Keeping Humans in the Driver's Seat: A Conceptual Framework for GenAI and Competence Debt

Clara Marie Lüders
clara.marie.lueders@optano.com
OPTANO GmbH
Paderborn, Germany

Olaf Krużycki
olaf.kruzycki@outlook.de
techagogics GmbH
Kiel, Germany

Carolin Brandt
c.e.brandt@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Abstract

With generative AI (GenAI) being taken up in many sectors of the economy, especially in software-related areas, there is a growing urgency to understand how GenAI impacts our work. Because GenAI enables us to produce seemingly great outputs with little skill, the impact on learning and skill-building is especially interesting. In this paper, we present a conceptual framework to describe the concept of competence debt—illustrating how extensive use of GenAI can inhibit learning and impact the quality of our outputs. We apply the conceptual framework to describe various already visible impacts of GenAI on organizations and use it to infer guidelines on how organizations can mitigate the negative effects of GenAI usage.

CCS Concepts

• **Software and its engineering** → **Software creation and management**; **Software creation and management**; • **Social and professional topics** → **Automation**; *Socio-technical systems*; *Employment issues.*

Keywords

Generative AI, Human-AI Interaction, Guidelines, Future of Work

ACM Reference Format:
Clara Marie Lüders, Olaf Krużycki, and Carolin Brandt. 2026. Keeping Humans in the Driver's Seat: A Conceptual Framework for GenAI and Competence Debt. In *Companion Proceedings of the 34th ACM Symposium on the Foundations of Software Engineering (FSE '26), June 5–9, 2026, Montreal, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The evolution of Generative Artificial Intelligence (GenAI) has brought many achievements. It leads to systems that can plan, reason, call tools, and execute actions with limited supervision [27].

GenAI and its advantages are sometimes compared to the invention of machine code or calculators. However, it differs crucially in two dimensions; first, it is stochastic, and second, we currently have no reliable way to verify the content produced by GenAI [20]. In part, this is due to the broadness of tasks that can be solved by GenAI, while evaluation approaches (such as reviews, audits, tests) are typically task-specific and can be time-intensive to create or execute. Depending on the task given to a GenAI system, the output can often only be evaluated by specific, often human, experts [20]. It is unlikely that we soon figure out a general approach for large-scale quality assurance for all tasks doable by GenAI. Currently, a common approach is to use GenAI-based approaches to evaluate GenAI output [20]. Since stochastic nature is inherent to GenAI, we cannot trust that GenAI can evaluate the output of GenAI-systems as accurately as a human expert can.

Several large-scale failures of unchecked GenAI results are already occurring. For example, *DER SPIEGEL*, a German news magazine, released an article, which ended in "If you would like, I'll adapt tone and detail depth (for example, more formal news vs more magazine news) or can mark you concrete changes in comparison to the original". This commonly used Call-to-Action from GenAI tools made it obvious that the journalists did not read the article before publishing it[1]. This is one example of preventable reputational damage that was caused by a lack of quality control. The consulting agency Deloitte used GenAI in a 440.000 Australian Dollar report for the Australian government, which contained several fabricated references and experts. Even though Deloitte corrected its report after the hallucinations were found by a researcher, this still caused reputational damage[2]. Lastly, there is a rising number of cases in which lawyers are using GenAI in an uncontrolled manner. Several lawyers have been proven to have used GenAI to convince courts in their favour, with GenAI, in some cases, hallucinating judgments on which said lawyers relied. This shows that a lack of quality control can not only damage reputations, but can also lead to potentially incorrect court judgments[3]. These three examples show that a lack of quality control led to reputational and societal damage caused by uncaught hallucinations and errors in GenAI. Placing unconditional trust in GenAI output is not only counterproductive but can also be harmful. This illustrates the necessity of comprehensive quality control of content produced by GenAI.

GenAI's speed and capacity to produce output far surpasses the human speed to produce or check output. This leads us to a conundrum. There is no universal and reliable way to check GenAI results [3], and our current quality control processes can hardly keep up with the amount of results produced by GenAI.

At the same time, by increasing the use of GenAI tools in their everyday work, humans have fewer incentives to practice core

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://www.20min.ch/story/ki-text-im-spiegel-wie-kam-es-zu-dieser-panne-103439186
[2] https://www.business-standard.com/technology/tech-news/deloitte-ai-hallucination-report-australia-gpt4o-fabricated-references-125100800915_1.html
[3] https://www.bbc.com/news/world-us-canada-65735769

skills because they increasingly offload the cognitive work to the tool [18]. This can lead to a dangerous spiral of knowledge loss and thus competence erosion. In an extreme scenario, this could lead to a point of no return for organizations, as they might ultimately lose the sufficient levels of competence required for quality control.

