

Seamful AI for Creative Software Engineering

Use in Software Development Workflows

Sarah Inman¹, Ambar Murillo², Sarah D'Angelo³, Adam Brown⁴, and Collin Green⁵, Google

// We explore the differences in goals for designing AI tools for productivity compared to creativity and propose strategies to elevate creativity in the software engineering workflow. Specifically, we consider the role of seamfulness in software development workflows as a way to support creativity. //



THE ROLE OF creativity is frequently overlooked in the design of software development tools, with emphasis

placed on productivity. This disparity is made more apparent as AI is integrated into software development tools, and much of the press and related research has been focused on demonstrating productivity

improvements. In contrast, disciplines, such as writing and visual design, have leveraged AI to support creativity. As the future of AI-driven software development takes shape, it is important to understand the long-standing focus on productivity and its relationship to creativity.

To date, IT and software development have predominantly focused on creating tools that increase developer productivity by way of automating tasks, simplifying processes, and detecting errors in code. The lack of AI-powered software development tools focused on creativity might be a result of the greater emphasis on productivity in software engineering, as well as a complicated notion of creativity as it pertains to engineering.

Creativity is not always seen as a goal in software engineering; in part, this can be explained by the unique way in which software engineers relate to creativity as centered around reusability rather than novelty.¹ However, creativity is a critical aspect of software engineering, and importantly, there is a clear possibility for AI to impact creativity in both positive or negative ways. For example, as engineers begin to rely more on AI-generated code, which is trained on preexisting code, we may prevent the evolution of code in creative and innovative ways. On the other hand, AI can be used to automate simple tasks and act as a collaborator for engineers, enabling them to engage in more creative thinking. Despite such possibilities, the current focus on productivity over creativity presents a potential problem: Will the design of AI systems that overindex for productivity overshadow or diminish opportunities for AI to support creativity?

To address this concern, we present this position article based on our review of relevant literature in

Digital Object Identifier 10.1109/MS.2025.3534085
Date of publication 29 January 2025;
date of current version 11 April 2025.

the fields of human–computer interaction, software engineering, and creativity, as well as our experience studying software developers in a professional context. We explore the differences in goals for designing AI tools for productivity compared to creativity and propose strategies to elevate creativity in the software engineering workflow. Specifically, we apply seamless design^{2,3} to AI-powered software development to consider the role of seamfulness in software development workflows as a way to support creativity. This article reflects our per-

context.” This definition provides a useful starting point for understanding creativity, but recent research has shown that engineers differentiate their forms of creativity from “mainstream” forms of creativity, namely by centering a definition of creativity on what they termed *clever reuse*.¹ This view differs from mainstream definitions of creativity, which typically highlight originality and novelty, and instead aligns with Boden’s concept of combinational creativity, i.e., “the combination of familiar ideas in unfamiliar ways.”⁶ This idea

is determined by the quantity and speed of what is being produced.

Understanding this tension between prioritizing productivity or creativity is critical. A productivity tool is often considered to be successful when time spent on the task (the input required) is reduced or the output produced is increased (or both), whereas a creativity tool is considered to be successful if a wide range of ideas are formed and something useful or novel is produced. As interviews with software engineers illustrated, creativity in engineering is not always defined as being unique nor is it aimed at being faster or more efficient, but rather about finding novel ways to refactor code or develop tools and artifacts that can be used by others in their team.¹ These competing goals have implications for the design, implementation, and use of AI-powered software development tools specifically.

Prior research has shown that engineers interact with AI suggestions in two ways: to accelerate workflows (where engineers know the next steps they want to take) or to aid in exploring potential solutions (in cases where the next step might be unclear).⁸ If we consider these ways of working as more productive (accelerate) or more creative (explore), they are naturally at odds with each other. This might be the case in scenarios in which AI suggestions are supporting engineers in acceleration of workflows but not exploration. However, this might not be true in scenarios that involve longer and more complex code suggestions where engineers may use the suggestions to accelerate so that they can explore new approaches. In other words, engineers can be both creative and productive, but our tools need to support both, intentionally, and not as a side effect.

