

What if Offloading Is Not the Problem?

Ilya Zakharov

Human-AI Experience team

JetBrains Research

ilia.zakharov@jetbrains.com

Abstract

Debates about generative AI in software engineering often center on cognitive offloading and the possibility that developers will lose skills. This position paper argues that this focus is important but too narrow. Evidence of durable cognitive decline remains mixed, and offloading can involve task substitution, skill displacement, skill decay, or failures of skill formation. A more useful unit of analysis is the changing human–AI work system, including its metacognitive, social, motivational, and organizational conditions. Drawing on research on extended cognition, learning, teamwork, motivation, automation, and developer productivity, the paper examines how GenAI may reshape judgment, novice learning, shared understanding, communication, ownership, accountability, professional identity, status, and the distribution of productivity gains. It argues that the central question is not whether developers should avoid offloading, but which capabilities and forms of agency human–AI systems should preserve and cultivate, for whom, and under whose goals. Evaluations of AI-assisted work should therefore extend beyond task-level speed to learning, team knowledge, meaningful work, well-being, and the allocation of any time and effort that AI releases.

1. The Fear of Offloading

One of the dominant fears surrounding AI-assisted programming is that, with work delegated to AI, developers may gradually lose their professional skills. Such a process of deskilling is often seen as an outcome of cognitive offloading, the act of reducing the mental processing requirements of a task by using external tools instead (Risko & Gilbert, 2016). Cognitive offloading is both common and consequential. Search engines have changed what people remember, and today people are more likely to retain where information can be found than the information itself (Sparrow, Liu, & Wegner, 2011). Access to online explanations inflates estimates of one’s own knowledge, creating the illusion of understanding (Fisher, Goddu, & Keil, 2015). In a survey of 319 knowledge workers, greater confidence in generative AI was associated with less self-reported critical thinking, while confidence in one’s own ability was associated with more; participants described critical thinking as shifting toward verification, integration, and stewardship of AI output (H.-P. Lee et al., 2025).

The risks of cognitive offloading are especially vivid in software engineering. The danger here is not merely that a particular skill becomes rusty; it is that the pipeline by which people become capable of judgment is interrupted just as judgment becomes the skill the organization claims to need most. Today, there is a growing fear among industry practitioners that an experienced developer who reviews an expanding stream of machine-produced changes may gradually become a rubber stamp for work they no longer understand well enough to assess (a risk sometimes called “comprehension debt”, Aragwal et al., 2026).

In this position paper, however, I want to claim that while cognitive offloading is a real risk and is likely already happening, the more important shift may be motivational: not only what developers can still do, but what they will still want to do (and why). When people discuss cognitive offloading, they often imply that the individual brain (and cognition) is the correct unit of analysis and that preserving the previous distribution of skills is the proper objective. I want to claim that, first, the relevant unit of analysis is often not the unaided developer but a changing human–AI team environment, and, second, that the central risk is not simply capability loss but a reconfiguration of motivation, ownership, accountability, and the social routes through which expertise is formed. I also want to raise the question of where the

resources released due to cognitive offloading might (and should) go. The potential benefits depend on the organizational allocation of the resources AI releases. Time saved can become learning, care, exploration, and better judgment, or merely a higher throughput target. The central question is therefore not “How do we prevent developers from offloading?” but rather: **“Which cognitive, motivational, social, and organizational capacities should a human–AI work system cultivate, for whom, and under whose goals?”**

2. Offloading Is an Outcome, Not a Diagnosis

The term *offloading* can include several distinct phenomena (ref). It can mean **task substitution**: a person no longer performs an activity, such as a software engineer recalling an API or writing a routine transformation. It can also mean **skill displacement**: practice in one skill decreases while another skill (prompting, evaluating, integrating, or coordinating) grows. Cognitive offloading can also lead to different outcomes. One is **skill decay**: a capability previously held by the person becomes less available through disuse. Another is **failure of skill formation**: a novice never develops the representations required for independent performance or evaluation. Other outcomes can include **miscalibration** (the person mistakes the joint system’s competence for their own or cannot identify when assistance is unreliable) or **loss of agency** (the person retains nominal responsibility but no longer meaningfully selects goals, methods, or acceptance criteria).

Generative AI is already being used to offload large parts of knowledge work. One of the strongest studies to date surveyed 319 knowledge workers across 936 concrete AI-use examples: greater confidence in GenAI was associated with less reported critical thinking, while the work shifted toward checking, integrating, and supervising AI output (H.-P. Lee et al., 2025). A cluster-randomized classroom study (Lindo & Cutad, 2024) found that structured GenAI use increased reported offloading and that greater offloading correlated with worse two-week retention (yet the AI condition itself had no reliable direct effect on delayed retention). In another study, across three groups (GenAI-aided, search-engine-aided, and unaided), GenAI users reported lower ownership and had more trouble quoting their own essays (Kosmyrna et al., 2025). Moreover, participants showed weaker and less distributed brain activation (measured as EEG connectivity) in the GenAI group than in search-engine or unaided groups. However, it should be noted that the main part of the experiment involving EEG recording included only 18 people, with the measures being task-specific proxies rather than a direct test of lasting ability, so the results should be treated as preliminary.

