A Preliminary Study on the Impact of AI in the Creativity and Collaboration in Software Teams

Danilo Monteiro Ribeiro
Universidade Federal de Pernambuco (UFPE), cesar.school
Recife, Brazil
dmr@cin.ufpe.br

Kiev Gama
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
kiev@cin.ufpe.br

Victoria Jackson
University of Southampton
Southampton, UK
v.jackson@soton.ac.uk

Abstract

Generative artificial intelligence (GenAI) tools are reshaping software engineering work, increasingly acting as cognitive partners rather than purely instrumental aids. Most AI tooling remains optimized for individual use, yet software development is inherently collaborative and creative, raising open questions about how AI affects team-based work. This paper examines how software engineering professionals perceive and integrate AI tools focusing on creativity and collaboration within teams. We conducted semi-structured interviews with 13 software professionals from four companies, each representing a distinct team. Our analysis identifies three central themes — AI Use, Consequences of AI, and Collaboration and Team Dynamics — alongside Emotions as a cross-cutting dimension. Key findings include: AI broadens developers' creative repertoires but may narrow independent ideation; developers increasingly consult AI instead of colleagues, weakening peer learning and mentoring; and teams are developing emerging triadic collaboration patterns in which developers, colleagues, and AI reason together.

Keywords

Generative AI, Software Engineering, Creativity, Collaboration

1 Introduction

In recent years, advances in Generative Artificial Intelligence (GenAI) tools have profoundly transformed Software Engineering (SE) practices. Language models and code assistants such as GitHub Copilot, ChatGPT, and Cursor are no longer merely support instruments; they increasingly act as collaborative partners capable of generating and refactoring code, suggesting solutions, and explaining errors [17, 23]. This shift has delineated a new frontier of investigation into human–AI collaboration, in which systems move beyond a purely instrumental role and become agents integrated into individual and team workflows. Recent studies suggest that transforming AI from an individual tool into a team member requires rethinking roles, orchestration, and work processes [23], and that the nature of interactions among developers is already changing as a result [21]. Throughout this paper, we use *AI* and *GenAI* interchangeably to refer specifically to generative AI tools (e.g., ChatGPT, GitHub Copilot); other forms of AI were not part of this study's scope.

Software development encompasses a wide range of creative activities, including feature ideation, interface design, architectural design [12], and coding [14]. Creativity can be understood as the ability to produce ideas that are both new and valuable [7], and it is central to innovation in software teams, their processes, and products [11]. AI tools have the potential to expand developers' creative repertoire [15], but they also introduce cognitive and ethical challenges, such as overreliance on algorithmic suggestions and dilution of authorship [2]. Creativity and collaboration are often intertwined, with creative work occurring both in collaborative settings such as whiteboarding [12] and in independent work [14].

Although prior work has examined GenAI's effects on individual developer productivity [16, 17, 19] and on changes in developer interactions [21], creativity and collaboration have been studied largely in isolation. This separation is a gap: in software teams, creative activities such as ideation, architectural reasoning, and design are inherently collaborative [12], and AI tools that reshape how developers interact necessarily reshape how they create together. Studying these two dimensions jointly allows us to surface tensions and trade-offs that neither lens alone can reveal, for instance, whether AI-driven autonomy gains come at the cost of collective learning, or whether emerging practices such as AI-mediated collaboration introduce new forms of creative work.

Understanding how developers perceive and integrate AI into team-based work, especially in terms of creativity and collaboration, is an urgent research need. Such insight will clarify the limits and benefits of human–AI collaboration and guide design principles, ethical guidelines, and organizational practices that support effective, healthy teams and individuals in the age of AI. To address this gap, we conducted semi-structured interviews with software professionals to explore two research questions:

- **RQ1** How do software developers perceive and use AI tools in creative activities within software development teams?

- **RQ2** How does the use of AI tools influence collaboration and team dynamics in software development teams?

We report preliminary findings from interviews with 13 software professionals across four companies and various technical roles. AI supports creativity by helping developers explore the solution space, fill knowledge gaps, and surface previously unconsidered alternatives. Teams are testing new AI-mediated collaboration practices (e.g., using AI in group discussions and shared prompt writing) having reduced peer interaction and informal knowledge exchange.

2 Background and Related Work

Scholars consider creativity as related to novelty and ideas, such as in Boden's definition [7]: "Creativity is the ability to come up with ideas or artifacts that are new, surprising, and valuable". Software developers have a narrower focus and consider creativity as a form of problem-solving that emphasizes reuse and usefulness over originality and novelty [14]. Popular creativity techniques used by developers include collaborative activities such as brainstorming and whiteboarding [12]. Collaborative work is key for developers' creativity [14] as is the ability to work independently,

free from distractions, which facilitates the attainment of the flow state conducive to creativity [20]. Along with a supportive work environment, technical skills, motivation for the task, and creativity-relevant processes also help individuals be creative at work [1].