The lack of AI-powered software development tools focused on creativity might be a result of the greater emphasis on productivity in software engineering, as well as a complicated notion of creativity as it pertains to engineering.

spectives on creativity in software development based on our professional experience and is not based on a singular study.

Creativity and Productivity in Software Development

Creativity is a nebulous concept that has been largely regarded as the production of novel ideas or solutions.⁴ For example, Plucker et al.⁵ have defined creativity as: “the interaction among aptitude, process and environment by which an individual or group produces a perceptible product that is both novel and useful as defined within a social

of combining the familiar in unfamiliar ways aligns with elements of engineers’ definition of creativity as connected to reuse.

Productivity, on the other hand, has been defined as “efficiency in production: how much output is obtained from a given set of inputs.”⁷ Although productivity for humans is more complex, this serves as a helpful juxtaposition with creativity. While software engineers can be both creative and productive, the core goals for each are different. Creativity is often determined by the characteristics of what is being produced (e.g., usable and novel) or by individual characteristics, while productivity

Vaithilingam et al.⁹ also conducted research on how engineers perceive and use Copilot, finding that while Copilot did not necessarily improve metrics like task-completion time (generally a productivity measure), and at times introduced difficulties (e.g., some participants had difficulties understanding or debugging the code introduced by Copilot), study participants still preferred to use it because it provided “a good starting point.” With the current focus on optimizing for time savings, we might overlook what it means to provide a good starting point, which could be a critical aspect of supporting creativity.

As AI in software development continues to expand to support more exploratory tasks (e.g., investigating or brainstorming) with cognitive assistants, there is increased potential to support creative thinking. Additionally, investing in creativity may lead to increased productivity in the long term. However, as discussed, designing for creativity in software engineering is an underexplored application. Therefore, we propose looking to other fields for design principles that could be applied to support creativity in software development, such as seamful design.

Seamful Design for Technological Systems

Seamful design² comes from ubiquitous computing as a foil to the long-standing design value of seamlessness. In contrast, seamful design is a technological design tenet that emphasizes bringing typically backgrounded operations to the foreground.¹⁰ Seamful design focuses on the thoughtful revealing of underlying complexity, which makes it specifically useful in technological design. In the theoretical explication

of seamful and seamless design, a seam is a design element that can be leveraged to achieve a goal or objective for the intended end-user.

Where a design goal of seamlessness might emphasize black-boxing or making technological details invisible, seamfulness highlights these complexities to account for downstream decisions. Further, seamfulness centers around user agency, whether that be downstream end-users, a system designer, or even the system itself.³ As Chalmers and Galani² write: “any design makes manifest designers’ implicit and explicit assumptions with regard to how to

We take two main themes from seamful design, which are breakdown and designing for reuse. Breakdown is defined as what happens when the qualities or properties “of even the most familiar tool may significantly differ from those of everyday ready-to-hand use.”² This signals a design principle that seeks to reveal what is not immediately obvious: for example, the configuration details of network connectivity. By revealing what is often made implicit, the end-user can make meaningful decisions about how to troubleshoot or how to tweak their configuration, which ultimately provides more autonomy

Seamful design focuses on the thoughtful revealing of underlying complexity, which makes it specifically useful in technological design.

reduce, formalize, or objectify context and activity.”

When considering the tradeoff between optimizing for productivity versus optimizing for creativity, seamful design offers a design principle for slowing down processes. In lieu of reaching for seamlessness in support of productivity gains, seamfulness focuses on highly visible design decisions and on revealing problems with or ways to adapt solutions. Seamfulness emphasizes user agency and deferring decision making to the end-user in lieu of making assumptions, which is a key component to enabling creativity. In other words, seamful design privileges divergent thinking, user agency, and transparency around how complex systems work.

or control over the system at hand. Ehsan et al.¹¹ use the concept of breakdown from seamful design to highlight how existing seams in the system can be leveraged to elicit engagement in the AI systems. They define seams in explainable AI as mismatches between the assumed reality that AI systems were designed for and the actual deployment context.

Second, the principle of designing for reuse is defined as designing for a specific end-user (e.g., a developer) rather than a general user. The principle behind designing for adaptation or reuse is that “inaccuracies, uncertainties, and boundaries in a system can be exploited as a resource.”³ Rather than attempting to offer an answer where uncertainty

exists, revealing the uncertainty can be more effective at providing an opportunity for the human in the loop. One clear use case for this is in high-risk scenarios like those in the medical AI field; however, it also has relevance in low-risk scenarios that involve engineering design decisions.