There is also a contradictory set of results. Two 2025 video-learning experiments (with 101 and 215 participants; Şimşek, Anders, Göth, Specht, & Huff, 2025) found no retention or transfer difference between GPT summaries and ordinary transcripts. A small randomized medical-student study likewise found higher immediate quiz performance with ChatGPT but no retention advantage one week later (Kalam et al., 2025). Some research has also shown benefits for the creativity of outputs generated with GenAI. The most directly positive results come from the study by B. C. Lee and Chung (2024). Across five experiments on everyday and innovation problems, people using GPT-3.5 generated ideas rated more creative than people working unaided or with conventional web search. The gain held across both low- and high-constraint problems, but was stronger for **incremental** novelty than for radical innovation; the authors attribute it partly to the model’s ability to connect distant concepts into coherent proposals. However, results for creativity are as mixed as results for cognition. An experiment with 60 participants (Wadinambiarachchi, Kelly, Pareek, Zhou, & Velloso, 2024) using an AI image generator during visual ideation found a decrease in creative output: greater fixation on an initial example, with fewer, less varied, and less original ideas. A larger 2025 randomized priming study (387 students, Medeiros, Cropley, Marrone, & Reiter-Palmon, 2025) likewise found no broad divergent-thinking advantage from ChatGPT examples over human examples; low-creativity ChatGPT prompts actually reduced performance.

Overall, it seems too early to establish from the available data that ordinary AI use causes durable detrimental effects on human skills and learning. However, given the historical precedents, I would say that we can state that cognitive offloading is likely to be happening. Before the invention of the printing

press, European education treated memory as a deliberately trained intellectual skill. Medieval scholars used spatial mnemonics, repeated recitation, and highly structured manuscripts to internalize material for preaching, argument, and composition. Mary Carruthers describes memory as one of the central instruments of medieval intellectual life, rather than merely a passive store of information (Carruthers, 2008). Although the available evidence is circumstantial, research supports the idea that as inexpensive, standardized books became dependable external stores, elaborate mnemonic training gradually declined as it lost much of its practical necessity (Rachal, 1988). A much more recent example of cognitive offloading is known as the “Google effect”: a shift from remembering *what* to remembering *where it can be found* (Sparrow et al., 2011). In one of the first studies on this topic, participants read and typed trivia statements. Those who expected the computer to erase the statements later recalled approximately 29–31% of them, whereas those who believed the statements had been saved recalled only 19–22%. Explicitly telling participants to remember the statements did not eliminate the effect. In another experiment, participants subsequently recalled the location where information had been stored better than the information itself: mean recall was 0.49 for location versus 0.23 for content (although the authors noted that location recall received a stronger cue).

Cognitive offloading itself is not new. Although some skills have already been lost, new skills have replaced them. The fear of cognitive offloading often comes hand-in-hand with a larger and weakly supported claim: that AI use is leading to the overall decline of human intelligence. However, evidence of declines in task-specific knowledge, effort, or even calibration does not imply that general intelligence is declining. Human intelligence is not a warehouse of isolated procedures. It includes learning relational structure, connecting representations, selecting goals, and adapting action to unfamiliar conditions. According to recent advances in biological theories of intelligence, it is based on a set of interconnected functional nodes that probably combine various skills and abilities into a complex network (Santarnecchi, Galli, Polizzotto, Rossi, & Rossi, 2014; Barbey, 2018; Zakharov, Tabueva, Adamovich, Kavas, & Malykh, 2020). While tools can atrophy particular routines, they can simultaneously enable much more complex activity. The normative question is not whether the pre-AI repertoire of skills remains intact in every person, but rather whether the emerging repertoire of new skills supports competent, autonomous, resilient, and meaningful action.

One of the hardest questions related to cognitive offloading due to GenAI may be how novices acquire new skills. Experts can often delegate an activity because prior experience lets them recognize implausible output and retrieve a deeper model when needed. Novices, however, risk receiving the product without the formative process. A within-subject study of graduate CS students working in a legacy code base found that Copilot improved task productivity but did not improve code-base comprehension (Qiao, Shihab, Haque, & Hundhausen, 2026). The authors also observed that faster performance was associated with poorer reverse-engineering comprehension. These results raise questions about student education. What can be learned through assisted performance? Which forms of struggle are diagnostically or developmentally productive, and which are merely friction? A design that is appropriate for an expert may be actively harmful to a beginner. However, it seems that such systems can be designed with AI in mind. A large randomized field experiment with nearly 1,000 high-school students (Bastani et al., 2025) demonstrated that access to a conventional GPT-4 interface improved performance on practice mathematics problems by 48%, but when AI was removed, these students scored 17% worse than students who had never used it. Crucially, they did not perceive that they had learned less. However, a tutor-style AI system that withheld complete answers and supplied progressive hints largely eliminated the negative effect (Bastani et al., 2025). This shows that, with a carefully designed approach to training novices (one that is yet to be firmly established) expertise can be developed.

