# AI and the Future of Software Development: Learning from Industry, Research, and Human-Centered Theories

CIbSE 2025, Ciudad Real, Spain

Margaret-Anne Storey

chisel

University of Victoria

# SPACE: A framework for understanding productivity


productivity 14 of 29

productivity 2 of 29

FIGURE 1: EXAM

LEVEL

INDIVIDUAL One person

TEAM OR GROUP People that work together

SYSTEM End-to-end work through a system (like a devel-opment pipeline)

productivity 1 of 29 TEXT ONLY

The SPACE of Developer Productivity

THERE'S MORE TO IT THAN YOU THINK.

NICOLE FORSGREN, Github
MARGARET-ANNE STOREY, University Of Victoria
CHANDRA MADDILA, THOMAS ZIMMERMANN, BRIAN HOUCK, AND JENNA BUTLER, Microsoft Research

Developer productivity is complex and nuanced, with important implications for software development teams. A clear understanding of defining, measuring, and predicting developer productivity could provide organizations, managers, and developers with the ability to make higher-quality software—and make it more efficiently.

Developer productivity has been studied extensively. Unfortunately, after decades of research and practical development experience, knowing how to measure productivity or even define developer productivity has remained elusive, while myths about the topic are common. Far too often teams or managers attempt to measure developer productivity with simple metrics, attempting to capture it all with "one metric that matters."

One important measure of productivity is personal perception;[1] this may resonate with those who claim to be in "a flow" on productive days.

There is also agreement that developer productivity is necessary not just to improve engineering outcomes, but also to ensure the well-being and satisfaction of developers, as productivity and satisfaction are intricately connected.[12,20]

acmqueue | january-february 2021 1

ACMQUEUE – March 2021
https://aka.ms/SpaceFramework

S Satisfaction and well-being

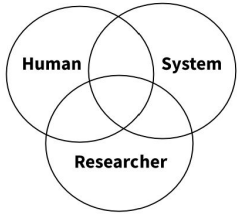
P Performance

A Activity

C Communication and collaboration

E Efficiency and flow

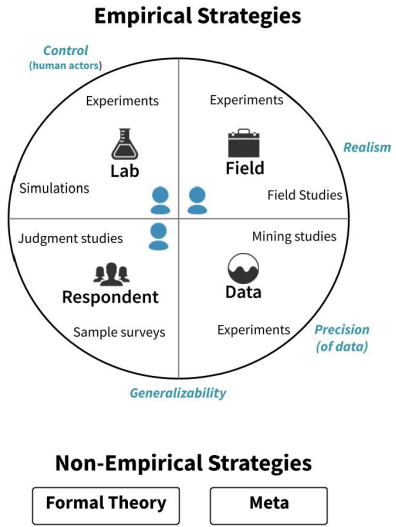
**Who?**
*(is the main beneficiary)*




**What?**
*(type of research contribution)*

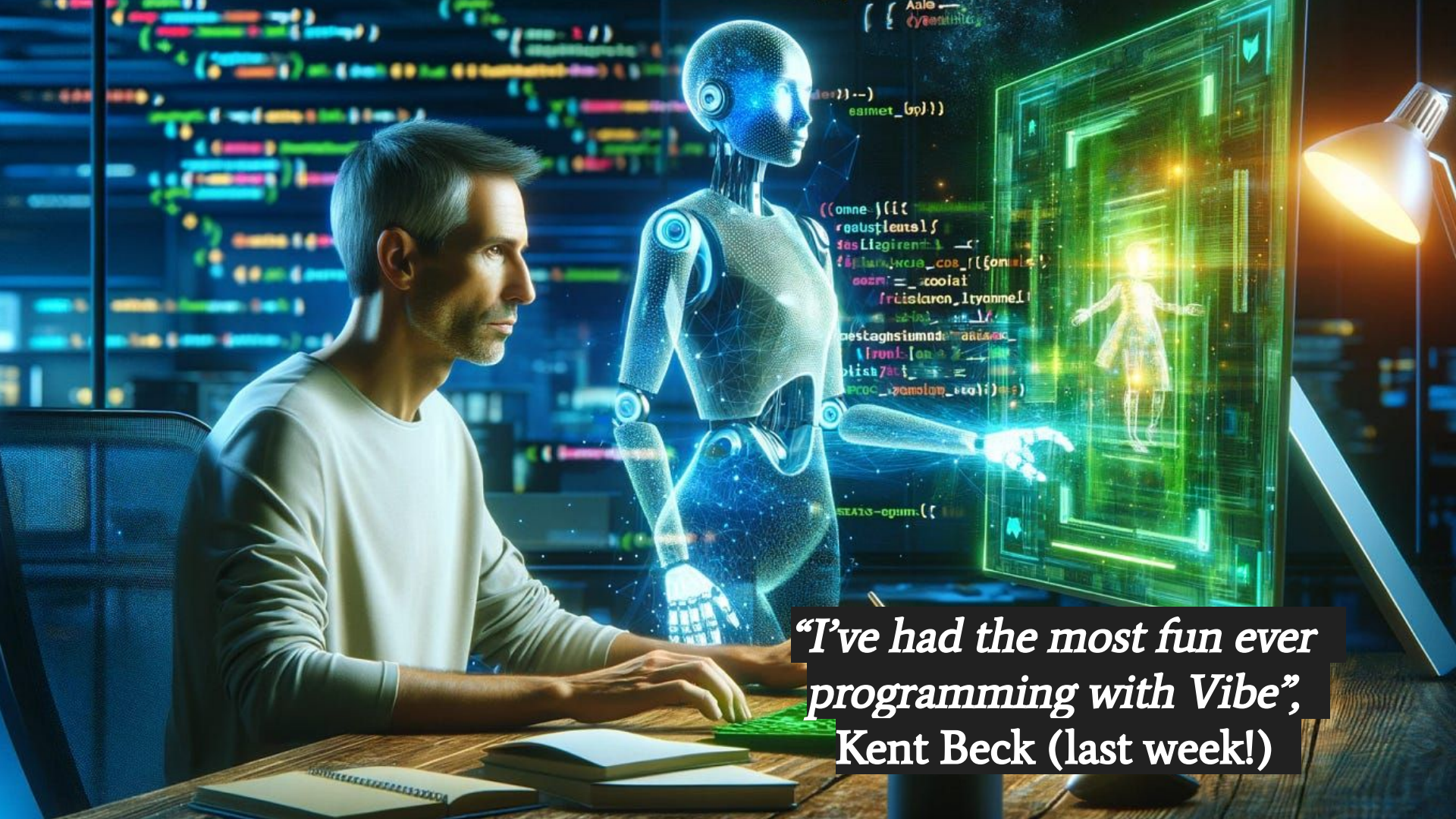
Descriptive | Solution

**How?**
*(which research strategies are used)*




https://bit.ly/wwh-framework

Storey, M., Ernst, N.A., Williams, C. et al. The who, what, how of software engineering research: a socio-technical framework. Empirical Software Eng 25, 4097–4129 (2020).