Seamful Design for AI

There is a precedent for introducing seams into AI systems in order to achieve specific goals, such as improved diagnostic accuracy. In the medical field, AI-assisted decision making, or AI-supported clinical decision support systems, have been introduced with the goal of optimizing decision making aided by both the strengths of medical professionals and AI systems. For example, Zhang et al.¹² present a solution where seams were added to an AI diagnostic system focused on sepsis. The authors designed an AI sepsis diagnostic system that moved away from the legacy “black box” approach, which makes technological details invisible, and introduced seams in their system to help physicians better understand the system complexities, such as adding model confidence signals, providing explanations of the model, and adding recommendations for more data (i.e., lab tests). These seams ensured physicians worked with the AI system in the intermediate stages of the diagnosis process (hypothesis generation and data gathering), rather than seamlessly presenting physicians a final diagnosis. Study participants preferred the new diagnostic system to the old system, perceiving that it improved the human–AI collaboration, ensuring the physician was more involved, and providing a greater degree of perceived control over the final diagnosis.

A further example of seamful design in the medical field highlights how added transparency (rather than making technical details invisible), aided human–AI onboarding for decision making. Cai et al.¹³ have found that clinicians want to understand the global properties of the AI models being used to aid clinical decision making, to form a better mental model of the AI and learn how to better engage in collaborative decision making. The medical field continues to explore how to add seams that can improve the joint decision outcome of users and AI, since accuracy and accountability are critical in the medical diagnostics field.

There are other examples where seams have been explored to achieve a multitude of goals in human–AI joint decision making: to reduce overreliance on AI,¹⁴ to improve model understanding, uncertainty awareness, and trust calibration.¹⁵ Rather than seeking to design seams out of systems, we propose design principles that focus on achieving more specific goals through seamfulness. These examples align with seamful design in the focus on leveraging inaccuracies, uncertainties, or the need for user agency in the design of complex technical systems.

Leveraging Seamful Design to Enable Creativity in Developer Workflows

Seamless integration of AI into developer workflows often promises to reduce inputs needed or to predict what would make a workflow most efficient. This is often characterized as increased productivity. However, a seamful approach to integrating AI into developer workflows seeks to increase the quality of outputs based on various goals (e.g., creativity,

code quality, etc.). This brings to the fore questions about what goals to orient AI solutions toward. As such, we argue for a seamful approach that facilitates opportunities for creativity in software development.

The two design principles central to seamful design—breakdown^{2,3} and designing for reuse¹—offer strategies for integrating AI into developer workflows in ways that enhance creativity. In the following section, we show how these two principles can specifically be used to increase creativity. These examples were crafted based on our conversations with dozens of professional software developers and our review of the literature. Additionally, they were validated by our colleagues in the field.

For breakdown, bringing technical complexities to the forefront can provide user agency in their work, which is a central part of creativity in the implementation stage. Furthermore, engineers defined creativity as the “clever reuse” of other pieces of code or of developing code that might be reused by others. In this way, a seamful approach to AI integration might focus on providing information about the reusability of code. For designing for reuse, we argue that conversational large language models (LLMs) can play a key role in diversifying perspectives during collaboration. Collaboration has not typically been a topic that AI has tackled; however, a seamful approach emphasizes more intelligent collaboration that emphasizes more diversity in thought rather than seeking to synthesize differences.

Design Principle 1: Breakdown

Code generation is a way that AI has been integrated into the implementation stage of the development lifecycle. Certain types of code may

pose lower risk of errors introduced by AI, for example boilerplate code; however, others may require closer scrutiny. In this light, breakdown as a design principle can be employed to give engineers a view of how complex or simple different tasks are in the code generation process, potentially through an explanation of the model or by surfacing confidence levels. This would give engineers the decision to have AI generate code for different parts of the process and give them signals as to what parts of the suggested code they would like to tweak, while allowing for more discovery in other parts of the process. For example, when showing an engineer a code completion suggestion, the user interface could present the confidence level of the model and include a description about why the suggestion is high confidence (e.g., boilerplate) or low confidence and encourage a more thorough review (see Figure 1).