There is at least one more important AI-related danger, which lies in the phenomenon of the illusion of explanatory depth. Overall, people believe they understand mechanisms (such as toilets, zippers, or helicopter flight) until they are asked to explain them step by step (Rozenblit & Keil, 2002). The illusion of understanding is also related to the perception of the availability of information. After searching online for explanations, people rated their own ability to explain unrelated topics more highly. They

appeared to partially mistake access to external knowledge for knowledge located in their own minds (Fisher et al., 2015). An important and real risk here is that easy access to AI tools can allow people to produce correct-looking work while overestimating how much they themselves understand.

So, we can claim that GenAI can accelerate cognitive offloading while trading task-specific skills for new coordination abilities (taking into account the risk of leaving novices without foundational knowledge and the risk of creating a dangerous illusion of personal understanding). With cautious optimism, we might also expect GenAI to contribute to an evolution of human intelligence rather than simply to its decline.

3. Extended Cognition and Metacognition

The image behind much offloading anxiety is a developer whose authentic cognition takes place inside the skull and whose tools subtract from that internal activity. This is historically and theoretically narrow. Vygotsky (1978) described higher mental functions as mediated by cultural tools and signs. Hutchins (1995) showed how cognition can be distributed across people, artifacts, and coordinated procedures. Clark and Chalmers (1998) argued that when external resources are reliably integrated into thought and action, they can be parts of a cognitive system rather than merely aids to an otherwise self-contained mind.

Programming has always made this point concrete. A developer thinks with notation, documentation, issue trackers, version history, debuggers, tests, code review, and colleagues (Ehn, 1990). These resources do not simply replace internal operations. They alter which internal operations are useful. Version control changes the value of remembering every prior state; a debugger changes how hypotheses can be tested; a high-level language makes some machine details less salient while making larger abstractions tractable. The resulting competence belongs partly to a coordinated arrangement.

Clark (2025) applies this perspective to generative AI. Human brains are unusually capable of adapting their own contributions to the opportunities created by external resources. The result of such adaptation can be a new, interwoven whole in which the person, interface, model, artifacts, and social setting continually change one another's roles. This view does not imply that every convenient tool is literally part of the mind, or that dependence is harmless, but it does change the evaluative target. Instead of measuring only what the unaided individual retains, we might consider the complex system that includes both intelligent tools and human decisions. For SE-related questions, evaluation might include **calibration** (whether people know when and why to trust an output), **contestability** (whether they can challenge, redirect, or override the AI output), **resilience** (whether work can continue when the model is wrong or unavailable), or even inter-individual phenomena, such as a team's **distribution of skills** (whether competence and control are spread across a team or concentrated in a vendor, model, or small group of experts).

Introducing new tools into human activity has been shown to alter human behavior in multiple areas. After superhuman AI entered Go, human players produced moves that were both of higher quality and more novel; the novelty was not reducible to copying the machine's distinctive moves (Shin, Kim, van Opheusden, & Griffiths, 2023). In a field experiment at Procter & Gamble, individuals working with AI performed comparably to conventional two-person teams and produced solutions that crossed established technical and commercial boundaries (Dell'Acqua et al., 2025). Generative AI can also increase the judged creativity of individual stories, especially for less creative writers, although it simultaneously makes the collection of stories more similar (Doshi & Hauser, 2024). In a different domain, personalized conversations with language models reduced belief in emerging conspiracy theories more than control interactions or static corrective text, with some effects persisting for one to two months (Costello, Rabb, Stagnaro, Pennycook, & Rand, 2026). The latter is a recent preprint outside software engineering, but it illustrates the theoretical possibility that AI can reinforce revisions to users' current knowledge.

There are, of course, objections to the idea that AI can meaningfully extend our cognition. Pritchard (2026) argues that while traditional technologies normally transform domain-specific skills, agentic AI

can take over activities through which executive intellectual abilities and character are exercised, leading to a decrease in skill formation without any corresponding change in human executive functions. This is an important hypothesis, but it should not be accepted as a definitional truth. The boundary between “domain” and “executive” cognition is neither fixed nor technologically determined. Planning, monitoring, and revision are learned in interaction with tools, and while AI can be configured to suppress these activities, it can also expose alternatives, solicit commitments before acting, or make uncertainty actionable. A good example of a system that enhances our judgment comes from Sarkar (2024), who proposes recasting AI from assistant to provocateur: a system that questions assumptions and offers counterarguments rather than obediently completing the task. In an experimental instantiation of this idea, brief AI-generated “provocations” that critiqued and proposed alternatives to AI suggestions during a shortlisting task induced critical and metacognitive thinking relative to plain AI assistance (Drosos, Sarkar, Xu, & Toronto, 2025).

However, the risk that cognitive offloading places too much burden on executive functions is real. As code becomes cheaper to produce, it can proliferate faster than it can be understood, shifting scarce effort toward choosing worthwhile problems, connecting technical decisions to user and business consequences, integrating changes into an evolving system, verifying behaviour, and preserving a shared account of why the system is as it is. This is often described as a move “from coding to judgment,” but judgment is not a residual capacity that appears automatically when implementation is automated. It depends on learned discrimination, a working model of the system, calibrated uncertainty, comparison among alternatives, and responsibility for the outcome. If organizations expand output as generation becomes cheaper, developers gain no spare capacity for this work; they simply receive more artifacts to inspect. More of the burden therefore moves from object-level production to metacognitive control: monitoring AI output, calibrating confidence in both the model and one’s own understanding, and deciding when scrutiny is sufficient (H.-P. Lee et al., 2025; Ackerman & Thompson, 2017). This may be frustrating rather than liberating. Continuous self-monitoring consumes limited cognitive resources (Safati, Seli, & Smilek, 2026), sustained monitoring of automated systems is effortful and stressful (Warm, Parasuraman, & Matthews, 2008), and mental effort is reliably associated with negative affect across tasks (David, Vassena, & Bijleveld, 2024). The result may be less cognitive relief than a new, less visible form of strain.