With the widespread usage of GenAI, several studies have examined its practical influence on developers' creativity, albeit indirectly. Two studies [16, 17] exploring developers' use of GenAI have reported varying perceptions of its influence on creativity: some developers report that it has improved their creativity, whereas others report that it has worsened it. Relatedly, practitioners report that GenAI is rarely used for tasks traditionally considered creative, such as design [19]. A recent interview study with 30 software development practitioners [25] found that GenAI can stimulate creative problem-solving and enhance productivity, while also raising concerns about overreliance, skill degradation, and reduced communication between developers. In theory, GenAI has the potential for aiding creativity by quickly addressing knowledge gaps [19], supporting learning [17], and helping developers explore potential options [6]. Conversely, GenAI may inhibit creativity as developers ask GenAI for help rather than colleagues [21], reducing social interactions and their inherent learning opportunities. Also, senior developers have voiced concerns that novice developers are overly reliant on GenAI [19], which may hinder the development of core competencies critical to software developers, including the collaboration skills [26] that support creativity. Overall, evidence suggests that GenAI's effect on creativity depends on how critically developers engage with its outputs.

The effects of GenAI on team collaboration have also begun to attract research attention. Salomon et al. [21] found that developers increasingly consult AI before asking colleagues, reducing interruptions but also weakening the peer exchanges through which tacit knowledge circulates, with colleagues being reserved for contextual reasoning and complex decisions. Seeber et al. [23] argue that for AI to act as a true team member, teams must rethink roles, orchestration, and shared situational awareness, which current tools only partly support. The combined impact of GenAI on creativity and collaboration in software teams remains underexplored. Prior studies tend to treat the two dimensions separately, leaving open how trade-offs between individual AI use and collective creative work manifest in practice. Our study differs from this prior work along three fronts. Salomon et al. [21] examine changes in developer-to-developer interaction patterns but do not address creativity. Seeber et al. [23] offer a conceptual research agenda on AI as a teammate rather than empirical evidence from practitioners. Zacharias et al. [25] report broad practitioner concerns about productivity, deskilling, and communication, but do not analyze creativity and collaboration jointly at the team level or examine AI's role in collective creative work.

3 Research Method

We conducted a qualitative interview study [22] to investigate how SE professionals perceive and integrate AI-assisted development tools into their daily practices, focusing on creativity and collaboration within teams. A qualitative design was selected to capture nuanced experiences, sensemaking processes, and contextual factors that quantitative measures alone cannot adequately address [22].

Table 1: Interview participants profile

ID	Current Role	Total experience (years)	Company
P1	Test team lead	14	C1
P2	Software Engineer	7	C1
P3	Software Engineer	8	C1
P4	Project Manager	23	C2
P5	Tech Lead	13	C1
P6	Tech Lead	14	C1
P7	Software Engineer Intern	<1	C3
P8	Software Engineer Intern	1	C3
P9	Software Engineer	2	C2
P10	QA Engineer	10	C4
P11	QA Lead	26	C4
P12	Software Engineer	5	C2
P13	Software Engineer	5	C2

We recruited 13 SE professionals (10 men and 3 women) from 4 companies (Table 1). Participants from the same company worked within the same team, providing both individual and within-team perspectives on AI use. Participants were required to: (i) have professional experience in software development and (ii) report active or recent use of AI-assisted tools (e.g., code assistants or generative AI systems) in collaborative development contexts. They were professionals from different roles (e.g., developers, designers, project managers), seniority levels, and organizational settings. For readability, we use the term *developers* broadly throughout this paper to refer to the full spectrum of software professionals in our sample, including QA, project management, and leadership roles, except where a specific role is relevant to the discussion. We employed an iterative, non-probabilistic sampling strategy [3].We initially recruited participants through convenience sampling within our professional networks. As data collection and preliminary analysis progressed, we noted the absence of junior developers who began learning programming after the release of ChatGPT and then used purposive sampling to recruit this relevant subgroup[3].

The study was approved by UFPE's ethics board. Participants were all volunteers and provided informed consent. All data were anonymized, treated confidentially, and stored securely.

