology and Organizational Behavior* 1, 1 (2014), 23–43.

[35] Nafise Eskandani and Guido Salvaneschi. 2024. Towards AI for software systems. In *1st ACM International Conference on AI-Powered Software (AIware '24)*. ACM, New York, NY, 79–84. DOI: https://doi.org/10.1145/3664646.3664767

[36] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. 2023. Large language models for software engineering: Survey and open problems. arXiv:2310.03533. Retrieved from https://arxiv.org/abs/2310.03533

[37] Hongbo Fang, James Herbsleb, and Bogdan Vasilescu. 2024. Novelty begets popularity, but curbs participation – A macroscopic view of the Python open-source ecosystem. In *International Conference on Software Engineering (ICSE '24)*. ACM, New York, NY, Article 55, 11 pages.

[38] Gilles Fauconnier and Mark Turner. 1998. Conceptual integration networks. *Cognitive Science* 22, 2 (1998), 133–187.

[39] World Economic Forum. 2023. *The Future of Jobs Report 2023*. World Economic Forum. Retrieved from https://www.weforum.org/publications/the-future-of-jobs-report-2023/

[40] Giorgio Franceschelli and Mirco Musolesi. 2024. On the creativity of large language models. *AI & SOCIETY* (2024), 1–11. Retrieved from https://arxiv.org/abs/arXiv:2304.00008

[41] Stan Franklin and Art Graesser. 1997. Is it an agent, or just a program? A taxonomy for autonomous agents. In *Intelligent Agents III Agent Theories, Architectures, and Languages*. Jörg P. Müller, Nicholas R. Wooldridge, and Michael J. Jennings (Eds.), Springer, Berlin, 21–35.

[42] César França, Fabio Q. B. Silva, and Helen Sharp. 2022. Motivation and satisfaction of software engineers. *IEEE Transactions on Software Engineering* 46, 2 (2022), 118–140. DOI: https://doi.org/10.1109/TSE.2018.2842201

[43] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the landscape of creativity support tools in HCI. In *2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, 1–18. DOI: https://doi.org/10.1145/3290605.3300619

[44] Vinitha Gadiraju, Shaun Kane, Sunipa Dev, Alex Taylor, Ding Wang, Emily Denton, and Robin Brewer. 2023. "I wouldn't say offensive but...": Disability-centered perspectives on large language models. In *2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23)*. ACM, New York, NY, 205–216. DOI: https://doi.org/10.1145/3593013.3593989

[45] Robert L. Glass. 1994. *Software Creativity*. Prentice-Hall, Inc., Atlanta.

[46] Öznur Göçmen and Hamit Coşkun. 2019. The effects of the six thinking hats and speed on creativity in brainstorming. *Thinking Skills and Creativity* 31 (2019), 284–295.

[47] Cristian Granados and Montserrat Pareja-Eastaway. 2019. How do collaborative practices contribute to innovation in large organisations? The case of hackathons. *Innovation* 21, 4 (Oct. 2019), 487–505. DOI: https://doi.org/10.1080/14479338.2019.1585190

[48] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. 2018. The dark (patterns) side of UX design. In *2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, 1–14. DOI: https://doi.org/10.1145/3173574.3174108

[49] Luca Grilli and Mattia Pedota. 2024. Creativity and artificial intelligence: A multilevel perspective. *Creativity and Innovation Management* 33, 2 (2024), 234–247.

[50] Wouter Groeneveld, Hans Jacobs, Joost Vennekens, and Kris Aerts. 2020. Non-cognitive abilities of exceptional software engineers: A Delphi study. In *51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. ACM, 1096–1102. DOI: https://doi.org/10.1145/3328778.3366811

[51] Wouter Groeneveld, Laurens Luyten, Joost Vennekens, and Kris Aerts. 2021. Exploring the role of creativity in software engineering. In *International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 1–9.

[52] Erik E. Guzik, Christian Byrge, and Christian Gilde. 2023. The originality of machines: AI takes the Torrance test. *Journal of Creativity* 33, 3 (2023), 100065.