So, the risks of cognitive offloading, deskilling, novice learning, and new burdens on metacognition are real. I would argue, however, that the current preoccupation with GenAI’s cognitive risks may divert attention from other, arguably more consequential ones.

4. GenAI Tools and Software Teams

One area of change that is overlooked in discussions of GenAI’s effects on software engineering is how it can affect software teams. Margaret-Anne Storey (2026) has recently proposed additions to the well-known notion of technical debt. The term technical debt is used to describe the additional cost of maintaining a system that may arise when one chooses an expedient solution to accelerate its development. Technical debt is a property of the technical artifact, the system that is being built. The idea of cognitive debt extends future costs and deficits to a team’s shared understanding of the system. Cognitive debt might accumulate alongside intent debt (the loss of goals, constraints, and rationale) that led to the creation of the technical artifact in the first place. It is important to notice that both cognitive and intent debt are, to a considerable extent, interpersonal. A codebase can compile and pass generated tests while no team member can explain its invariants or why a tradeoff was accepted, leading to potentially catastrophic problems when some part of the system breaks. Thus, retrieving information about architectural solutions is becoming one of the bottlenecks in contemporary software engineering. Rising communication barriers are already among the managerial concerns associated with technological change. Microsoft’s natural-experiment study of 61,182 US employees (Yang et al., 2022) found that the firm-wide move to remote work produced more static and siloed collaboration networks, fewer bridges across structural groups, and a shift toward more asynchronous communication, potentially leading to

a decline in innovation within the company. At the same time, a recent systematic review of 39 peer-reviewed studies on LLM assistants and developer productivity found that communication and team activity remain largely underexplored, with little longitudinal or team-level evidence (Mohamed, Assi, & Guizani, 2026).

But the move from building with a colleague to building with GenAI can change more than information retrieval. Asking another developer a “small” question can transmit tacit context, reveal who knows what, establish permission to ask again, and make uncertainty visible to the team. Asking AI may be faster, less embarrassing, and available at any hour, and it can democratize access to explanation. But if many such exchanges disappear, the organization may lose the weak ties through which trust, mentorship, and situation awareness develop. In a study of 33 student teams, groups with an AI teammate showed lower human-to-human responsiveness and lower reported belonging and status than all-human groups; greater AI dominance was associated with participants feeling less valued (Nixon et al., 2026). The study warns that while a GenAI tool can increase the number of useful utterances or the speed of a solution, it can also decrease the social conditions under which people learn and remain willing to contribute. For software engineering, cognitive debt may therefore be partly a network property. If AI answers questions privately, the answer may never enter the team’s shared memory. The immediate exchange is efficient, while the organization becomes less aware of itself.

5. Offloading and Motivation

Another weakness in the offloading debate is that it often treats developers as information processors whose main relationship to work is cognitive demand. Cat Hicks (2026) calls this the “brains-in-jars” view: developers are separated from the social, emotional, and organizational conditions in which software is actually made. Once the problem is framed this way, it is easy to overlook why people become developers and why they remain in the profession.

People work in software engineering for many reasons: the pleasure of making something, technical challenge, problem-solving, usefulness, autonomy, recognition, and participation in a community. Beecham, Baddoo, Hall, Robinson, and Sharp (2008) found recurring roles for problem-solving, meaningful or beneficial work, and interest in the work itself, while stressing that motivation varies across contexts. A more recent survey of 521 developers similarly found that enjoyment, ownership, learning, challenge, importance, community, and recognition all mattered, but that no single motive was sufficient (Amit & Feitelson, 2026). Self-determination theory provides a broader explanation: autonomy, competence, and relatedness support sustained, high-quality motivation and well-being at work (Ryan & Deci, 2000; Gagné et al., 2022). Developer-specific research also links agency, motivation, a learning culture, and belonging with self-reported productivity (C. M. Hicks, Lee, & Ramsey, 2024).

Church and Marasoiu (2025), in a deliberately provocative opinion paper, argue that software organizations should treat having a good time at work as a legitimate aim rather than translate every humane concern into the language of productivity. Enjoyment cannot settle questions of safety or responsibility. But a future of programming that is efficient while draining the work of curiosity, inquiry, and inspiration would still be a design failure.