AI solutions have potential to aid in refactoring, an area that is cited as an opportunity for creativity for engineers.¹ A seamful approach could present options to “refactor code”

with varying levels of automation, leaving some tasks considered mundane to the AI while still allowing the engineer to engage in creative problem solving. For example, a tool could present the engineer with the ability to select certain aspects to refactor with AI (e.g., changes within the file) and which aspects to do themselves (e.g., changes across files). The seamful strategy in this example is to allow for user agency by highlighting areas where AI may or may not be useful. As user agency is central to engineers’ creativity, this is crucial to design for as AI becomes more integrated into developer workflows.

Additionally, AI in the form of conversational agents can help with meaning making by aiding engineers in understanding concepts. A seamful approach to AI explanations could be to break down the explanation to the smallest constituent parts, and make it visible how the different concepts or code relate. By breaking down concepts in ways that reveal the multiple perspectives involved, engineers could more effectively make decisions on what is applicable for their problem

space, and what parts that may be reusable in other contexts.

Design Principle 2: Designing for Reuse and Adaptation

Prior research has found that collaboration, brainstorming, and exploring a solution space fosters creativity among engineers.¹ Seamful AI can foster creativity in these stages of the developer workflow by better engaging conversational agents for divergent thinking. For collaboration, intentional seams can be designed into AI systems by prompting users to engage different stakeholders when they have been excluded from the process and, ultimately, to gather a diversity of perspectives. In terms of brainstorming, conversational AI presents engineers a way to generate a diverse set of ideas without necessarily having to rely on colleagues. A seamful approach to brainstorming with AI could involve the AI purposefully showing a diverse range of solutions (rather than showing one result), or suggesting prompts to the engineer that might produce a more diverse set of results. While there has been



FIGURE 1. An example of leveraging the breakdown principle to display model confidence in line with an AI-generated code suggestion.

a big focus on accuracy in LLMs, accuracy may be less important for enabling creativity. In some cases, creativity can be supported by providing an AI experience to think with and come up with new combinations of ideas, which the engineer will then decide what to do with.

As engineers explore a new problem space, they also experience creativity as they learn, whether it be new concepts or a simple fix.¹ Past research has highlighted that having access to knowledge and information is key for the experience of creativity through learning. Conversational LLMs can help engineers have access to a wealth of information presented in easily digestible formats, such as summaries, aiding engineers in finding similar solutions to their problem spaces, which they may then learn from or cleverly adapt. For example, AI could provide links to external sources and alternative response options to prompt creative thinking (see Figure 2). AI in knowledge consumption could incorporate the seams described previously, providing engineers a diverse set of solutions.

However, seamful AI could also aid engineers in knowledge creation.

Annotating code in a file is a key seamful approach for engineers to identify inaccuracies or uncertainties, or simply document decisions. As artifacts can often be used downstream of the software engineers' team, documenting the learning process along the way is valuable to creating reusable and reproducible code. Using the design principle of fostering adaptability and reusability, documenting processes is a strategy for integrating with AI in ways that do not occlude creativity. AI could aid creativity in this process by capturing rejected alternatives from existing documentation. In other words, preserving the "paths not taken" would help downstream developers make sense of their own process as it relates to the documentation rather than subsuming all of these rejected designs into a clean piece of text. This achieves two goals: the preservation of engineers' research efforts and the creation of downstream artifacts that can be used in the future.

Design Implications: Balancing Simplicity and Configurability

Aligning with Bućinca et al.,¹⁴ our argument acknowledges the tradeoff when adding seams, in terms of acceptability of design and performance. We argue that this does not necessarily have to be the case, and there are several ways to mitigate potential negative impacts on user experience and productivity. There are important considerations when introducing seams for supporting creativity; if the development task is time sensitive, a seam might not be well received or foster creativity. On the other hand, if the task involves designing a system or exploring ways to solve a complex problem, this could be an opportunity to introduce seams to foster creativity. In tasks where it is important for engineers to feel ownership over the process and product, adding seams that break down technical complexities could support creative thinking and increase a sense of agency in decision making. There is also a tradeoff here in allowing the AI to fully own parts of the developer workflow that

✦ Powered by AI

Concept 1 is a

1 2

Sources

1 2

Alternative responses

Explain to me like I'm 5

Pros and Cons List

Comparison with other options

FIGURE 2. An example of leveraging the reuse and adaptation principle to show display information sources and alternative explanations.

may have been assessed by productivity metrics, while designing for more user agency in other parts, such as refactoring, writing code, debugging, and brainstorming.