3.1 Data Collection and Analysis

Data were collected through semi-structured interviews conducted online via Google Meet in February 2026. Each session ranged from 45 to 60 minutes and was audio-recorded with participants' consent. The interview protocol [1] included open-ended questions addressing: integration of AI tools into development workflows; perceived impact on creativity and ideation; effects on team coordination and shared understanding; and perceived risks, limitations, and organizational implications. The protocol was piloted with 2 participants; based on their feedback, we adjusted question wording and these pilot interviews were discarded from the final dataset. The protocol served as a flexible guide rather than a rigid script, but interviewers ensured that all planned topics were covered to avoid omissions. Interviews were transcribed verbatim, with participants having pseudonyms assigned and identifying information removed.

We analyzed the data using inductive thematic analysis [8]. Two researchers independently open-coded the interview transcripts and regularly compared interpretations, resolved disagreements,

[1] https://doi.org/10.6084/m9.figshare.31489231

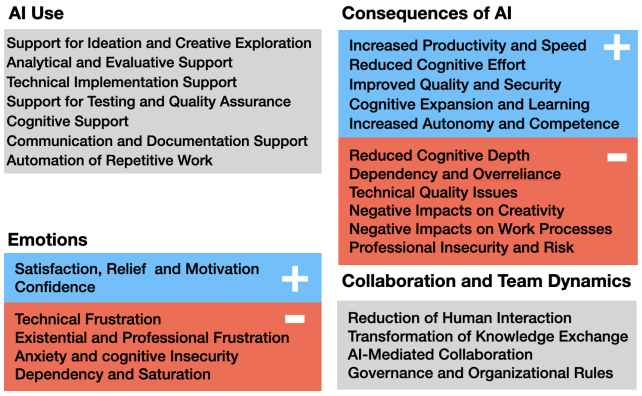

Figure 1: Themes and categories of the thematic analysis. Collaboration and Team Dynamics are presented without positive/negative distinctions, unlike AI Use and Consequences of AI, where participants reported directional effects.

and refined the coding structure. Codes were then grouped into higher-level themes through a researcher-led process supported by an LLM, used only to suggest alternative groupings and exploratory interpretations. All analytical decisions remained the responsibility of the research team. The use of LLMs to support qualitative analysis in SE research has been discussed recently, provided that human judgment governs analytical decisions [5, 24].As coding progressed and themes emerged, peer debriefing with a third researcher was used to critically examine interpretations, challenge assumptions, and refine themes, strengthening the study's credibility and trustworthiness. Coding and theme development used shared documents to ensure transparent, traceable analytic decisions.

4 Findings

The two RQs addressed, respectively, the impacts of AI on creativity and on collaborative work in software teams. Figure 1 summarizes the themes and categories identified. Our analysis revealed three central themes: (1) AI Use; (2) Consequences of AI; and (3) Collaboration and Team Dynamics. *Emotions* emerged as a cross-cutting dimension: emotional responses were not confined to a separate domain but arose in direct connection with creative and collaborative experiences, shaping how participants made sense of both.

4.1 Thematic Analysis Results

4.1.1 AI Use. This theme captures how developers integrate AI tools into their workflows. Participants described AI not only as a coding assistant but as a cognitive, creative, analytical, and communicative resource. AI was used for ideation, technical implementation, documentation and automation, indicating that AI tools are a multi-functional component embedded in daily development practices. These patterns suggest that AI is being appropriated across the full breadth of development work, including activities with creative components, rather than routine or purely technical tasks.

Support for Ideation and Creative Exploration. Participants described using AI to stimulate alternative solutions, support architectural brainstorming, and generate proofs of concept. AI was frequently used to explore design possibilities and iteratively refine ideas. While some perceived AI as expanding creative exploration, others noted that ideas sometimes "arrive already formed," potentially constraining deeper ideation. For example, P6 described using AI to generate multiple solutions and evaluate the cost-benefit of each before selecting one: "*Trying to explore possible solutions, what ends up happening is a list of options and the cost-benefit of each one...to explore options that we haven't even considered yet.*"

Analytical and Evaluative Support. Participants emphasized the importance of critically evaluating AI outputs. Responses were compared, cross-validated, and reviewed by experienced developers. AI was not perceived as a source of truth, but as a hypothesis generator requiring domain knowledge and human judgment: "*Sometimes it suggests things, we review the document and think, 'Oh no, what it said doesn't make any sense. Let's remove it.'*" (P4)

Technical Implementation Support. AI was used in everyday development tasks, particularly for generating code snippets and simplifying implementations. Participants described AI as an assistant, intern, or partner capable of accelerating implementation while developers retained responsibility for final decisions. As noted by P13: "*The coding, well, I think 70% is AI, 30% is me. I just give some pointers, see something that can be done more efficiently.*"

Support for Testing and Quality Assurance. Participants noted that AI makes it easier to create tests, including integration tests (P1), but expressed caution in security-sensitive or quality-critical contexts. AI-generated unit tests and review suggestions were considered limited without sufficient contextual knowledge, with senior developers viewed as essential for final validation.