Recent studies of developer preferences support this account. In a mixed-methods study of 860 developers, Choudhuri, Badea, et al. (2026) found strong use of and demand for AI in coding and testing, as well as demand for reducing toil in documentation and operations. Developers drew firmer boundaries around work tied to identity and relationships, such as mentoring. Their responsible-AI priorities also varied by task: reliability and security mattered most for system-facing work; transparency, alignment, and steerability mattered where control was central; and fairness and inclusion mattered for human-facing tasks. These were not general attitudes toward “AI,” but judgments about particular forms of support in particular tasks. A second study of 448 professional developers sharpened this point. Task identity (the extent to which developers identified with and intrinsically enjoyed a task) predicted less willingness to give AI final decision-making authority. Accountability reduced willingness to let AI produce work on the developer’s behalf, while high task demands made delegation of decisions more

acceptable (Choudhuri, Bird, Badea, Gerosa, & Sarma, 2026). Resistance to AI is therefore not simply resistance to change. It often reflects what developers find meaningful, what they feel responsible for, and which parts of the work they regard as their own.

AI changes developers' authorship and agency. Contribution, understanding, control, and responsibility were never perfectly aligned in software engineering, but GenAI can pull them further apart. A developer may define the goal without writing the implementation, understand the intended behavior without understanding the mechanism, approve a change without controlling how it was produced, and remain accountable when it fails. Elish's (2019) concept of the *moral crumple zone* describes situations in which a human absorbs blame for an automated system despite having little meaningful control. Recent interviews with experienced developers using software agents show that meaningful oversight involves more than final approval. It includes setting constraints, planning with the agent, monitoring its execution, and reviewing the result. Developers also reported difficulty assessing agent-generated code and sometimes relied on imperfect heuristics to decide whether it was acceptable (Dhanorkar, Passi, & Vorvoreanu, 2026). Human-in-the-loop design is therefore not automatically human-centered. If the human's practical role is only to approve output under time pressure, the loop may preserve liability without preserving agency. Together, these changes may reorganize not only how software is produced, but also which kinds of work carry identity, authority, and status.

6. Social Hierarchy and Offloading

Software engineering often presents itself as a meritocracy: skill can be demonstrated through working artifacts, difficult problems, and visible contributions. AI adoption can also affect the social structure of software engineering. On the one hand, AI can widen participation by lowering some barriers. In three randomized field experiments involving 4,867 developers, access to an AI coding assistant increased the number of completed tasks on average, with larger gains among less experienced developers (Cui et al., 2026). The finding concerns task completion rather than the quality of every resulting artifact, but it supports the possibility that AI can help some developers overcome practical barriers to production. At the same time, AI may produce a Matthew effect: people with stronger prior knowledge, better access, and greater organizational power may be better able to direct, verify, and integrate its output. A recent study of open-source projects found that GitHub Copilot increased participation and code contributions overall, but peripheral contributors gained less and faced larger increases in coordination work than core contributors (Song, Agarwal, & Wen, 2026, preprint). A similar pattern has been observed outside software engineering: AI-assisted translation increased content production across Wikipedia language communities, but the largest gains went to communities that already had the strongest resources (Zhu & Walker, 2026). AI can therefore broaden participation while also allowing established groups to capture disproportionate benefits.

For many developers, the ability to demonstrate skill through working artifacts is also a source of professional pride. If routine code no longer signals mastery, status may shift toward architecture, taste, problem selection, system judgment, domain knowledge, or the ability to coordinate people and agents across a project. Some of these changes may be valuable. They may also create new barriers whose criteria are less visible and harder for newcomers to acquire.

7. Where Do the Saved Resources Go?

Perhaps the most neglected question in debates about offloading is what happens to the productivity dividend. If AI makes a task 30% faster, the released time could support reflection, more ambitious problems, contact with users, learning, mentorship, or recovery. It could instead be absorbed by higher output targets, a larger review queue, or a smaller team. Which outcome follows depends less on the tool than on organizational choices about workload, staffing, incentives, and performance measures.

Before deciding how saved time should be used, however, we must establish that time was saved at all. In a preregistered study of 1,237 participants completing short cognitive tasks, Yu et al. (2026) found no overall difference between unaided and AI-assisted completion times. AI reduced reported

effort, and participants expected it to be faster; it produced a measurable speed gain only on the more difficult tasks. The authors call this mismatch a speedup illusion. The tasks were brief and should not be treated as a model of professional programming. Still, a small randomized study of experienced open-source developers found a similar gap: participants were 19% slower with AI while expecting and (later believing) that it had made them faster (Becker, Rush, Barnes, & Rein, 2025, preprint). These results should not be generalized, but the *speedup illusion* illustrates the possibility that AI can feel easier without being faster. Task-level speed is also too narrow a measure of value. A dashboard showing faster pull-request completion does not tell us whether quality, rework, system understanding, learning, collaboration, or well-being improved. As Goodhart’s law suggests, once a measure becomes a target, it can reorganize the practice it was meant to observe (Goodhart, 1975; Strathern, 1997). The relevant outcome is not whether an AI feature saves minutes in isolation but whether the surrounding system converts those minutes into capabilities and conditions that people value. This question must be asked at several levels: individual workload and learning, team communication and mentorship, organizational staffing and incentives, and the profession’s entry routes and educational opportunities.