“I’ve had the most fun ever programming with Vibe”,
Kent Beck (last week!)

*"**AI measurement is a mess** – a tangle of sloppy tests, apples-to-oranges comparisons and self-serving hype that has left users, regulators and AI developers themselves grasping in the dark"* [Roose 2024, New York Times]

*"Industry has to move fast, and doesn't have the time to study many of the **human and social aspects** that we see emerging"*
Peter Rigby (Meta), March 2024 (Victoria BC)

Most studies of GenAI4SE focus on
development speed, code quality, and tool satisfaction

We need to leverage and extend
**foundational human-centric theories**
that outlive tools and aggregate our findings


https://www.kidovate.ca/


https://www.kidovate.ca/

Entrepreneurship for kids invokes Creative Problem Solving (Goals, Constraints, Opportunities)

**Flow** experiences are critical for creative problem solving

**Tools augment** our capabilities to support creative problem solving

**Collaboration** enhances our experience and our creative problem solving abilities

# Three Lenses to Understand how GenAI may Support *Creative Problem Solving in Software Development*

Flow Experiences

Cognitive Augmentation

Collaboration

# Three Lenses to Understand how GenAI may Support *Creative Problem Solving in Software Development*

**Flow Experiences**

Cognitive Augmentation

Collaboration

# Theory of Flow - Characteristics

Intense **concentration**

Merging of **action and awareness**

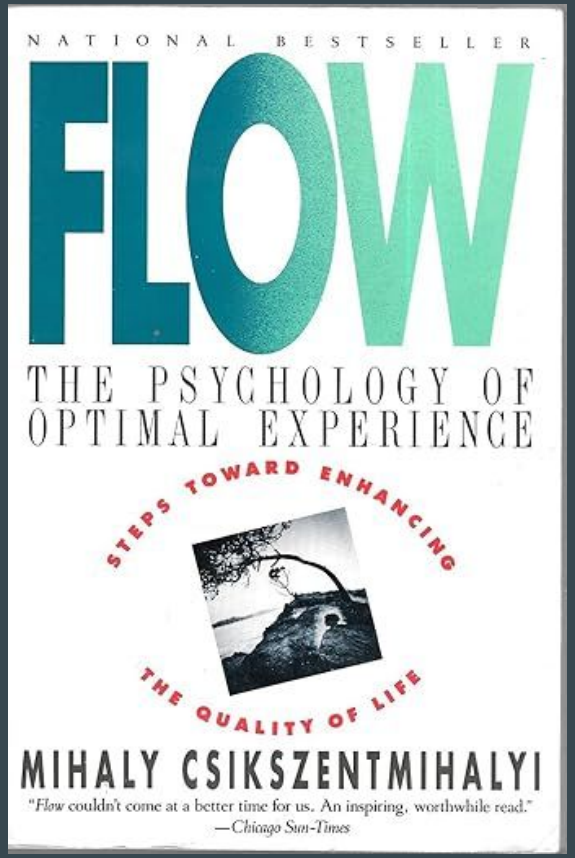
Loss of reflective **self-consciousness**

Sense of **control** over one's actions

Distorted **temporal** experience

**Autotelic** experience (activity is intrinsically rewarding)

[Csikszentmihalyi, 1991]

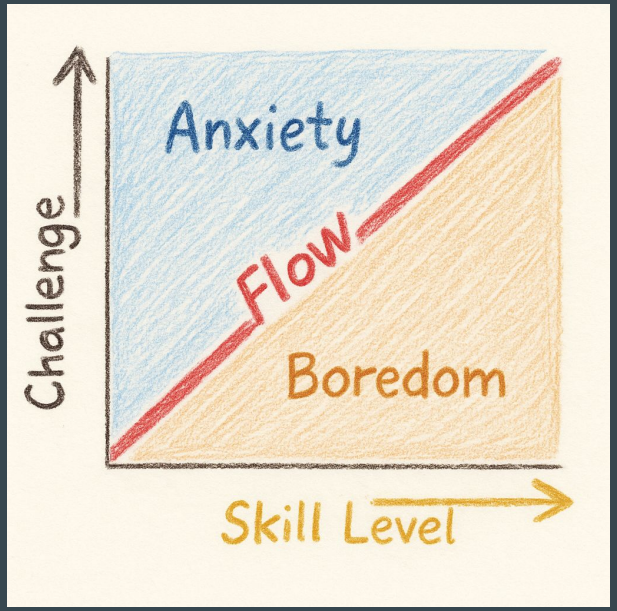


# Conditions For Flow

**Balance** between one's skill and challenges

Clear proximal **goals**

Immediate **feedback**




[Csikszentmihalyi, 1991]

# Collective flow

Flow experiences are often enhanced when we are in the company of others [Csikszentmihalyi, 1991]

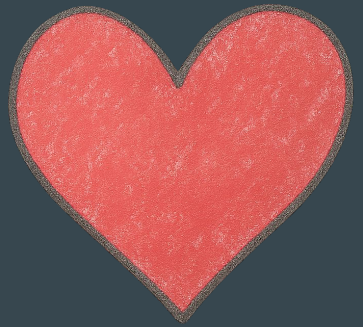
Efficacy ***beliefs*** positively predict ***collective flow*** over time (and vice versa) [Salanova, 2014]

# What happens when we achieve flow?

Enhances **creativity** , **learning** and **motivation**

We feel **joy** during times of optimal experience

We reach **peak performance**



# Studies about **Flow** in software development (pre GenAI)

***Barriers*** *may be personal, interpersonal or situational* *[Ritonummi2023]*

**Interruptions** [Meyer2017, Ritonummi2023, Ma2024]
**Distractions** (context switching) [Meyer2019]

Insufficient **focus time** (for creativity, learning) [Brown2023, Ritonummi2023]
Interventions to improve focus time (Flowlight) [Meyer2017]

Tool **friction** [Brown2023, Ritonummi2023]
Waiting on others, getting **stuck** [Ritonummi2023]

Not enough (or too much) **challenge** [Ritonummi2023]