[53] Martin Hoegl and K. Praveen Parboteeah. 2007. Creativity in innovative projects: How teamwork matters. *Journal of Engineering and Technology Management* 24, 1 (Mar. 2007), 148–166. DOI: https://doi.org/10.1016/j.jengtecman.2007.01.008

[54] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large language models for software engineering: A systematic literature review. arXiv:2308.10620. Retrieved from https://arxiv.org/abs/2308.10620

[55] Sarah Inman, Sarah D'Angelo, and Bogdan Vasilescu. 2024. Developer productivity for humans, part 8: Creativity in software engineering. *IEEE Software* 41, 2 (2024), 11–16.

[56] Zorana Ivcevic and Mike Grandinetti. 2024. Artificial intelligence as a tool for creativity. *Journal of Creativity* 34, 2 (2024), 100079.

[57] Victoria Jackson, Rafael Prikladnicki, André van der Hoek, and Lisa Marshall. 2022. Team creativity in a hybrid software development world: Eight approaches. *IEEE Software* 40, 2 (2022), 60–69.

[58] Rodi Jolak, Andreas Wortmann, Grischa Liebel, Eric Umuhoza, and Michel R. V. Chaudron. 2023. Design thinking and creativity of colocated versus globally distributed software developers. *Journal of Software: Evolution and Process* 35, 5 (2023), e2377.

[59] Uday Kamath, Kevin Keenan, Garrett Somers, and Sarah Sorenson. 2024. *LLM Challenges and Solutions*. Springer Nature, Switzerland, Cham, 219–274. DOI: https://doi.org/10.1007/978-3-031-65647-7_6

[60] Sungmin Kang, Gabin An, and Shin Yoo. 2024. A quantitative and qualitative evaluation of LLM-based explainable fault localization. *Proceedings of the ACM on Software Engineering* 1, FSE, Article 64 (Jul. 2024), 23 pages. DOI: https://doi.org/10.1145/3660771

[61] James C. Kaufman and Ronald A. Beghetto. 2009. Beyond big and little: The four C model of creativity. *Review of General Psychology* 13, 1 (Mar. 2009), 1–12. DOI: https://doi.org/10.1037/a0013688

[62] Yared H. Kidane and Peter A. Gloor. 2007. Correlating temporal communication patterns of the eclipse open source community with performance and creativity. *Computational and Mathematical Organization Theory* 13 (2007), 17–27.

[63] Jingoog Kim and Mary Lou Maher. 2023. The effect of AI-based inspiration on human design ideation. *International Journal of Design Creativity and Innovation* 11, 2 (2023), 81–98.

[64] Jeongyeon Kim, Sangho Suh, Lydia B. Chilton, and Haijun Xia. 2023. Metaphorian: Leveraging large language models to support extended metaphor creation for science writing. In *2023 ACM Designing Interactive Systems Conference.* ACM, 115–135.

[65] Kyung Hee Kim. 2006. Can we trust creativity tests? A review of the Torrance tests of creative thinking (TTCT). *Creativity Research Journal* 18, 1 (Jan. 2006), 3–14. DOI: https://doi.org/10.1207/s15326934crj1801_2

[66] Rochelle King, Elizabeth F. Churchill, and Caitlin Tan. 2017. *Designing with Data: Improving the User Experience with A/B Testing*. O'Reilly Media, Inc.

[67] Rakesh Kochhar. 2023. Which U.S. Workers Are More Exposed to AI on Their Jobs? Retrieved from https://www.pewresearch.org/social-trends/2023/07/26/which-u-s-workers-are-more-exposed-to-ai-on-their-jobs/

[68] Mika Koivisto and Simone Grassini. 2023. Best humans still outperform artificial intelligence in a creative divergent thinking task. *Scientific Reports* 13, 1 (2023), 13601.

[69] Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *ACM Collective Intelligence Conference (CI '23).* ACM, New York, NY, 12–24. DOI: https://doi.org/10.1145/3582269.3615599

[70] Uplevel Data Labs. 2024. Gen AI for Coding Research Report—Uplevel Data Labs. Retrieved from https://resources.uplevelteam.com/gen-ai-for-coding

[71] John D. Lee and Kristin Kolodge. 2020. Exploring trust in self-driving vehicles through text analysis. *Human Factors* 62, 2 (2020), 260–277. DOI: https://doi.org/10.1177/0018720819872672

[72] Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-AI collaborative writing dataset for exploring language model capabilities. In *2022 CHI Conference on Human Factors in Computing Systems.* ACM, 1–19.