AI may also move some developers closer to problem framing, users, and business decisions. This should not be framed as a sudden move away from “real” software engineering. Professional software work has never consisted only of writing code. In one observational study, coding accounted for 21% of monitored work time, while substantial time went to communication, information seeking, planning, and other collaborative activities (Meyer, Barton, Murphy, Zimmermann, & Fritz, 2017). Value-based software engineering likewise treats stakeholder perspectives and trade-offs between short- and long-term goals as central concerns of the discipline (Salleh, Mendes, Mendes, Lekamlage, & Petersen, 2023). If implementation becomes cheaper, requirements, architecture, integration, user understanding, and product judgment may become relatively more important (this is, however, just a plausible direction, not yet a demonstrated industry-wide outcome). Some developers may not welcome it, but organizations will need to teach and reward the required capabilities rather than assume that they appear automatically.

Reduced working time is one concrete way to distribute productivity gains. Fan, Schor, Kelly, and Gu (2025) studied a six-month four-day-week intervention with no reduction in pay, involving 2,896 employees in 141 organizations and 12 control companies. Burnout, job satisfaction, mental health, and physical health improved; reduced fatigue and sleep problems and better self-reported work ability partly explained these gains. The study was not about software development or AI, and it did not test the quality of technical decisions, but it nevertheless shows that reducing paid working time without reducing pay can support recovery and well-being, which might, in turn, benefit decision-making. Research on human–automation systems shows that sustained monitoring is effortful and stressful, while a recent experiment found that continuous self-monitoring impaired concurrent task performance (Warm et al., 2008; Safati et al., 2026). Better judgment is unlikely to come from maximizing throughput alone. Overall, better decisions are unlikely to come from increased speed, but rather from slowing down.

8. Conclusion

Cognitive offloading is a real concern, but it has become the default lens through which AI-assisted software work is discussed. That lens is too narrow. The more consequential changes may concern motivation, shared understanding, agency, accountability, status, and the use of any time that AI saves. A system can preserve an individual skill while damaging mentorship or ownership; it can also replace a routine while enabling more ambitious and meaningful work. The relevant question is therefore not simply what developers may forget, but what kinds of capabilities, relationships, and working conditions human–AI systems should sustain. A modest next step is for researchers and organizations to evaluate AI adoption with a few measures beyond output speed (for example, learning, team understanding, ownership, and well-being) and to reserve some of any released capacity for reflection, mentoring, and skill development. This would broaden the debate without dismissing the genuine cognitive risks.

9. References

Ackerman, R., & Thompson, V. A. (2017). Meta-reasoning: Monitoring and control of thinking and

- reasoning. *Trends in Cognitive Sciences*, 21(8), 607–617. doi: 10.1016/j.tics.2017.05.004
- Amit, I., & Feitelson, D. G. (2026). A large scale survey of motivation in software development. *Journal of Systems and Software*, 231, 112596. doi: 10.1016/j.jss.2025.112596
- Barbey, A. K. (2018). Network neuroscience theory of human intelligence. *Trends in Cognitive Sciences*, 22(1), 8–20. doi: 10.1016/j.tics.2017.10.001
- Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakçı, Ö., & Mariman, R. (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences*, 122(26), e2422633122. doi: 10.1073/pnas.2422633122
- Becker, J., Rush, N., Barnes, E., & Rein, D. (2025). *Measuring the impact of early-2025 AI on experienced open-source developer productivity*. (Preprint) doi: 10.48550/arXiv.2507.09089
- Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, 50(9–10), 860–878. doi: 10.1016/j.infsof.2007.09.017
- Carruthers, M. (2008). *The book of memory: A study of memory in medieval culture* (2nd ed.). Cambridge: Cambridge University Press. doi: 10.1017/CBO9781107051126
- Choudhuri, R., Badea, C., Bird, C., Butler, J. L., DeLine, R., & Houck, B. (2026). AI where it matters: Where, why, and how developers want AI support in daily work. In *Proceedings of the 48th IEEE/ACM international conference on software engineering: Software engineering in practice (icse-seip 2026)*. New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3786583.3786885
- Choudhuri, R., Bird, C., Badea, C., Gerosa, M., & Sarma, A. (2026). *You shall not pass! where and why developers draw the line on AI autonomy*. (Preprint) doi: 10.48550/arXiv.2607.00533
- Church, L., & Marasoiu, M. (2025). It's OK to want to have a good time. In *Companion proceedings of the 9th international conference on the art, science, and engineering of programming (programming 2025)* (Vol. 134, pp. 5:1–5:4). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/OASICS.Programming.2025.5
- Clark, A. (2025). Extending minds with generative AI. *Nature Communications*, 16, 4627. doi: 10.1038/s41467-025-59906-9
- Clark, A., & Chalmers, D. J. (1998). The extended mind. *Analysis*, 58(1), 7–19. doi: 10.1093/analys/58.1.7
- Costello, T. H., Rabb, N., Stagnaro, M. N., Pennycook, G., & Rand, D. G. (2026). *Reducing belief in conspiracy theories as they unfold using large language models*. (Preprint) doi: 10.48550/arXiv.2608.06151
- Cui, Z. K., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2026). The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Management Science*. (Published online) doi: 10.1287/mnsc.2025.00535
- David, L., Vassena, E., & Bijleveld, E. (2024). The unpleasantness of thinking: A meta-analytic review of the association between mental effort and negative affect. *Psychological Bulletin*, 150(9), 1070–1093. doi: 10.1037/bul0000443
- Dell'Acqua, F., Ayoubi, C., Lifshitz-Assaf, H., Sadun, R., Mollick, E., Mollick, L., ... Lakhani, K. R. (2025). *The cybernetic teammate: A field experiment on generative AI reshaping teamwork and expertise* (Working Paper No. 33641). National Bureau of Economic Research. (A revised version was published in Organization Science in 2026) doi: 10.3386/w33641
- Dhanorkar, S., Passi, S., & Vorvoreanu, M. (2026). *Human oversight of agentic systems in practice: Examining the oversight work, challenges, and heuristics of developers using software agents*. (Preprint; accepted at FAccT 2026) doi: 10.48550/arXiv.2606.05391
- Doshi, A. R., & Hauser, O. P. (2024). Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science Advances*, 10(28), eadn5290. doi: 10.1126/sciadv.adn5290
- Drosos, I., Sarkar, A., Xu, X., & Toronto, N. (2025). *“it makes you think”: Provocations help restore critical thinking to AI-assisted knowledge work*. (Preprint) doi: 10.48550/arXiv.2501.17247