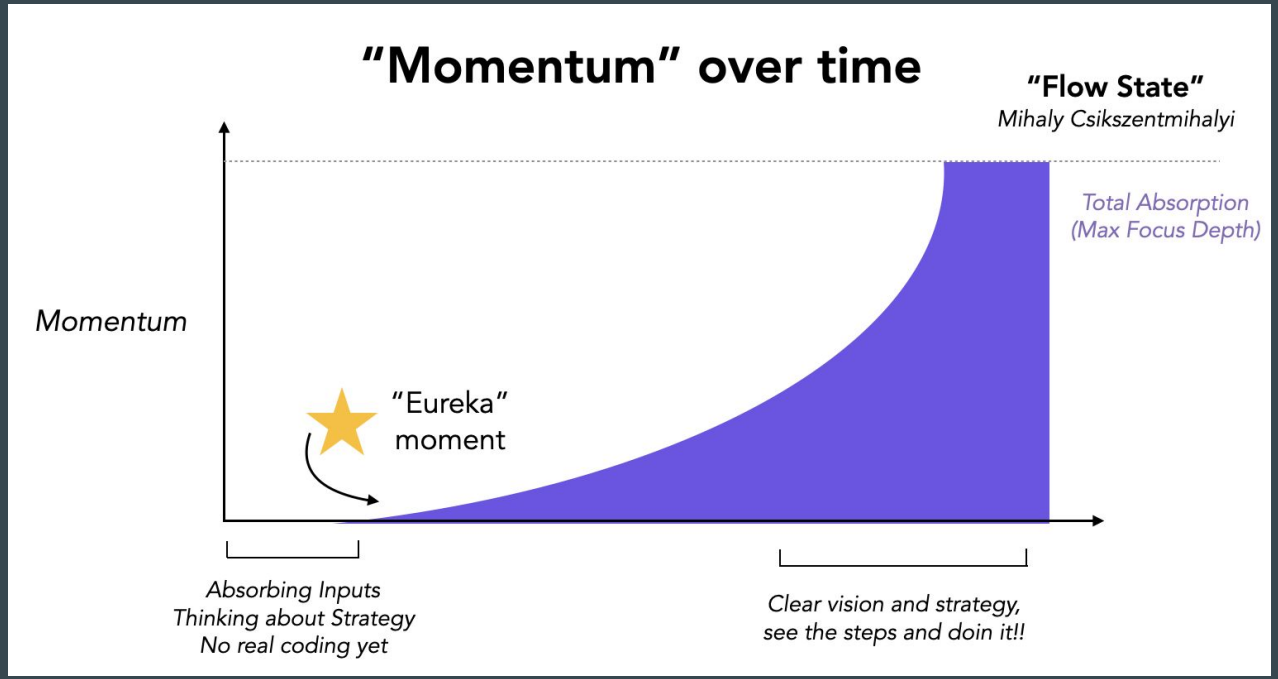
# Studies about Flow in software development (pre GenAI)

***Enablers:***

Importance of **goal setting** to facilitate flow in SE [Meyer2019]

**Fast feedback** facilitates flow in SE [Petersen2011] [Noda2023]

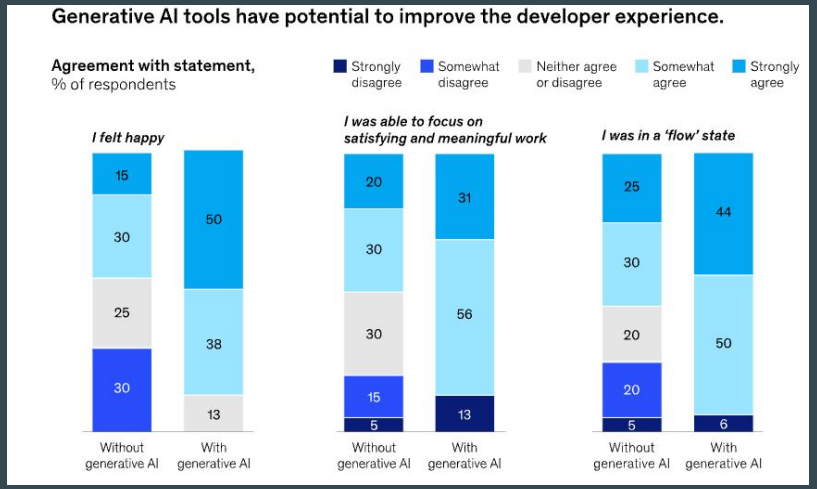
...fewer studies about enablers!



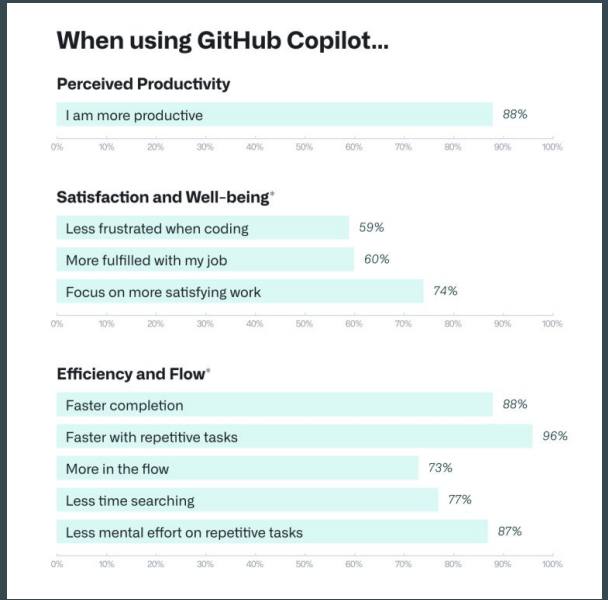

https://www.linkedin.com/in/artystarr/

# A New Theory of Developer Flow: Troubleshooting and Momentum

# Flow in the Age of GenAI: what have we learned? (1)




More likely to be in flow and to focus using GenAI [McKinsey, 2023]




GitHub: More in flow, reduced repetitive tasks [Kalliamvakou, 2024]

# Flow in the Age of GenAI: what have we learned? (2)

*Microsoft study:* 88% of users reported a change in how they worked with GitHub Copilot, reporting more **"fun work"** and less "boilerplate work" [Butler, 2025]

*Meta:* Multi-line recommendations if not done carefully can interrupt flow (**jarring effect**). Experiments led to designs with 99% adoption [Dunay, 2024]

*Academic Thesis:* AI helps some developers achieve flow, but may cause others to get **stuck** if there is a lack of **task context** [Lange, 2024]

*"Ade" is finding their way of programming in which you do* ***a lot of prompt*** *engineering,* ***less satisfying, more dull, and less likely to put them into flow.*** *And they think it is because of the structure of the activity. At the end, they are less productive because they cannot force themself as much to stay focused on the task.*
[private communication from a colleague]

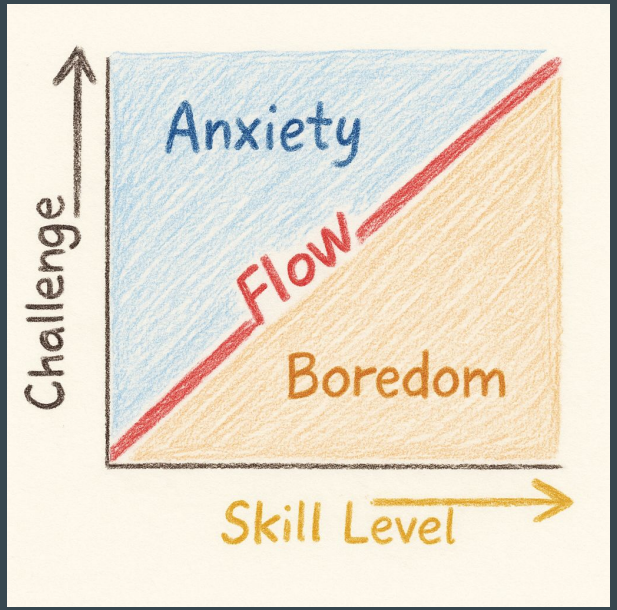
# Conditions For Flow (recap)