Cognitive Support. AI functioned as an on-demand tutor, helping developers overcome knowledge gaps and clarify doubts, "*shortening the learning curve*" (P12) when learning new topics. However, effective use required knowing how to write prompts and interpret responses critically. P7 noted they asked AI to explain unfamiliar generated code: "*I either ask it to explain it better or I also use the chat to explain what's going on.*"

Communication and Documentation Support. AI was used to improve language clarity, draft documentation, summarize materials, assist in English communication, and support discussions. Notably, some participants reported that interacting with AI had become faster than consulting colleagues for these tasks, a pattern with direct implications for peer interaction within teams. Documentation tasks ranged from summarizing requirements to authoring commit messages generated entirely by AI (P13).

Automation of Repetitive Work. Participants reported using AI to optimize workflows and automate repetitive tasks (P4). At the same time, they acknowledged the risks of excessive reliance, particularly for trivial activities that may erode manual proficiency. P3 worried that such a "*strong bond*" had formed that removing AI would drastically reduce their productivity.

4.1.2 Consequences of AI. Beyond usage patterns, participants reflected on the broader cognitive, professional, and creative consequences of AI integration. AI tools were associated with productivity gains and increased autonomy, but also raised concerns about reduced analytical depth, dependency, and potential impacts on creativity and expertise development, illustrating a duality of AI as

both an enabler and a potential disruptor of professional practice. Positive (+) and negative (-) subcategories are indicated below.

(+) Increased Productivity and Speed. AI was seen to speed up tasks and boost efficiency, especially for implementation-heavy work, as P5 noted: "*[AI] autocomplete helps a lot.*"

(+) Reduced Cognitive Effort. AI reduced the need to interrupt colleagues and provided rapid answers to technical questions. This improved workflow continuity, but some participants expressed concern about reduced cognitive engagement. P7 reframed this as a positive contribution to team dynamics: "*I see it a lot like, in a way, I'm freeing up the time and energy of these people who spend more time there, things that are often basic and just that.*"

(+) Improved Quality and Security. Some participants perceived quality gains when AI suggestions were carefully validated, though improvements were conditional on human oversight and domain expertise. P1 noted that AI "*provides a bit better coverage than the developer might have considered before*", while still acknowledging ongoing concerns about security issues such as SQL injection.

(+) Cognitive Expansion and Learning. AI facilitated exposure to alternative approaches and supported learning through iterative exploration. P5 described using AI to acquire targeted knowledge before committing to a solution: "*I can ask specific questions so I can understand what it's for, in what contexts that technology or approach can or cannot solve a problem. And then I decide what is worth dedicating my effort to.*"

(+) Increased Autonomy and Competence. Participants reported greater independence, requiring less assistance from teammates. AI enabled more autonomous problem-solving: "*we can have more autonomy to solve things, and it eliminates several bottlenecks, even in terms of learning.*" (P12)

(-) Reduced Cognitive Depth. Several participants raised concerns about diminished analytical engagement and superficial understanding of solutions, particularly among less experienced developers. P9, a novice developer, felt that increasingly capable AI tools could lead developers to become "*a little lazy sometimes.*"

(-) Dependency and Overreliance. Overdependence on AI was reported as a risk, including difficulty reverting to manual practices and incomplete understanding of generated solutions. P6 described the phenomenon directly: "*Today it's almost become an addiction, you know, to do everything with artificial intelligence.*"

(-) Technical Quality Issues. AI-generated errors, hallucinations, and lack of contextual awareness were observed. Undetected issues could delay delivery or compromise architectural decisions. P5 cautioned: "*It might provide information that isn't 100% accurate.*"

(-) Negative Impacts on Creativity. While AI supported idea generation, some participants believed that excessive reliance weakens originality and deeper creative reasoning. P6 observed: "*It doesn't let things flow like they did before using artificial intelligence.*"

(-) Negative Impacts on Work Processes. AI changed workflow dynamics, shifting how teams coordinate tasks and manage responsibilities. P1 noted reduced involvement in discussions: "*I don't really see that interaction between people. It's just one or two people who are there.*"

(-) Professional Insecurity and Risk. Participants had concerns about generational skill differences, erosion of expertise, and potential long-term career impacts. P7 warned: "*If you don't think, don't ask questions, don't read what AI is doing, I think AI can harm a career.*"

4.1.3 Emotional Responses to AI. AI tool usage generated varied emotional responses. Participants described feelings ranging from satisfaction (e.g., P1, P5) and relief (e.g., P3) to frustration (e.g., P1, P5), anxiety (e.g., P13), and professional insecurity (e.g., P3, P11). These affective reactions were closely linked with perceptions of control, competence, trust, and dependency, highlighting the emotional dimension of AI-mediated work.