- Ehn, P. (1990). *Work-oriented design of computer artifacts*. Hillsdale, NJ: Lawrence Erlbaum Associates. (The original Scandinavian edition is often dated 1988; this record follows the 1990 English-language edition cited in the manuscript)
- Elish, M. C. (2019). Moral crumple zones: Cautionary tales in human-robot interaction. *Engaging Science, Technology, and Society*, 5, 40–60. doi: 10.17351/ests2019.260
- Fan, W., Schor, J. B., Kelly, O., & Gu, G. (2025). Work time reduction via a four-day workweek finds improvements in workers' well-being. *Nature Human Behaviour*, 9, 2153–2168. doi: 10.1038/s41562-025-02259-6
- Fisher, M., Goddu, M. K., & Keil, F. C. (2015). Searching for explanations: How the internet inflates estimates of internal knowledge. *Journal of Experimental Psychology: General*, 144(3), 674–687. doi: 10.1037/xge0000070
- Gagné, M., Parker, S. K., Griffin, M. A., Dunlop, P. D., Knight, C., Klonek, F. E., & Parent-Rochelleau, X. (2022). Understanding and shaping the future of work with self-determination theory. *Nature Reviews Psychology*, 1(7), 378–392. doi: 10.1038/s44159-022-00056-w
- Goodhart, C. A. E. (1975). Problems of monetary management: The U.K. experience. In *Papers in monetary economics* (Vol. 1, pp. 1–20). Sydney: Reserve Bank of Australia. (Conference-paper version; a revised collection appeared in 1976)
- Hicks, C. (2026). *The psychology of software teams* (1st ed.). CRC Press.
- Hicks, C. M., Lee, C. S., & Ramsey, M. (2024). Developer thriving: Four sociocognitive factors that create resilient productivity on software teams. *IEEE Software*, 41(4), 68–77. doi: 10.1109/MS.2024.3382957
- Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.
- Kalam, K. A., Masoud, F. D., Muntaser, A., Ranga, R., Geng, X., & Goyal, M. (2025). ChatGPT as a learning tool for medical students: Results from a randomized controlled trial. *Cureus*. doi: 10.7759/cureus.85767
- Kosmyna, N., Hauptmann, E., Yuan, Y. T., Situ, J., Liao, X.-H., Beresnitzky, A. V., ... Maes, P. (2025). *Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task*. (Preprint, version 2) doi: 10.48550/arXiv.2506.08872
- Lee, B. C., & Chung, J. (2024). An empirical investigation of the impact of ChatGPT on creativity. *Nature Human Behaviour*, 8, 1906–1914. doi: 10.1038/s41562-024-01953-1
- Lee, H.-P., Sarkar, A., Tankelevitch, L., Drosos, I., Rintel, S., Banks, R., & Wilson, N. (2025). The impact of generative AI on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 chi conference on human factors in computing systems* (pp. 1–22). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3706598.3713778
- Lindo, M. R., & Cutad, C. A. (2024). Impact of generative AI on self-regulated learning and cognitive offloading. *IMCC Journal of Science*, 4(2), 18–22. doi: 10.65931/c7k2p5z9
- Medeiros, K., Cropley, D. H., Marrone, R. L., & Reiter-Palmon, R. (2025). Human–AI co-creativity: Does ChatGPT make us more creative? *Journal of Creative Behavior*, 59(2), e70022. doi: 10.1002/jocb.70022
- Meyer, A. N., Barton, L. E., Murphy, G. C., Zimmermann, T., & Fritz, T. (2017). The work life of developers: Activities, switches and perceived productivity. *IEEE Transactions on Software Engineering*, 43(12), 1178–1193. doi: 10.1109/TSE.2017.2656886
- Mohamed, A., Assi, M., & Guizani, M. (2026). *The impact of LLM-assistants on software developer productivity: A systematic review and mapping study*. (Preprint, version 3; the latest version reports 39 included studies) doi: 10.48550/arXiv.2507.03156
- Nixon, N., Choi, J., Martins De Bastos, P., Samadi, M. A., Mehner, L., Park, S., & JaQuay, S. (2026). *The social cost of an AI teammate: How an artificial teammate reshapes human–human communication in small-team decision-making*. (Preprint) doi: 10.48550/arXiv.2607.27179
- Pritchard, D. (2026). Educating for intellectual character and the AI revolution in education. *Studies in Philosophy and Education*. (Published online 24 July 2026) doi: 10.1007/s11217-026-10094-z