Even for tasks that may be well-suited for fostering creativity through AI seams, not all users may want to experience disruption throughout the developer lifecycle. Buçinca et al.¹⁴ have proposed “adaptive strategies” as a mitigation strategy, where seams are introduced only in a small number of cases where they are thought to be most effective in aiding human–AI decision making. This idea could be extended to the developer space, where AI seams for creativity are introduced in cases where they will be most impactful.

Prior research¹⁵ found that domain expertise moderated the effectiveness of an AI seam (AI explanations). Buçinca et al.¹⁴ examined how cognitive forcing functions (a seam) could reduce overreliance in AI, noting that “human cognitive motivation moderates the effectiveness of explainable AI solutions,” also highlighting an individual characteristic that may have an impact on the effectiveness of seams. Therefore, introducing personalization of AI seams for creativity may be another way in which their effectiveness may be increased. Giving users more control over where their AI seams for creativity are introduced is another way to mitigate potential negative effects on the user experience or productivity. This could be done by allowing users to opt out of the experience, or fine-tune how they want to interact with AI seams through easily modifiable settings.

In other words, if only assessed through the lens of productivity metrics, one might easily draw the conclusion that incorporating seams degrades the user experience of developer tooling, ultimately hindering the productivity

of engineers. However, we do not have enough evidence to argue that an introduction of seams geared toward creativity gains would lead to increased time on task or decreased productivity as we currently define productivity. This opens up an area where more research around the relationship between productivity and creativity is needed.

The tension between productivity and creativity in software development tools can be seen as one between simplicity and configurability. On the one hand, AI seeks to make the engineer’s day-to-day work simpler, faster, and more efficient by generating code snippets, reviewing code, automating design, and so on. In turn making the engineer more productive. On the other hand, the engineer as an expert wants autonomy and control over their environment, seeking to configure their environment or potentially explore different options. Put another way, engineers want to be creative. Engineers often possess a deeper contextual and holistic understanding of the development lifecycle, which extends beyond technical proficiency often met by AI. This expertise is key to ensuring overall quality of software products but also central to engineers’ own creative expression in their day to day work.

We have highlighted how a seamless approach to AI can address needs for configurability and can leverage engineer expertise. As we have reviewed, engineers relate to creativity in their work in unique ways: They see themselves as creative when engaged in learning new concepts or understanding how systems work, in refactoring code they have not seen before, in engaging in work that will be reusable by their team or imagined users, in reusing code in clever ways,

and in creative problem solving with others. Integrating AI in ways that make the inner-workings of code visible to the engineer and supports the annotation of documents has potential to aid creativity in its focus on reuse. In certain stages of development, there may be cases in which AI could aid creativity, e.g., if the user is presented with many unique ideas through conversations with AI, rather than being problematic to accuracy, it could engender more creative thinking in brainstorming and collaboration stages of the developer workflow. 

Acknowledgment

We thank Alberto Elizondo for his contributions in product design mocks that helped clarify our design concepts; and André van der Hoek, David Ribes, Micheal Bachman, Betty Liao, Will Seidelman, and the Engineering Productivity Research team for their valuable feedback.