(+) Satisfaction, Relief, Motivation, and Confidence. AI's speed in completing tasks generated relief and satisfaction, even in tasks where developers lacked expertise (P10). For some, AI served as a supportive partner that increased confidence to assume more responsibilities: "*We end up trusting ourselves to take on more responsibilities because, before, we had all this uncertainty.*" (P6). P2 said that AI-enabled overdelivery against managerial expectations boosted motivation: "*Motivation comes from a consequence.*"

(-) Technical Frustration. Similar to other studies [19], hallucinations, verbosity, and unreliable outputs caused frustration, especially when correction effort was required. P2 described the experience: "*It is the same as talking to a wall when the AI is hallucinating.*"

(-) Existential and Professional Frustration. Some participants expressed concerns about the erosion of professional identity. P6 felt "*not needed*", with the leading role falling to AI. P12 felt their skills were no longer required, as AI goes straight to the solution, leaving them feeling less "*active in achieving the desired result*". These concerns extended to fears of being replaced (P13).

(-) Anxiety and Cognitive Insecurity. Insecurity emerged when AI-generated solutions were not fully understood. P5 worried that AI would "*probably lead to solutions that you won't even understand*", while P13 feared forgetting how to work without AI support.

(-) Dependency and Overuse. Some participants described fatigue associated with intensive AI interaction. P6 reported feeling isolated from colleagues due to AI-only workflows. P11 expressed concern about no longer trusting their own unassisted work: "*no longer trust what I do on my own.*"

4.1.4 Collaboration and Team Dynamics. AI integration extended beyond individual cognition, reshaping team-level dynamics. Participants reported shifts in communication patterns, knowledge exchange, and collaboration structures, including increased individualization of work and reduced peer interaction. At the same time, teams experimented with AI-mediated collaboration and established governance mechanisms to regulate its use.

Reduction of Human Interaction. AI consultation sometimes replaced peer collaboration, leading to more individualistic workflows. P8 described the shift: "*Instead of asking my colleagues for advice, I end up asking the AI and researching, creating a flow there that's mostly on my own.*"

Transformation of Knowledge Exchange. AI altered how experience and tacit knowledge circulate within teams. Prompt engineering has become a shared practice, but some participants worried that mentoring and contextual knowledge transfer have weakened. P1 reflected: "*The exchange of experiences I had in the past with code [...] I see that this has diminished; this learning, this searching, this exchange between people makes us learn much more. And I don't see that much anymore today.*"

AI-Mediated Collaboration. Teams experimented with collaborative prompt writing, AI-assisted discussions, and AI-integrated pair programming. P2 described how their team incorporates AI into group conversations: "*Usually, it's people chatting and someone says, "No, wait a minute, guys, I'm going to create a GPT chat here, let's see what it says." That's the reality of how it works.*" This represents an emerging triadic collaboration pattern involving developers, colleagues, and AI, in which the tool participates in collective reasoning rather than serving individual workflows alone.

Governance and Organizational Rules. Organizations implemented varying levels of formal governance, including usage policies, experimentation strategies, and tool standardization. P9 described a structured approach: "*We have some pre-established commands with pre-made, reviewed templates [...] before I send it for review by other humans, I already have it reviewed by AI.*"

4.2 Answers to Research Questions

RQ1: How do software developers perceive and use AI tools in creative activities within software development teams? We interpret creative activities broadly, encompassing not only ideation and design but also the exploratory and problem-solving dimensions of everyday development work (e.g., testing, documentation, automation), since participants described these tasks as involving choices among alternative approaches rather than purely mechanical execution. Participants reported using AI across multiple stages of the development process, including activities with creative components. AI was frequently used to support ideation and exploration of alternative solutions, particularly during architectural discussions and prototyping (see Section 4.1.1). It was also used for implementation, analytical tasks such as comparing approaches, cognitive tasks such as learning unfamiliar technologies, and communication tasks such as drafting documentation. Developers described AI as a flexible tool supporting both exploration and execution. However, they consistently emphasized the need for human validation and critical evaluation, while warning that heavy reliance may reduce independent ideation over time (Section 4.1.2).