[73] Paul Luo Li, Amy J. Ko, and Jiamin Zhu. 2015. What makes a great software engineer? In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 700–710. DOI: https://doi.org/10.1109/ICSE.2015.335

[74] Soo Ling Lim, Peter J. Bentley, and Fuyuki Ishikawa. 2016. The effects of developer dynamics on fitness in an evolutionary ecosystem model of the app store. *IEEE Transactions on Evolutionary Computation* 20, 4 (2016), 529–545. DOI: https://doi.org/10.1109/TEVC.2015.2494382

[75] Yiren Liu, Pranav Sharma, Mehul Jitendra Oswal, Haijun Xia, and Yun Huang. 2024. PersonaFlow: Boosting research ideation with LLM-simulated expert personas. arXiv:2409.12538. Retrieved from https://arxiv.org/abs/2409.12538

[76] David Lo. 2023. Trustworthy and synergistic artificial intelligence for software engineering: Vision and roadmaps. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE).* IEEE, 69–85. DOI: https://doi.org/10.1109/ICSE-FoSE59343.2023.00010

[77] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. 2020. Novice-AI music co-creation via AI-steering tools for deep generative models. In *2020 CHI Conference on Human Factors in Computing Systems.* ACM, 1–13.

[78] Li-Chun Lu, Shou-Jen Chen, Tsung-Min Pai, Chan-Hung Yu, Hung-yi Lee, and Shao-Hua Sun. 2024. LLM discussion: Enhancing the creativity of large language models via discussion framework and role-play. arXiv:2405.06373. Retrieved from https://arxiv.org/abs/2405.06373

[79] Egan Lua, Dong Liu, and Christina E. Shalley. 2024. Multilevel outcomes of creativity in organizations: An integrative review and agenda for future research. *Journal of Organizational Behavior* 45, 2 (2024), 209–233. DOI: https://doi.org/10.1002/job.2690

[80] N. Maiden, A. Gizikis, and S. Robertson. 2004. Provoking creativity: Imagine what your requirements could be like. *IEEE Software* 21, 5 (Sept. 2004), 68–75. DOI: https://doi.org/10.1109/MS.2004.1331305

[81] Elizabeth Mannix and Margaret A. Neale. 2005. What differences make a difference? The promise and reality of diverse teams in organizations. *Psychological Science in the Public Interest* 6, 2 (2005), 31–55. DOI: https://doi.org/10.1111/j.1529-1006.2005.00022.x

[82] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2024. Prompt engineering in large language models. In *Data Intelligence and Cognitive Informatics*. I. Jeena Jacob, Selwyn Piramuthu, and Przemyslaw Falkowski-Gilski (Eds.), Springer Nature, Singapore, 387–402.

[83] Jack McGuire, David De Cremer, and Tim Van de Cruys. 2024. Establishing the importance of co-creation and self-efficacy in creative collaboration with artificial intelligence. *Scientific Reports* 14, 1 (2024), 18525.

[84] Mckinsey. 2023. *Unleashing Developer Productivity with Generative AI*. Mckinsey. Retrieved from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai

[85] Marshall McLuhan. 1977. Laws of the media. *ETC: A Review of General Semantics* 34 (1977), 173–179.

[86] Brian Merchant. 2023. *Blood in the Machine: The Origins of the Rebellion against Big Tech* (1st. ed.). Little, Brown and Company.

[87] Miro. 2024. Miro AI — Miro. Retrieved from https://miro.com/ai/

[88] Piotr Mirowski, Juliette Love, Kory Mathewson, and Shakir Mohamed. 2024. A robot walks into a bar: Can language models serve as creativity SupportTools for comedy? An evaluation of LLMs' humour alignment with comedians. In *the 2024 ACM Conference on Fairness, Accountability, and Transparency*. ACM, 1622–1636.