References

1. S. Inman, S. D’Angelo, and B. Vasilescu, “Developer productivity for humans, part 8: Creativity in software engineering,” *IEEE Softw.*, vol. 41, no. 2, pp. 11–16, Mar./Apr. 2024, doi: [10.1109/MS.2023.3340831](https://doi.org/10.1109/MS.2023.3340831).
2. M. Chalmers and A. Galani, “Seamful interweaving: Heterogeneity in the theory and design of interactive systems,” in *Proc. 5th Conf. Designing Interactive Syst.: Processes, Pract., Methods, Techn.*, 2004, pp. 243–252.
3. S. Inman and D. Ribes, ““Beautiful seams” strategic revelations and concealments,” in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2019, pp. 1–14.
4. T. M. Amabile et al., “A model of creativity and innovation in organizations,” *Res. Organizational Behav.*, vol. 10, no. 1, pp. 123–167, 1988.
5. J. A. Plucker, R. A. Beghetto, and G. T. Dow, “Why isn’t creativity more

ABOUT THE AUTHORS



SARAH INMAN is a user experience researcher on Google Cloud's Storage team at in Google, Seattle, WA 98109 USA. Contact her at icsarah@google.com.



AMBAR MURILLO is a user experience researcher on the Core Developer team at Google, 80636 Munich, Germany. Contact her at ambarm@google.com.



SARAH D'ANGELO is a user experience researcher on the Engineering Productivity Research team at Google, 1010 Auckland, New Zealand. Contact her at sdangelo@google.com.



ADAM BROWN is a quantitative user experience researcher on the Engineering Productivity Research team at Google, New York, NY 10011 USA. Contact him at adambrovn@google.com.



COLLIN GREEN is a user experience researcher on the Engineering Productivity Research team at Google, Sunnyvale, CA 94089 USA. Contact him at colling@google.com.

important to educational psychologists? Potentials, pitfalls, and future directions in creativity research," *Educational Psychol.*, vol. 39, no. 2, pp. 83–96, 2004, doi: [10.1207/s15326985ep3902_1](https://doi.org/10.1207/s15326985ep3902_1).

6. M. A. Boden, *The Creative Mind: Myths and Mechanisms*. London, U.K.: Routledge, 2004.

7. C. Syverson, "What determines productivity?" *J. Econ. Literature*, vol. 49, no. 2, pp. 326–365, 2011, doi: [10.1257/jel.49.2.326](https://doi.org/10.1257/jel.49.2.326).

8. S. Barke, M. B. James, and N. Polikarpova, "Grounded copilot: How programmers interact with code-generating models," in *Proc. ACM Program. Lang.*, 2023, vol. 7,

no. OOPSLA1, pp. 85–111, doi: [10.1145/3586030](https://doi.org/10.1145/3586030).

9. P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models," in *Proc. CHI Conf. Human Factors Comput. Syst. Extended Abstr.*, 2022, pp. 1–7.

10. M. Weiser, "The computer for the 21st century," *Sci. Am.*, vol. 265, no. 3, pp. 3–11, 1991, doi: [10.1038/scientificamerican0991-94](https://doi.org/10.1038/scientificamerican0991-94).

11. U. Ehsan, Q. V. Liao, S. Passi, M. O. Riedl, and H. Daumé Iii, "Seamful XAI: Operationalizing seamful design in explainable AI," in *Proc. ACM Human-Comput. Interact.*, 2024, vol. 8, no. CSCW1, pp. 1–29, doi: [10.1145/3637396](https://doi.org/10.1145/3637396).

12. S. Zhang et al., "Rethinking human-AI collaboration in complex medical decision making: A case study in sepsis diagnosis," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2024, pp. 1–18, doi: [10.1145/3613904.3642343](https://doi.org/10.1145/3613904.3642343).

13. C. J. Cai, S. Winter, D. Steiner, L. Wilcox, and M. Terry, "Hello AI": Uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making," in *Proc. ACM Human-Comput. Interact.*, 2019, vol. 3, no. CSCW, pp. 1–24, doi: [10.1145/3359206](https://doi.org/10.1145/3359206).

14. Z. Bućinca, M. B. Malaya, and K. Z. Gajos, "To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making," in *Proc. ACM Human-Comput. Interact.*, 2021, vol. 5, no. CSCW1, pp. 1–21, doi: [10.1145/3449287](https://doi.org/10.1145/3449287).

15. X. Wang, and M. Yin, "Are explanations helpful? A comparative study of the effects of explanations in AI-assisted decision-making," in *Proc. 26th Int. Conf. Intell. User Interfaces*, 2021, pp. 318–328, doi: [10.1145/3397481.3450650](https://doi.org/10.1145/3397481.3450650).