**Balance** between one's skill and challenges

Clear proximal **goals**

Immediate **feedback**

[Csikszentmihalyi, 1991]




***Do we need to explore how GenAI can help set the conditions for development flow?***

# Three Lenses to Question how GenAI may Support *Creative Problem Solving in Software Development*

Flow Experiences

**Cognitive Augmentation**

Collaboration

Laws of Media applied to GenAI in SE

Cognitive support theories, make hard things easier

Cognitive augmentation, to expand human thinking and creativity

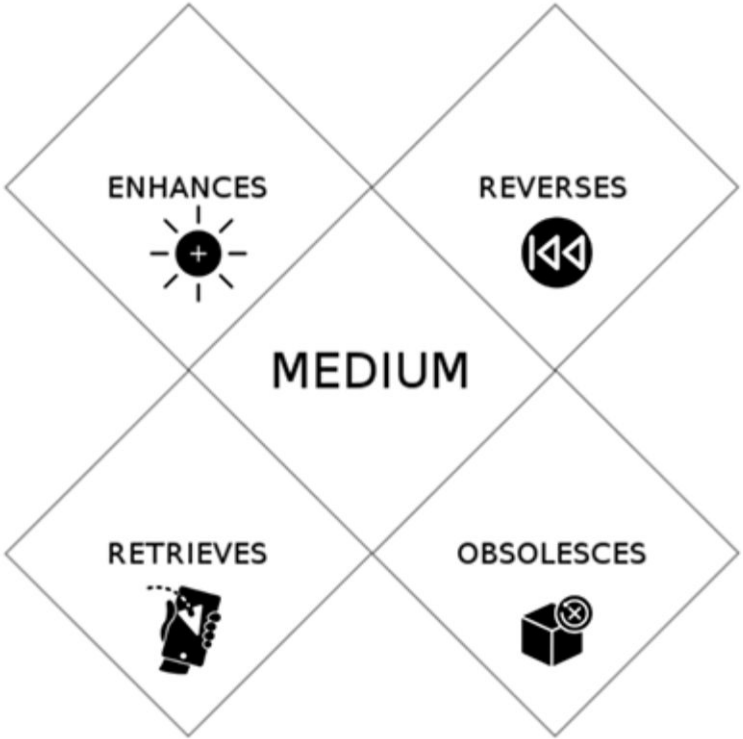
**Cognitive Augmentation**

## Laws of media applied to GenAI in SE

Cognitive support theories, make hard things easier

Cognitive augmentation, to expand human thinking and creativity

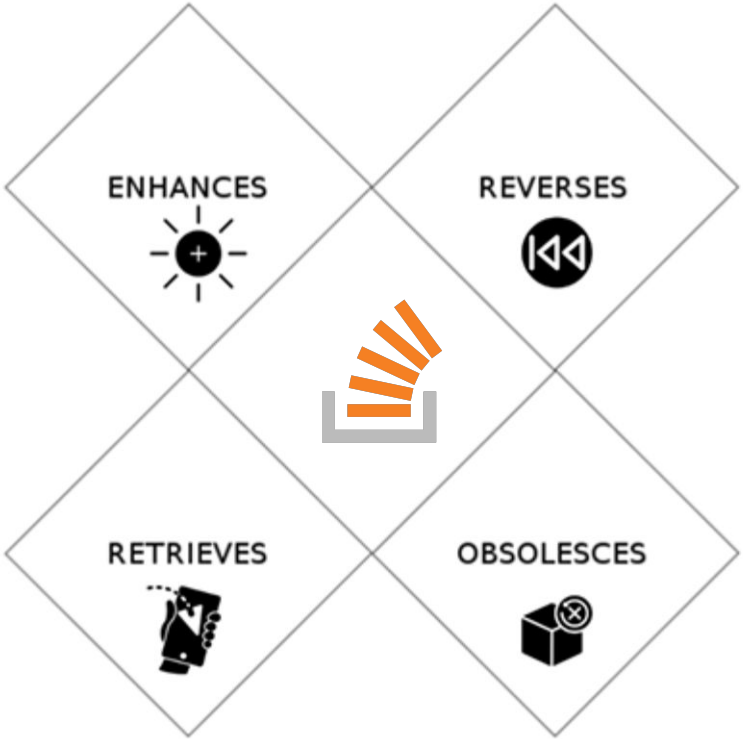
**Cognitive Augmentation**




# 30 | McLuhan’s **tetrad**: Four laws of media

McLuhan, M. (1975). McLuhan's Laws of the Media. *Technology and culture*, *16*(1), 74-78.