[89] Ingrid Kajzer Mitchell and Jennifer Walinga. 2017. The creative imperative: The role of creativity, creative problem solving and insight as key drivers for sustainability. *Journal of Cleaner Production* 140 (2017), 1872–1884.

[90] Rahul Mohanani, Paul Ralph, Burak Turhan, and Vladimir Mandić. 2021. How templated requirements specifications inhibit creativity in software engineering. *IEEE Transactions on Software Engineering* 48, 10 (2021), 4074–4086.

[91] Rahul Mohanani, Prabhat Ram, Ahmed Lasisi, Paul Ralph, and Burak Turhan. 2017. Perceptions of creativity in software engineering research and practice. In *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 210–217.

[92] Gavin Mueller. 2021. *Breaking Things at Work* (1st. ed.). Verso, London, UK.

[93] Michael D. Mumford, Kimberly S. Hester, and Issac C. Robledo. 2012. Chapter 1 - Creativity in organizations: Importance and approaches. In *Handbook of Organizational Creativity*. Michael D. Mumford (Ed.), Academic Press, 3–16. DOI: https://doi.org/10.1016/B978-0-12-374714-3.00001-X

[94] Surabhi S. Nath, Peter Dayan, and Claire Stevenson. 2024. Characterising the creative process in humans and large language models. In *International Conference on Computational Creativity (ICCC)*.

[95] Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: Origins, inventory, and discussion. *Journal of Data and Information Quality* 15, 2, Article 10 (Jun. 2023), 21 pages. DOI: https://doi.org/10.1145/3597307

[96] Anh Nguyen-Duc, Beatriz Cabrero-Daniel, Adam Przybylek, Chetan Arora, Dron Khanna, Tomas Herda, Usman Rafiq, Jorge Melegati, Eduardo Guerra, Kai-Kristian Kemell, et al. 2023. Generative artificial intelligence for software engineering – A research agenda. arXiv:2310.18648. Retrieved from https://arxiv.org/abs/2310.18648

[97] Tim O'Reilly. 1999. Lessons from open-source software development. *Communications of the ACM* 42, 4 (1999), 32–37.

[98] William Orwig, Emma R. Edenbaum, Joshua D. Greene, and Daniel L. Schacter. 2024. The language of creativity: Evidence from humans and large language models. *The Journal of Creative Behavior* 58, 1 (2024), 128–136.

[99] Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? arXiv:2309.05196. Retrieved from https://arxiv.org/abs/2309.05196

[100] Jenny Eriksson Lundström, Patrick Mikalef, Kieran Conboy and Aleš Popovič. 2022. Thinking responsibly about responsible AI and "the dark side" of AI. *European Journal of Information Systems* 31, 3 (2022), 257–268. DOI: https://doi.org/10.1080/0960085X.2022.2026621

[101] Paul B. Paulus and Vincent R. Brown. 2007. Toward more creative and innovative group idea generation: A cognitive-social-motivational perspective of brainstorming. 1, 1 (2007), 248–265. DOI: https://doi.org/10.1111/j.1751-9004.2007.00006.x

[102] Paul B. Paulus and Huei-Chuan Yang. 2000. Idea generation in groups: A basis for creativity in organizations. *Organizational Behavior and Human Decision Processes* 82, 1 (2000), 76–87. DOI: https://doi.org/10.1006/obhd.2000.2888

[103] Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? In *International Conference on Computational Creativity (CC)*.

[104] Guilherme Pereira, Rafael Prikladnicki, Victoria Jackson, André van der Hoek, Luciane Fortes, and Igor Macaubas. 2024. Early results from a study of GenAI adoption in a large Brazilian company: The case of Globo. In *Generative AI for Effective Software Development*. Anh Nguyen-Duc, Pekka Abrahamsson, and Foutse Khomh (Eds.), Springer, 275–293.

[105] Jill E. Perry-Smith and Pier Vittorio Mannucci. 2017. From creativity to innovation: The social network drivers of the four phases of the idea journey. *Academy of Management Review* 42, 1 (2017), 53–79.

[106] Alexander Poth, Anna Wildegger, and Dan-Alexander Levien. 2024. Considerations about integration of GenAI into products and services from an ethical and legal perspective. In *Systems, Software and Services Process Improvement*. Murat Yilmaz, Paul Clarke, Andreas Riel, Richard Messnarz, Christian Greiner, and Thomas Peisl (Eds.), Springer Nature, Switzerland, Cham, 155–171.