RQ2: How does the use of AI tools influence collaboration and team dynamics in software development teams? Participants reported that AI reshapes how developers interact and collaborate within teams. A recurring observation was that developers increasingly consult AI instead of asking colleagues, reducing interruptions but also decreasing opportunities for peer discussion and informal learning (see *Reduction of Human Interaction* and *Transformation of Knowledge Exchange*, Section 4.1.4). At the same time, some teams developed emerging triadic collaboration patterns involving developers, colleagues, and AI, such as shared prompt writing and integrating AI into group discussions (see *AI-Mediated Collaboration*, Section 4.1.4). AI also enabled greater individual autonomy, allowing developers to solve problems without relying on teammates (see *Increased Autonomy and Competence*, Section 4.1.2). Despite these shifts, participants consistently positioned humans as central to development teams: architectural decisions, code validation, and leadership responsibilities were described as human tasks, with AI perceived as a supporting resource rather than a replacement for human expertise (see Section 4.1.1).

5 Discussion

Our findings indicate that AI is operating simultaneously as an amplifier of individual capability and as a fragmenting force on collective work practices. Developers report gains in autonomy, speed, and creative exploration, but these benefits come with observable costs to peer interaction, mentoring, and shared knowledge. We discuss the implications of these tensions below.

Emerging Triadic Collaboration Patterns. A finding with limited precedent in prior literature is the emergence of triadic collaboration patterns involving developers, colleagues, and AI acting jointly in team settings. Participants described practices such as collectively prompting AI during group discussions and using AI outputs as shared reference points for decision-making. These practices differ qualitatively from individual AI use and from traditional human-human collaboration, suggesting a new mode of collective work that existing frameworks for team collaboration [23] do not yet fully account for. Understanding how these patterns develop, stabilize, and affect team cohesion and knowledge sharing is an important direction for future research.

Asymmetric Human–AI Collaboration. Participants described AI as generating ideas, code, and explanations, while humans remained responsible for evaluating outputs. Developers framed AI as proposing alternatives rather than making decisions, resulting in a collaboration where AI provides generative support, and humans retain oversight and accountability. This aligns with human-centered AI principles [2] and with findings by Salomon et al. [21], about developers using AI for routine technical questions while relying on colleagues for contextual reasoning and complex decisions. Our data extend this picture by showing that asymmetry also manifests in creative activities: AI generates options, but the evaluative and selective dimensions of creative work remain human.

AI Primarily Supporting the Solution Space. Participants reported using AI mainly for implementation tasks such as generating code, exploring algorithms, summarizing documentation, and debugging. In terms of the Double Diamond model [9], these uses fall primarily within the *solution space*. AI was rarely mentioned in activities related to the *problem space*, such as identifying user needs or framing requirements, possibly reflecting the participants' roles as developers rather than product managers or designers [11]. This asymmetry suggests that AI may reinforce existing divisions of creative labor in software teams rather than redistribute them.

Changes in Perceptions of Creativity. Our results indicate that AI may be shifting how developers understand creativity in software development. Participants described creativity not only as generating new ideas but also as selecting, refining, and improving alternatives. Because AI can generate multiple candidate solutions quickly, developers may increasingly engage in evaluative forms of creativity, where the creative task involves selecting and adapting AI-generated options rather than producing ideas from scratch. This shift resonates with Amabile's componential model [1], which distinguishes generative from evaluative creative processes: AI appears to be compressing the generative phase while expanding the evaluative one. Yet, some participants suggested that heavy reliance on AI could reduce opportunities for independent ideation [11], a tension that warrants further empirical investigation.

Potential Deskilling Effects. Some participants expressed concern that frequent reliance on AI-generated solutions may reduce the need for deep reasoning or encourage superficial understanding of generated code, particularly among less experienced developers. Similar risks have been discussed in automation research [18], where extensive reliance on automated systems reduces opportunities to develop expertise. Zacharias et al. [25] report convergent findings from a broader practitioner study, documenting concerns about skill degradation and an "illusion of competence" among novice developers who mistake AI-generated outputs for genuine understanding. However, other participants in our study described AI as a learning aid that helps them understand unfamiliar technologies or explore alternative approaches. This contrast suggests that deskilling is not inevitable but may depend on whether developers engage critically with AI outputs or defer to them uncritically.