*Faster answers to questions, Debugging, Community*




*Trust, Blind Hacking, Flips into: LLMs (drive for more customization)*

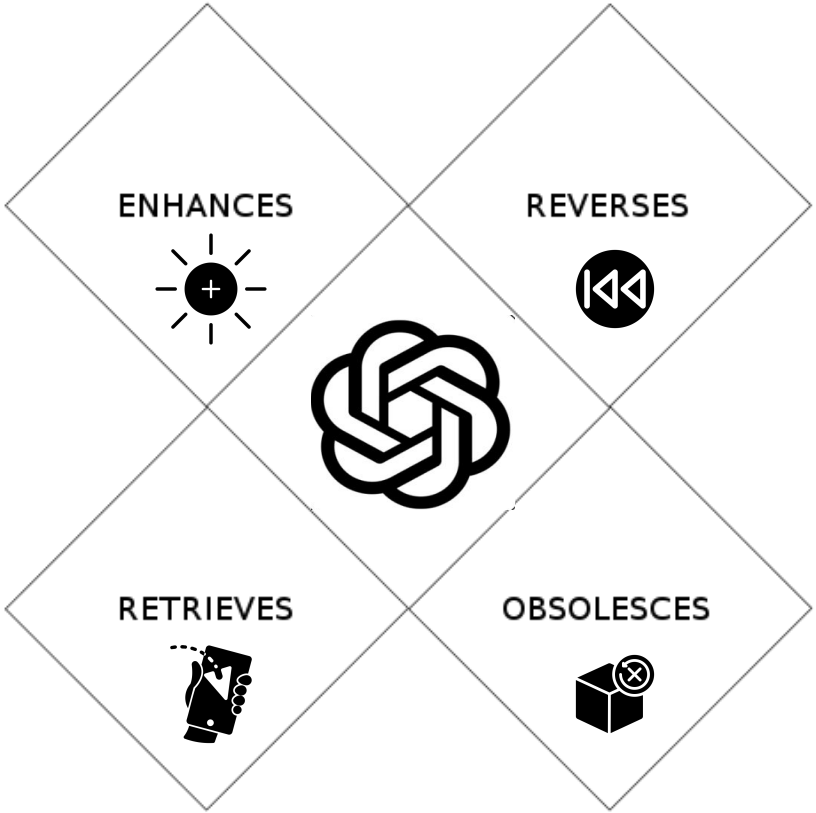
*Gurus, Portfolios*

*Email, Documentation, Textbooks, Onboarding docs*



https://bit.ly/research-playbook

*Automation possibilities*
*Productivity enhancements*
*Quality improvements*
*Personalized solutions*
*Documentation, code reviews*
*Better time estimates*
*Communication, Teamwork*
*Creativity of dev and teamwork*
*Who can be a developer*




*Over-reliance*
*Homogeneous solutions*
*Devaluation of dev craft*
*Loss of control, trust*
*Low understanding*
*Lost provenance*
*Model collapse*
*Lack of empathy for humans*

*Natural language (pseudocode)*
*Chatbots, AI agents*
*End user programmers*
*Ethical/privacy concerns*
*Improved diversity*

*Barriers to entry*
*Writing code*
*Search, Stack Overflow*
*Manual documentation,*
*Manual tests, code reviews*
*Human troubleshooting*
*Nerd culture, narrow skills*
*Traditional education*
*Onboarding support*

Software Engineering by and for Humans in an AI Era
Authors: Silvia Abrahão, John Grundy, Mauro Pezzè, Margaret-Anne Storey, Damian Andrew Tamburri, TOSEM, Feb 2025.

https://bit.ly/research-playbook



***What will GenAI reverse into when overused in (some particular) SE task ?***

Laws of Media applied to GenAI in SE

**Cognitive support theories, make hard things easier**

Cognitive augmentation, to expand human thinking and creativity

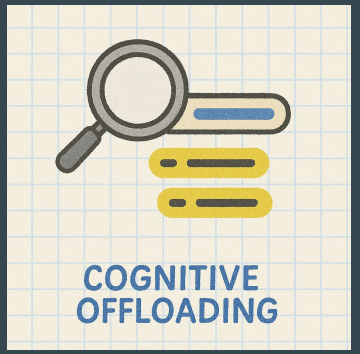
**Cognitive Augmentation**

# Cognitive Support Theories [Walenstein, 2000]

Humans perform cognitive tasks like searching, matching, recalling, reasoning, problem solving

Tools support cognition by answering “What is hard about this task, and how can this tool help?”




https://www.researchgate.net/publication/200086146_Cognitive_Support_In_Software_Engineering_Tools_A_Distributed_Cognition_Framework

## Types of cognitive support:

[Walenstein, 2000]

- **Search** support (locate relevant information)
- **Inference** support (e.g., showing dependencies)
- **Transformation** support (e.g., refactor code)
- **Traceability** support (e.g., helps users understand changes over time)

## Evaluating cognitive support:

- Task-Artifact **fit**
- Does it reduce **cognitive cost** or does it overload memory or attention

***How can cognitive support theories be applied to understanding impact of***

Laws of Media applied to GenAI in SE

Cognitive support theories, make hard things easier

- **Cognitive augmentation, to expand human thinking and creativity**

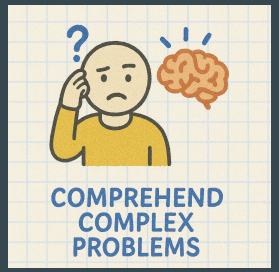
**Cognitive Augmentation**

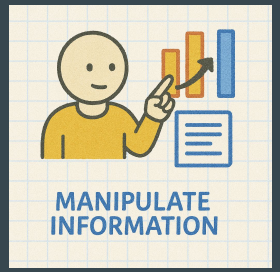
# **Augmenting Human Intellect:** A Conceptual Framework [Engelbart, 1962]

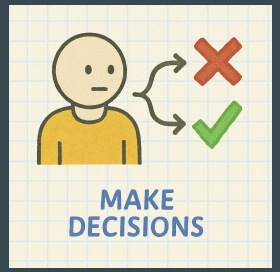
**Augmentation vs. Automation:**
Engelbart emphasized augmenting human capabilities (e.g., thinking, learning, reasoning) rather than replacing these activities

# **Augmenting Human Intellect:** A Conceptual Framework [Engelbart, 1962]

Engelbart proposed technology can **amplify** human intellect by improving our ability to:





Engelbart predicted a **co-evolution of humans and tools** where cognitive work is distributed between individual and external systems (tools, representations, environments)

# Engelbart's theory of Cognitive Augmentation cont.

**Bootstrapping Strategy:** Improving tools that help us improve tools — a recursive model for accelerating intellectual growth (GPTs!)

***How can we improve tools that help improve tools to augment developer intellect - but without forgetting the goal is to improve developer intellect?***

Engelbart also influenced ongoing ideas about **Human-AI collaboration!**

# Human-AI Collab Calls for New **Meta-Cognitive** Skills [Tankelevitch2024]

**Awareness** of what you know and don't know: reflecting on how you prompt and how to adapt if results are suboptimal, knowing "how to ask"

**Monitoring** your own understanding and strategy use: critiquing the output and asking if it aligns with your goals

**Regulating** your learning and problem-solving: asking if you are over relying on it, calibrating one's level of trust, knowing when to ask for help

***What meta-cognitive skills do we need to develop to use GenAI so it truly augments our creative problem solving ability in SE?***