[107] Gerard Puccio and Matei Schwartz. 2023. Chapter 4 - Outcomes of creativity in organizations: From organization to human-centered benefits. In *Handbook of Organizational Creativity* (2nd. ed.). Roni Reiter-Palmon and Sam Hunter (Eds.), Academic Press, 37–50. DOI: https://doi.org/10.1016/B978-0-323-91840-4.00004-9

[108] Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S. Weld. 2024. Scideator: Human-LLM scientific idea generation grounded in research-paper facet recombination. arXiv:2409.14634. Retrieved from https://arxiv.org/abs/2409.14634

[109] Paul Ralph, Miikka Kuutila, Hera Arif, and Bimpe Ayoola. 2024. *Teaching Software Metrology: The Science of Measurement for Software Engineering*. Springer. Retrieved from https://arxiv.org/abs/2406.14494

[110] Mel Rhodes. 1961. An analysis of creativity. *The Phi Delta Kappan* 42, 7 (1961), 305–310.

[111] Mark A. Runco and Garrett J. Jaeger. 2012. The standard definition of creativity. *Creativity Research Journal* 24, 1 (Jan. 2012), 92–96. DOI: https://doi.org/10.1080/10400419.2012.650092

[112] Mark A. Runco, Aubra Shepard, and Harun Tadik. 2022. How much creative potential is expressed at work? *Journal of Creativity* 32, 1 (2022), 100016. DOI: https://doi.org/10.1016/j.yjoc.2021.100016

[113] Daniel Russo. 2024. Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology* 33, 5, Article 135 (Jun. 2024), 50 pages. DOI: https://doi.org/10.1145/3652154

[114] Daniel Russo, Sebastian Baltes, Niels van Berkel, Paris Avgeriou, Fabio Calefato, Beatriz Cabrero-Daniel, Gemma Catolino, Jürgen Cito, Neil Ernst, Thomas Fritz, et al. 2024. Generative AI in software engineering must be human-centered: The Copenhagen manifesto. *Journal of Systems and Software* 216 (2024), 112115. DOI: https://doi.org/10.1016/j.jss.2024.112115

[115] Robson Santos, Italo Santos, Cleyton Magalhaes, and Ronnie de Souza Santos. 2024. Are we testing or being tested? Exploring the practical applications of large language models in software testing. In *2024 IEEE Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 353–360. DOI: https://doi.org/10.1109/ICST60714.2024.00039

[116] Ian Scheffler. 2023. GitHub CEO Says Copilot Will Write 80% of Code "Sooner Than Later." Retrieved from https://www.freethink.com/robots-ai/github-copilot

[117] Albrecht Schmidt, Passant Elagroudy, Fiona Draxler, Frauke Kreuter, and Robin Welsch. 2024. Simulating the human in HCD with ChatGPT: Redesigning interaction design with AI. *Interactions* 31, 1 (2024), 24–31.

[118] Orit Shaer, Angelora Cooper, Osnat Mokryn, Andrew L. Kun, and Hagit Ben Shoshan. 2024. AI-augmented brainwriting: Investigating the use of LLMs in group ideation. In *CHI Conference on Human Factors in Computing Systems.* ACM, 1–17.

[119] Simmone Shah. 2023. *The Writers Strike Is Taking a Stand on AI*. Retrieved from https://time.com/6277158/writers-strike-ai-wga-screenwriting/

[120] Jieke Shi, Zhou Yang, and David Lo. 2024. Efficient and green large language models for software engineering: Vision and the road ahead. arXiv:2404.04566. Retrieved from https://arxiv.org/abs/2404.04566

[121] Ben Shneiderman. 2022. *Human-Centered AI* (1st. ed.). Oxford University Press.

[122] Paola Spoletini and Alessio Ferrari. 2024. The return of formal requirements engineering in the era of large language models. In *Requirements Engineering: Foundation for Software Quality*. Daniel Mendez and Ana Moreira (Eds.), Springer Nature, Switzerland, Cham, 344–353.