Erosion of Social Learning Dynamics. Participants reported that AI increasingly replaces situations in which developers previously consulted colleagues. Salomon et al. [21] reported a similar pattern, where developers consult AI before asking colleagues, reducing interruptions but potentially altering knowledge exchange within teams. From the perspective of Bandura's social learning theory [4], learning often occurs through observation and interaction with others. If AI substitutes for some of these interactions, opportunities for informal knowledge exchange may decrease. Our data add a further dimension: participants reported not only reduced peer consultation but also weakened mentoring relationships, suggesting that the effects on collective learning may be more structural than previously documented. Chen and Lee [10] demonstrate that team self-efficacy and team identification are prerequisites for failure-based learning behavior, which fully mediates team performance in adaptive development contexts. This is the kind of interaction AI is displacing, risking the "illusion of competence" [25] at scale.

Effects on Collective Creativity. Beyond individual learning, this reduction in peer interaction may also suppress collective creativity. Hargadon and Bechky [13] argue that creative solutions often emerge not from individual effort but from momentary collective processes, in which social interactions activate knowledge associations that no individual could reach alone. Practices such as asking a colleague for help, explaining a problem out loud, or observing how others approach a task are precisely the interactions their framework identifies as generative. This perspective aligns with Amabile's argument that failures are learning experiences that can lead to creative and successful outcomes [1]. In collaborative work, when teams face failed ideas and misunderstandings, this creates opportunities for shared reflections and creative recombination. If AI is now absorbing these interactions, it may be displacing not only knowledge exchange but also the social contexts from which collective creativity arises. This raises an important open question for future research: whether AI is genuinely substituting for collective creativity, or merely shifting the form it takes, as suggested by the triadic collaboration patterns observed in this study.

Preservation of Human Agency. Despite growing use of AI, participants consistently positioned humans as responsible for decision-making, code validation, and architectural reasoning. This active preservation of human agency can be understood as a response to the deskilling concerns and collective learning losses described above: by maintaining ownership of high-stakes decisions, developers appear to be negotiating a boundary that protects both individual expertise and team accountability. This informal norm (positioning AI as a supporting tool rather than an autonomous decision-maker) aligns with recent findings [21] and with human-centered AI design principles [2], and may be worth formalizing in organizational governance frameworks.

6 Limitations and Threats to Validity

This study is subject to social desirability bias and selective recall when participants report their GenAI usage. Recurring themes across participants reduce, though do not eliminate, these risks. Researcher subjectivity in analysis and interpretation is also a concern; to mitigate this, the analytical process was documented and the research team comprises SE academics with industry experience. This was a single country study, and findings may not transfer to contexts with different cultural or organizational norms around AI adoption. The sample included only technical staff, excluding roles such as product managers who may engage with AI differently in creative work. Also, although participants from the same company worked within the same team, our data collection was based on individual interviews instead of team-level observation; some findings framed at the team level (e.g., collaboration and governance patterns) are therefore inferred from aggregated individual perceptions rather than directly observed shared team practices. As a preliminary study, no claims to thematic saturation are made.

7 Conclusion and Future Work

This preliminary study investigated how AI tools influence creativity and collaboration in software development teams. Developers integrate AI across multiple stages of the development process, using it as a resource for ideation, exploration of alternatives, implementation, and communication. At the same time, participants reported tensions: reduced peer interaction, risk of overreliance, concerns about the erosion of critical thinking, and weakened mentoring relationships. Taken together, the findings suggest that AI reshapes both creative practices and collaborative dynamics rather than simply improving individual productivity.

Four directions emerge for future work. First, longitudinal studies should examine whether AI-related deskilling accumulates or diminishes over time. Second, the triadic collaboration observed among developers, colleagues, and AI warrants investigation into how teams establish shared norms, roles, and situational awareness. Third, future research should explore why AI use remains concentrated in solution development rather than problem framing and requirements engineering, and how organizational factors shape this distribution. Fourth, the decline in peer interaction raises the question of whether AI suppresses collective creativity [13] or enables new forms of AI-mediated collaboration. Longitudinal and observational studies could distinguish between these possibilities.

Artifact Availability

The interview protocol is publicly available at https://doi.org/10.6084/m9.figshare.31489231. Interview transcripts are not shared to protect participant confidentiality, aligned with the study's ethics approval.

References

[1] Teresa M. Amabile and Michael G. Pratt. 2016. The dynamic componential model of creativity and innovation in organizations: Making progress, making meaning. *Research in Organizational Behavior* 36 (2016), 157–183.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[3] Sebastian Baltes and Paul Ralph. 2022. Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering* 27, 4 (2022), 94.

[4] Albert Bandura. 1977. *Social learning theory*. Prentice-Hall, Englewood Cliffs, NJ.

[5] Muneera Bano, Rashina Hoda, Didar Zowghi, and Christoph Treude. 2024. Large language models for qualitative research in software engineering: exploring opportunities and challenges. *Automated Software Engineering* 31, 1 (2024), 8.