# Three Lenses to Question how GenAI may Support *Creative Problem Solving in Software Development*

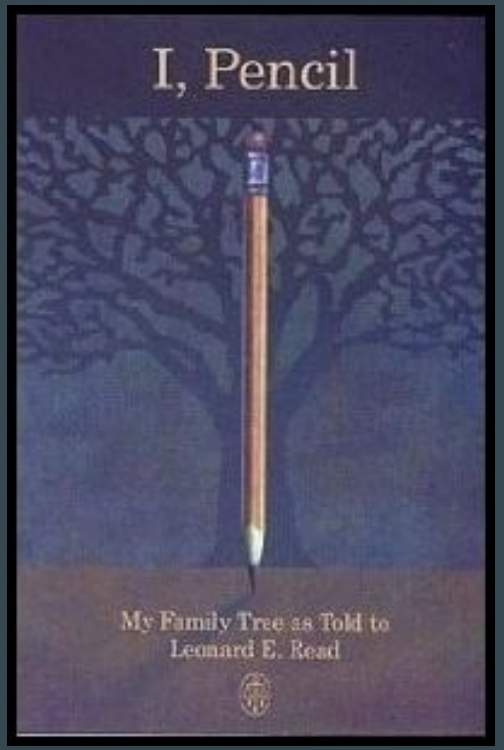
Flow Experiences

Cognitive Augmentation

**Collaboration**

# Collaboration

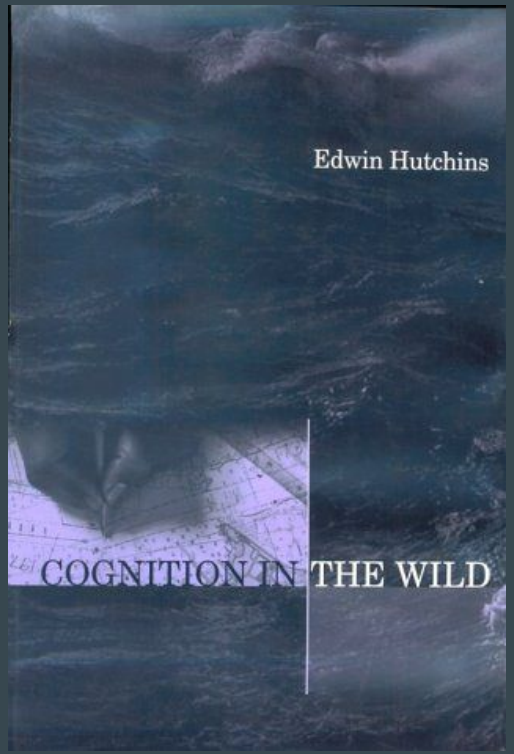
*A program is a shared mental construct, a "theory of the program", which lives in the minds of people who work on it*
[Naur, 1985]



# **Distributed Cognition** [Hutchins, 1996]

Theoretical framework that expands the boundaries of cognitive processes beyond the individual mind to include cognition that is distributed across people, tools and time

- **People** (and their social interactions)
- **Artifacts** (tools, representations, documents)
- **Environments** (physical or digital contexts)



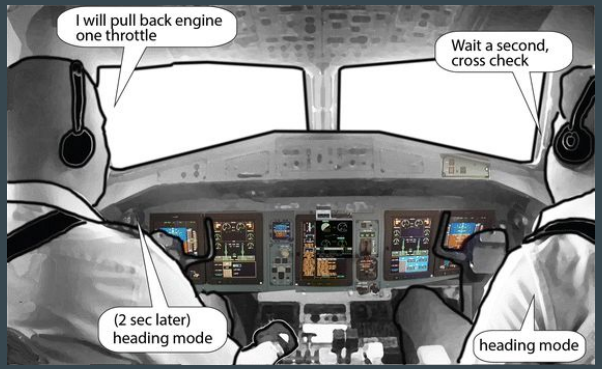

https://mitpress.mit.edu/9780262581462/cognition-in-the-wild/

# Distributed Cognition cont. [Hutchins, 1996]

Cognition is **social** – decision making happens collaboratively

Knowledge, reasoning and ideas **emerge from human interactions**

Reasoning and learning are **temporal** - spread across multiple episodes of work




https://mitpress.mit.edu/9780262581462/cognition-in-the-wild/

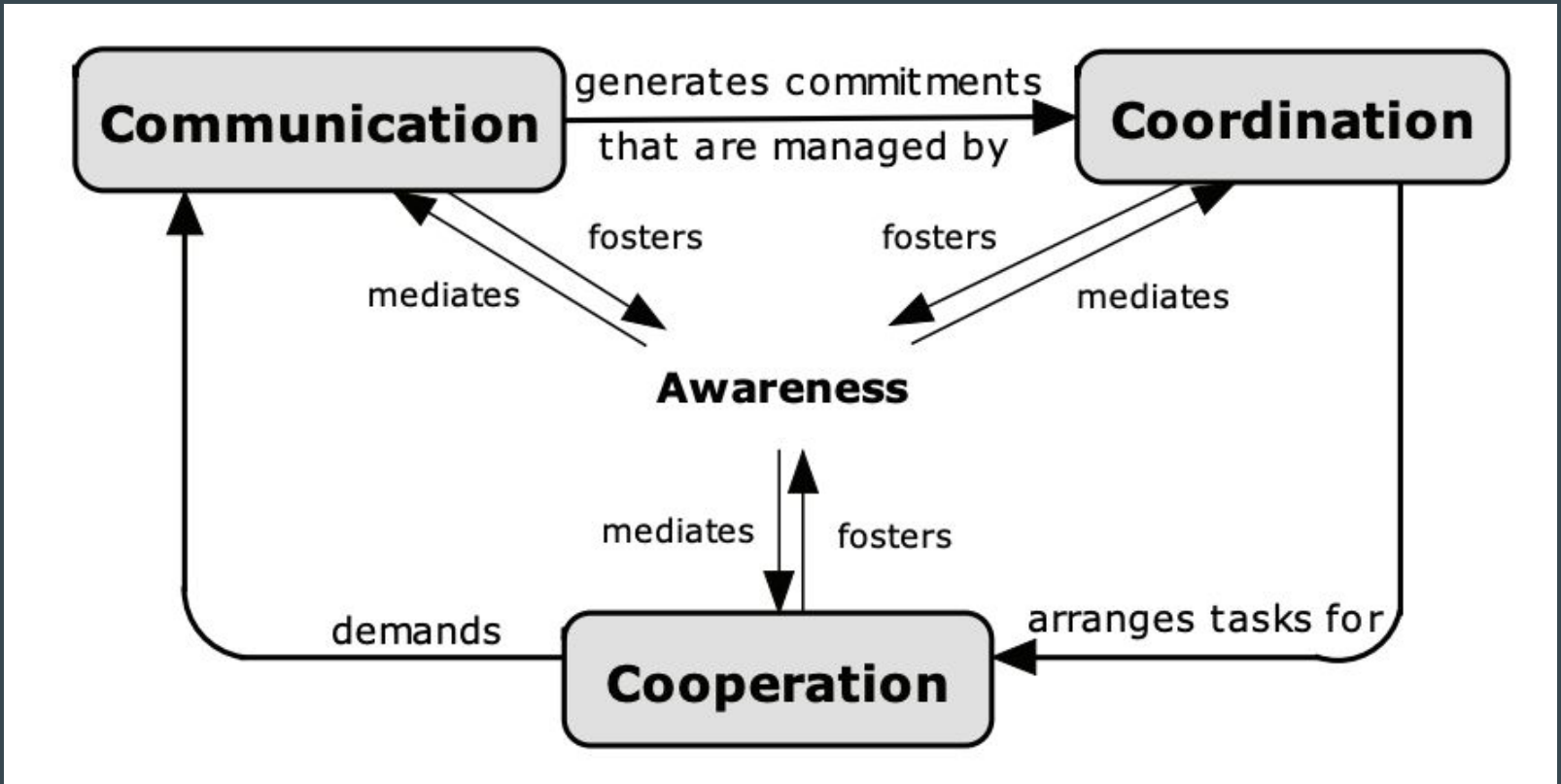
# Distributed Cognition and Software Development