- Qiao, Y., Shihab, M. I. H., Haque, S., & Hundhausen, C. (2026). *Code comprehension with GitHub Copilot: Performance gains, comprehension trade-offs, and behavioral predictors in brownfield programming*. (Preprint, version 2, 11 May 2026) doi: 10.48550/arXiv.2511.02922
- Rachal, J. R. (1988). Gutenberg, literacy, and the ancient arts of memory. *Adult Education Quarterly*, 38(3), 125–135.
- Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. *Trends in Cognitive Sciences*, 20(9), 676–688. doi: 10.1016/j.tics.2016.07.002
- Rozenblit, L., & Keil, F. C. (2002). The misunderstood limits of folk science: An illusion of explanatory depth. *Cognitive Science*, 26(5), 521–562. doi: 10.1207/s15516709cog2605_1
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78. doi: 10.1037/0003-066X.55.1.68
- Safati, A. B., Seli, P., & Smilek, D. (2026). The costs of monitoring: Does self-catching the wandering mind influence task performance and/or residual probe-caught mind-wandering? *Consciousness and Cognition*, 144, 104112. doi: 10.1016/j.concog.2026.104112
- Salleh, N., Mendes, E., Mendes, F., Lekamlage, C. D., & Petersen, K. (2023). Value-based software engineering: A systematic mapping study. *e-Informatica Software Engineering Journal*, 17(1), 230106. doi: 10.37190/e-Inf230106
- Santarnecchi, E., Galli, G., Polizzotto, N. R., Rossi, A., & Rossi, S. (2014). Efficiency of weak brain connections support general cognitive functioning. *Human Brain Mapping*, 35(9), 4566–4582. doi: 10.1002/hbm.22495
- Sarkar, A. (2024). AI should challenge, not obey. *Communications of the ACM*, 67(10), 18–21. doi: 10.1145/3649404
- Shin, M., Kim, J., van Opheusden, B., & Griffiths, T. L. (2023). Superhuman artificial intelligence can improve human decision-making by increasing novelty. *Proceedings of the National Academy of Sciences*, 120(12), e2214840120. doi: 10.1073/pnas.2214840120
- Şimşek, A. C., Anders, G., Göth, J., Specht, L., & Huff, M. (2025). Is ChatGPT a good study companion? the role of AI-generated summaries and reflective prompts in learning from educational videos. *Computers and Education: Artificial Intelligence*, 9, 100512. doi: 10.1016/j.caeai.2025.100512
- Song, F., Agarwal, A., & Wen, W. (2026). *The impact of generative AI on collaborative open-source software development: Evidence from GitHub Copilot*. (Preprint; record uses the 2026 version cited in the manuscript) doi: 10.48550/arXiv.2410.02091
- Sparrow, B., Liu, J., & Wegner, D. M. (2011). Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333(6043), 776–778. doi: 10.1126/science.1207745
- Storey, M.-A. (2026). *From technical debt to cognitive and intent debt: Rethinking software health in the age of AI*. (Preprint) doi: 10.48550/arXiv.2603.22106
- Strathern, M. (1997). Improving ratings: Audit in the british university system. *European Review*, 5(3), 305–321. doi: 10.1002/(SICI)1234-981X(199707)5:3<305::AID-EURO184>3.0.CO;2-4
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Cambridge, MA: Harvard University Press.
- Wadinambiarachchi, S., Kelly, R. M., Pareek, S., Zhou, Q., & Velloso, E. (2024). The effects of generative AI on design fixation and divergent thinking. In *Proceedings of the 2024 chi conference on human factors in computing systems* (pp. 1–18). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3613904.3642919
- Warm, J. S., Parasuraman, R., & Matthews, G. (2008). Vigilance requires hard mental work and is stressful. *Human Factors*, 50(3), 433–441. doi: 10.1518/001872008X312152
- Yang, L., Holtz, D., Jaffe, S., Suri, S., Sinha, S., Weston, J., ... Teevan, J. (2022). The effects of remote work on collaboration among information workers. *Nature Human Behaviour*, 6(1), 43–54. (Published online in 2021; assigned to the 2022 journal issue) doi: 10.1038/s41562-021

-01196-4

- Yu, S., Cheng, M., Jabbar, A., Sucholutsky, I., Collins, K. M., Jurafsky, D., & Hawkins, R. D. (2026). Cognitive offloading and the speedup illusion in human–AI interaction. In *Proceedings of the 48th annual meeting of the cognitive science society*. doi: 10.48550/arXiv.2605.23177
- Zakharov, I., Tabueva, A., Adamovich, T., Kovas, Y., & Malykh, S. (2020). Alpha band resting-state EEG connectivity is associated with non-verbal intelligence. *Frontiers in Human Neuroscience*, 14, 10. doi: 10.3389/fnhum.2020.00010
- Zhu, K., & Walker, D. (2026). The AI democratization paradox: Evidence from decentralized knowledge communities. *Management Science*. (Published online) doi: 10.1287/mnsc.2024.04717