[6] Shraddha Barke, Michael B James, and Nadia Polikarpova. 2023. Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages* 7, OOPSLA1 (2023), 85–111.

[7] Margaret A Boden. 2004. *The creative mind: Myths and mechanisms*. Routledge.

[8] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[9] British Design Council. 2019. *Double Diamond Design Process*. https://www.designcouncil.org.uk/our-resources/the-double-diamond/

[10] Chung-Yang Chen and Jung-Chieh Lee. 2026. The Effects of Learning from Failure and Team Cognitive Factors on Software Process Tailoring. *ACM Transactions on Software Engineering and Methodology* 35, 4 (2026), 1–40.

[11] Kiev Gama, Cleidson R. B. de Souza, Alexander Nolte, and George Valença. 2025. A C-Level Perspective on the Role of Developers in the Creative Process of Software Startups: Fostering Developer Creativity and Participation in Innovation. *IEEE Software* 42, 3 (2025), 98–106. doi:10.1109/MS.2025.3541435

[12] Wouter Groeneveld, Laurens Luyten, Joost Vennekens, and Kris Aerts. 2021. Exploring the Role of Creativity in Software Engineering. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. 1–9. doi:10.1109/ICSE-SEIS52602.2021.00009

[13] Andrew B Hargadon and Beth A Bechky. 2006. When collections of creatives become creative collectives: A field study of problem solving at work. *Organization science* 17, 4 (2006), 484–500.

[14] Sarah Inman, Sarah D'Angelo, and Bogdan Vasilescu. 2024. Developer Productivity for Humans, Part 8: Creativity in Software Engineering. *IEEE Software* 41, 2 (March 2024), 11–16. doi:10.1109/MS.2023.3340831

[15] 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. doi:10.1145/3708523

[16] Jetbrains. 2025. State of Developer Ecosystem: 2025. https://devecosystem-2025.jetbrains.com/.

[17] Fairuz Nawer Meem, Justin Smith, and Brittany Johnson-Matthews. 2025. Why Do Software Practitioners Use ChatGPT for Software Development Tasks?. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering* (Clarion Hotel Trondheim, Trondheim, Norway) *(FSE Companion '25)*. Association for Computing Machinery, New York, NY, USA, 1508–1514.

[18] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human factors* 39, 2 (1997), 230–253.

[19] Guilherme Vaz Pereira, Victoria Jackson, Rafael Prikladnicki, André van der Hoek, Luciane Fortes, Carolina Araújo, André Coelho, Ligia Chelli, and Diego Ramos. 2025. Exploring GenAI in software development: Insights from a case study in a large Brazilian company. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 330–341.

[20] Saima Ritonummi, Valtteri Siitonen, Markus Salo, Henri Pirkkalainen, and Anu Sivunen. 2023. Flow Experience in Software Engineering *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 618–630.

[21] Marie Salomon, Ekaterina Koshchenko, Agnia Sergeyuk, Reid Holmes, Gail C. Murphy, and Thomas Fritz. 2026. From Disruptions to Discussions: How GenAI Impacts Human Interactions in Software Development. *IEEE Transactions on Software Engineering* (2026), 1–16. doi:10.1109/TSE.2026.3655626

[22] Carolyn B Seaman. 2008. Qualitative methods. In *Guide to advanced empirical software engineering*. Springer, 35–62.

[23] Isabella Seeber, Eva Bittner, Robert O Briggs, Triparna De Vreede, Gert-Jan De Vreede, Aaron Elkins, Ronald Maier, Alexander B Merz, Sarah Oeste-Reiß, Nils Randrup, et al. 2020. Machines as teammates: A research agenda on AI in team collaboration. *Information & management* 57, 2 (2020), 103174.

[24] Bianca Trinkenreich, Fabio Calefato, Geir Hanssen, Kelly Blincoe, Marcos Kalinowski, Mauro Pezzè, Paolo Tell, and Margaret-Anne Storey. 2025. Get on the Train or be Left on the Station: Using LLMs for Software Engineering Research. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 1503–1507.

[25] Jan Zacharias, Alina Popova, Moritz von Zahn, Johannes Chen, and Oliver Hinz. 2026. Developers' Dilemma: Opportunities and Pitfalls of Generative AI for Software Development. *Business & Information Systems Engineering* (2026), 1–27. doi:10.1007/s12599-026-00998-y

[26] Cheng Zhou, Sandeep Kaur Kuttal, and Iftekhar Ahmed. 2018. What Makes a Good Developer? An Empirical Study of Developers' Technical and Social Competencies. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 319–321. doi:10.1109/VLHCC.2018.8506577