[123] Daniel Ståhl and Torvald Mårtensson. 2021. Mob programming: From avant-garde experimentation to established practice. *Journal of Systems and Software* 180 (2021), 111017.

[124] Claire Stevenson, Iris Smal, Matthijs Baas, Raoul Grasman, and Han van der Maas. 2022. Putting GPT-3's creativity to the (alternative uses) test. In *International Conference on Computational Creativity (ICCC).*

[125] Margaret-Anne Storey, Daniel Russo, Nicole Novielli, Takashi Kobayashi, and Dong Wang. 2024. A disruptive research playbook for studying disruptive innovations. *ACM Transactions on Software Engineering and Methodology* 33, 8 (2024), 1–29.

[126] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured generation and exploration of design space with large language models for human-AI co-creation. In *CHI Conference on Human Factors in Computing Systems.* ACM, 1–26.

[127] Rasmus Ulfsnes, Nils Brede Moe, Viktoria Stray, and Marianne Skarpen. 2024. *Transforming Software Development with Generative AI: Empirical Insights on Collaboration and Workflow*. Springer Nature, Switzerland, Cham, 219–234. DOI: https://doi.org/10.1007/978-3-031-55642-5_10

[128] Viswanath Venkatesh, James Y. L. Thong, and Xin Xu. 2012. Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly* 36, 1 (2012), 157–178.

[129] Mathias Peter Verheijden and Mathias Funk. 2023. Collaborative diffusion: Boosting designerly co-creation with generative AI. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems.* ACM, 1–8.

[130] B. Violino. 2024. *The Early Returns on Gen AI for Software Development*. Retrieved from https://www.cio.com/article/1310540/the-early-returns-on-gen-ai-for-software-development.html

[131] Jing Wang and John M. Carroll. 2011. Beyond fixing bugs: Case studies of creative collaboration in open source software bug fixing processes. In *8th ACM Conference on Creativity and Cognition (C&C '11).* ACM, 397–398. DOI: https://doi.org/10.1145/2069618.2069714

[132] Sitong Wang, Savvas Petridis, Taeahn Kwon, Xiaojuan Ma, and Lydia B. Chilton. 2023. PopBlends: Strategies for conceptual blending with large language models. In *2023 CHI Conference on Human Factors in Computing Systems*. ACM, 1–19.

[133] Michael A. West. 2002. Sparkling fountains or stagnant ponds: An integrative model of creativity and innovation implementation in work groups. *Applied Psychology* 51, 3 (2002), 355–387. DOI: https://doi.org/10.1111/1464-0597.00951

[134] Fan Wu, Emily Black, and Varun Chandrasekaran. 2024. Generative monoculture in large language models. arXiv:2407.02209. Retrieved from https://arxiv.org/abs/2407.02209

[135] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing* 4, 2 (2024), 100211.

[136] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*. ACM, 841–852.

[137] Amy Zadow, May Young Loh, Maureen Frances Dollard, Gro Ellen Mathisen, and Bella Yantcheva. 2023. Psychosocial safety climate as a predictor of work engagement, creativity, innovation, and work performance: A case study of software engineers. *Frontiers in Psychology* 14 (2023). 1664–1078

[138] Yunpu Zhao, Rui Zhang, Wenyi Li, Di Huang, Jiaming Guo, Shaohui Peng, Yifan Hao, Yuanbo Wen, Xing Hu, Zidong Du, et al. 2024. Assessing and understanding creativity in large language models. arXiv:2401.12491. Retrieved from https://arxiv.org/abs/2401.12491

[139] Eric Zhou and Dokyun Lee. 2024. Generative artificial intelligence, human creativity, and art. *PNAS Nexus* 3, 3 (2024), pgae052.

[140] Albert Ziegler, Eirini Kalliamvakou, X. Alice Li, Andrew Rice, Devon Rifkin, Shawn Simister, Ganesh Sittampalam, and Edward Aftandilian. 2024. Measuring GitHub copilot's impact on productivity. *Communications of the ACM* 67, 3 (Feb. 2024), 54–63. DOI: https://doi.org/10.1145/3633453

Received 4 April 2024; revised 2 October 2024; accepted 12 November 2024