Examples of **distributed cognition** across people in software development (pre GenAI):

Pair programming [Lui, 2006]
Code reviews [Sharp, 2006]
Design patterns [Mangalaraj, 2014]

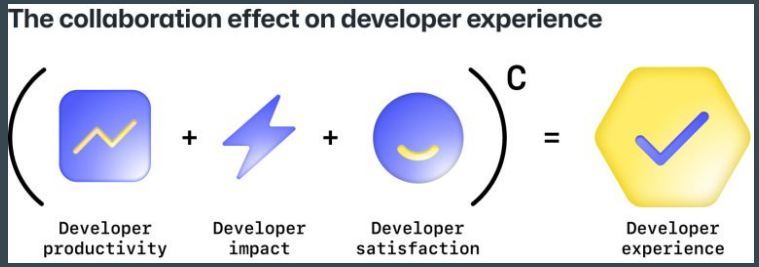
***Will the use of GenAI in software development impact how knowledge is distributed among team members? Will it impact the dynamics of team***

# 3C Model of Collaboration [Ellis, 1991] [Fuks, 2008]



# Collaboration is the multiplier for productivity and experience

Devs work with avg 21 other engineers, spend ~30% of their time talking to others, over half collaborate with other teams once or more per week [GitHub, 2023]




Many want **collaboration to be a** **top metric** in performance reviews
4 out of 5 devs think **GenAI** will make their **teams more collaborative** ! [GitHub, 2023]

***Are the number of code reviews or the number of pull requests good "metrics" for collaboration when GenAI is used?***
***Which of the 3C's do these measure?***

# Collaboration in software development is not just coding...

- ➔ GenAI shows potential for improving **meetings** (collaboration) [Microsoft, 2024]
- ➔ GenAI enhances performance, builds **expertise** and fosters **social connectivity** in design teams [Proctor&Gamble, 2025]

***Which theories should we use or build to understand how GenAI may shape meetings and social connections in SE?***

Flow

*The Future of Creative Problem Solving in Software Development*

Augmentation

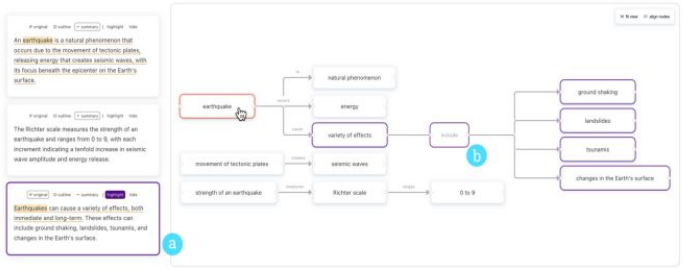
Collaboration

# Generative AI begs for a new wave of **natural UIs** beyond Chat

User Experience aka.ms/nfw

## UX, besides technical capabilities, is essential for AI to light up the next wave of tools for thinking. Post-chat UX and notebooks are taking a central stage.

- As discussed elsewhere in the deck, chat as an UX may be at a plateau (Morris 2024), as it can lack expressive power for both abstraction and specificity (Zamfirescu-Pereira et al., 2024). New and existing interaction patterns closer to direct manipulation are used for better expressing intent (Masson et al., 2024, Figma 2024) and parsing AI outputs (Jiang et al., 2023).
- Unlike chat interfaces, notebooks allow for a more structured, versatile, and familiar (Allen 2024) ways for people to create and consume knowledge. The written page and the notebook are re-emerging as a medium and the UX for thinking with AI: as companions to chat experiences (OpenAI 2024, Anthropic 2024) or on their own (Google 2024, Notion 2024).




The Graphologue interface, by Jieng et al (2023).


Morris, M. R., (2024). Prompting Considered Harmful. *CACM*.
Zamfirescu-Pereira, J.D., et al., (2024). Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. *CHI 2024*.
Masson, D., et al., (2024). DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models.
Figma, (2024). Figma AI.
Jiang, P., et al., (2023). Graphologue: Exploring Large Language Model Responses with Interactive Diagrams. *UIST 2023*.
Allen, R. (2024). The Notebook: A History of Thinking on Paper.
OpenAI, (2024). ChatGPT Canvas.
Anthropic, (2024). Claude Artifacts.
Google, (2024). Notebook M.
Notion, (2024). Notion AI.


***What tools can help us expand our creative problem solving***

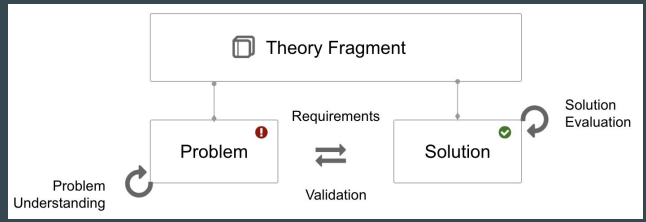
# Why a **theory-based approach** for designing and studying GenAI4SE tools?

- Provides a **conceptual framework** to guide what to look for or measure
- Supports **explanation and actionability**, not just descriptions
- Supports **comparisons** across tools and studies
- Provides **design guidance** with a focus on **humans**
- May support **critical reflection** (e.g., to justify or question assumptions and findings)

**Theory of Flow**

**Laws of Media**
**Cognitive Support Theories**
**Cognitive Augmentation**

**Distributed Cognition**
**3C Model of Collaboration**



*Remember how Google "changed our minds"....*
*Just imagine* how Generative AI (and Vibe) will impact how today's children will creatively problem solve tomorrow!

# Three Lenses to Understand how GenAI Supports *Creative Problem Solving in Software Development*

Flow Experiences

Cognitive Augmentation

Collaboration

Margaret-Anne Storey
https://www.linkedin.com/in/margaret-anne-storey-8419462/

# References - Theories


Brown, A., Chang, A., Holtz, B., D'Angelo, S., 2023. Developer Productivity for Humans, Part 6: Measuring Flow, Focus, and Friction for Developers. IEEE Software 40, 16–21. https://doi.org/10.1109/MS.2023.3305718

Cognition in the Wild [WWW Document], n.d. . MIT Press. URL https://mitpress.mit.edu/9780262581462/cognition-in-the-wild/ (accessed 4.24.25).

Cognitive Support In Software Engineering Tools: A Distributed Cognition Framework [WWW Document], n.d. . ResearchGate. URL https://www.researchgate.net/publication/200086146_Cognitive_Support_In_Software_Engineering_Tools_A_Distributed_Cognition_Framework (accessed 4.24.25).

Fuks, H., Raposo, A., Gerosa, M.A., Pimental, M., Lucena, C.J.P., 2008. The 3C Collaboration Model, in: Encyclopedia of E-Collaboration. IGI Global Scientific Publishing, pp. 637–644. https://doi.org/10.4018/978-1-59904-000-4.ch097

Kuusinen, K., Petrie, H., Fagerholm, F., Mikkonen, T., 2016. Flow, Intrinsic Motivation, and Developer Experience in Software Engineering, in: Sharp, H., Hall, T. (Eds.), Agile Processes, in Software Engineering, and Extreme Programming. Springer International Publishing, Cham, pp. 104–117. https://doi.org/10.1007/978-3-319-33515-5_9

Lange, E.M., 2024. Flow in the Age of AI : The Impact of AI-enabled Work Tools on IT Professionals and Design Recommendations.

Ma, Y., Huang, Y., Leach, K., 2024. Breaking the Flow: A Study of Interruptions During Software Engineering Activities, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, ICSE '24. Association for Computing Machinery, New York, NY, USA, pp. 1–12. https://doi.org/10.1145/3597503.3639079

Marchetti, F., Guastavino, S., Campi, C., Benvenuto, F., Piana, M., 2023. A comprehensive theoretical framework for the optimization of neural networks classification performance with respect to weighted metrics. https://doi.org/10.48550/arXiv.2305.13472

Meyer, A.N., 2019. Fostering software developer productivity through awareness increase and goal-setting. University of Zurich. https://doi.org/10.5167/UZH-174312

Salanova, M., Rodríguez-Sánchez, A.M., Schaufeli, W.B., Cifre, E., 2014. Flowing Together: A Longitudinal Study of Collective Efficacy and Collective Flow Among Workgroups. The Journal of Psychology.

Tankelevitch, L., Kewenig, V., Simkute, A., Scott, A.E., Sarkar, A., Sellen, A., Rintel, S., 2024. The Metacognitive Demands and Opportunities of Generative AI, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, CHI '24. Association for Computing Machinery, New York, NY, USA, pp. 1–24. https://doi.org/10.1145/3613904.3642902

Treude, C., Gerosa, M.A., 2025. How Developers Interact with AI: A Taxonomy of Human-AI Collaboration in Software Engineering. https://doi.org/10.48550/arXiv.2501.08774

Züger, M., Corley, C., Meyer, A.N., Li, B., Fritz, T., Shepherd, D., Augustine, V., Francis, P., Kraft, N., Snipes, W., 2017. Reducing Interruptions at Work: A Large-Scale Field Study of FlowLight, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. Presented at the CHI '17: CHI Conference on Human Factors in Computing Systems, ACM, Denver Colorado USA, pp. 61–72. https://doi.org/10.1145/3025453.3025662

# References - Industry studies of GenAI in software dev


AI | 2024 Stack Overflow Developer Survey [WWW Document], n.d. URL https://survey.stackoverflow.co/2024/ai#sentiment-and-usage (accessed 4.24.25).

AI and the New Future of Work, n.d. . Microsoft Research. URL https://www.microsoft.com/en-us/research/academic-program/ai-and-the-new-future-of-work/ (accessed 4.24.25).

Brown, A., Chang, A., Holtz, B., D'Angelo, S., 2023. Developer Productivity for Humans, Part 6: Measuring Flow, Focus, and Friction for Developers. IEEE Software 40, 16–21. https://doi.org/10.1109/MS.2023.3305718

Butler, J., Suh, J., Haniyur, S., Hadley, C., 2024. Dear Diary: A randomized controlled trial of Generative AI coding tools in the workplace. https://doi.org/10.48550/arXiv.2410.18334

Dell'Acqua, F., Ayoubi, C., Lifshitz-Assaf, H., Sadun, R., Mollick, E.R., Mollick, L., Han, Y., Goldman, J., Nair, H., Taub, S., Lakhani, K.R., 2025. The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise. https://doi.org/10.2139/ssrn.5188231

Dunay, O., Cheng, D., Tait, A., Thakkar, P., Rigby, P.C., Chiu, A., Ahmad, I., Ganesan, A., Maddila, C., Murali, V., Tayyebi, A., Nagappan, N., 2024a. Multi-line AI-Assisted Code Authoring, in: Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering, FSE 2024. Association for Computing Machinery, New York, NY, USA, pp. 150–160. https://doi.org/10.1145/3663529.3663836

Dunay, O., Cheng, D., Tait, A., Thakkar, P., Rigby, P.C., Chiu, A., Ahmad, I., Ganesan, A., Maddila, C., Murali, V., Tayyebi, A., Nagappan, N., 2024b. Multi-line AI-assisted Code Authoring. https://doi.org/10.48550/arXiv.2402.04141

Hicks, C., Lee, C., Foster-Marks, K., 2024. The New Developer: AI Skill Threat, Identity Change & Developer Thriving in the Transition to AI-Assisted Software Developmen. https://doi.org/10.31234/osf.io/2gej5

Kalliamvakou, E., 2022. Research: quantifying GitHub Copilot's impact on developer productivity and happiness. The GitHub Blog. URL https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/ (accessed 4.24.25).

Khemka, M., Houck, B., 2024. Toward Effective AI Support for Developers: A survey of desires and concerns. Commun. ACM 67, 42–49. https://doi.org/10.1145/3690928

Staff, I.S., GitHub, 2023. Survey reveals AI's impact on the developer experience. The GitHub Blog. URL https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/ (accessed 4.24.25).

Unleash developer productivity with generative AI | McKinsey [WWW Document], n.d. URL https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai?utm_source=chatgpt.com (accessed 4.24.25).

# References cont.


Ritonummi et al. “Exploring Barriers That Prevent Employees from Experiencing Flow in the Software Industry | Emerald Insight.” *Journal of Workplace Learning* 36, no. 3 (n.d.): 223–38.

Forsgren, Nicole, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. “The SPACE of Developer Productivity: There’s More to It than You Think.” *Queue* 19, no. 1 (February 28, 2021): 20–48. https://doi.org/10.1145/3454122.3454124.

Janssens, Sabine, and Vadim Zaytsev. “Go with the Flow: Software Engineers and Distractions.” In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 934–38. Montreal Quebec Canada: ACM, 2022. https://doi.org/10.1145/3550356.3559101.

Petersen “Measuring the Flow in Lean Software Development - Petersen - 2011 - Software: Practice and Experience - Wiley Online Library.” Accessed May 9, 2025. https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.975.

Meyer, André N. “Fostering Software Developer Productivity through Awareness Increase and Goal-Setting.” University of Zurich, 2019. https://doi.org/10.5167/UZH-174312.

Sweller, John. “Cognitive Load During Problem Solving: Effects on Learning.” *Cognitive Science* 12, no. 2 (April 1988): 257–85. https://doi.org/10.1207/s15516709cog1202_4.