Motivated by the conundrum described above, the work from Kosmyna et al. [21] and Brynjolfsson [9], this paper aims to conceptualise the different factors and their interaction stemming from unguided GenAI usage into a framework. We want to draw attention to competence debt[4], defined as "The gap between what is in your codebase and how much of it you understand.". We want to extend this definition beyond software engineering. We define competence debt as: *"The gap between what is created in an organisation (assets, decisions, obligations, practices, codebases) and how well the responsible people can reliably explain, predict, and change it."* Our conceptual framework aims to explain how competence debt can arise from short-sighted, unguided use of GenAI tools. In the long term, unguided GenAI use can lead to adverse consequences due to a lack of skills within the company. One of these potential consequences are dangerous black boxes in workflows, components whose effect is not fully understood and whose behaviour cannot be traced back at later point. We argue that unguided GenAI use acts as an accelerator for competence debt, as skills can now be compensated for cost-effectively by GenAI without considering possible long-term consequences.

Our proposed conceptual framework also helps identify interventions to mitigate the competence debt and its adverse consequences through reflected and moderated use of GenAI. As the immediate impacts of GenAI are observable in the Software Engineering field, we focus the paper on the software engineering field and use examples from this field.

It is not our aim to demonise GenAI, as its tools can accelerate learning if used correctly. For example, improved self-teaching of juniors and as troubleshooting and problem-solving assistants. We want to promote a sensitive and informed approach to GenAI, as it is ultimately a double-edged sword in terms of the education and training of future skilled workers and should therefore not be used without careful consideration.

We think that such a framework is necessary because only through understanding can we derive effective measures to protect against unwanted consequences. Without the factors and their interaction that lead to competence debt, how could we build future systems that are safer against it?

The remainder of this paper is organised into seven sections. After the brief literature review in Section 2, Section 3 shortly outlines the used methodology. Section 4 presents and explains the conceptual framework. In Section 5, we examine the various negative effects which we can explain with the framework, while Section 6 offers remedies and recommendations to mitigate the adverse outcomes. Section 7 presents an outline for a research agenda to extend and evaluate the conceptual framework and discusses the limitations of the conceptual framework. Finally, Section 8 concludes the paper, summarising the main insights.

[4]We found a first definition here https://www.leanway.no/competence-debt/.

2 Related Work

The popularity and rise of GenAI evolved from completing simple tasks to autonomous workflows. This section reviews previous research related to productivity with GenAI, as well as learning, GenAI adoption strategies, and other forms of debt in SE.

2.1 Measurement and Evaluation of GenAI Success

Productivity gains due to the usage of GenAI are being extensively researched. Brynjolfsson et al. [9] reported employment growth in jobs where GenAI is augmentative, while employment is decreasing where GenAI is used to automate. Peng et al. [29] found that software developers who use GitHub CoPilot complete coding tasks 55.8% faster with varying benefits depending on the software developers' experience. Noy et al. [27] also reported that ChatGPT users were more productive and efficient in writing tasks.

However, these productivity gains come with downsides. He et al. [19] studied the adoption of Cursor on GitHub and found that while productivity grows, so does code complexity and warnings, indicating technical debt, which reduces productivity gains long-term. Perry et al. [30] found that developers using GenAI coding assistants wrote less secure code while being confident that their code is secure. Macnamara et al. [23] demonstrated that even experts risk losing their skill by relying on GenAI assistants.

The metrics used to evaluate the performance of GenAI usage are primarily technical. Meimandi et al. [25] analysed 84 papers on agentic GenAI evaluation and found that 83% emphasise technical performance metrics while only 30% include human-centred evaluation dimensions. He et al. [20] reported that human evaluation is effective, but very costly and slow.

2.2 Skill Formation and Skill Decay

Several research studies investigated how working with GenAI influences the development of skills. Grinschgl et al. [18] showed that cognitive offloading to external systems boosts immediate performance but diminishes retention. In correlation, Passalacqua et al. [28] found that using fully automated decisions reduced perceived autonomy, self-determined motivation, behavioural task engagement, and skill acquisition during training compared to those who used it as a decision aid. Similar findings were replicated by Lee et al. [22] who described a correlation of higher confidence in GenAI tools with reduced critical thinking. This corresponds to the findings of Ericsson et al. [14] who found that engagement is needed to foster competence. Moreover, Prather et al. [31] found in that students who did not have any difficulties with a course they were taking and the subject matter were able to use GenAI sufficiently to accelerate their learning progress, while students with difficulties further exacerbated their existing problems. This was also found by Rahe and Maalej [32]. They observed two usage strategies in students, one group used ChatGPT to seek knowledge and the other tried to unload their task to ChatGPT, prompting it to solve the task directly. Dickey et al. [13] coin a fitting term to describe this: the "junior-year wall": students who increasingly rely on GenAI are more likely to encounter problems in advanced courses, as GenAI can no longer keep up with the knowledge required, thus widening the knowledge gaps between GenAI and

students. Shen and Tamkin [36] tasked developers to learn a new library with and without the use of GenAI. They found that the use of GenAI impaired valuable skills without delivering efficiency. Productivity was won at the cost of learning. Kosmyna et al. [21] similarly examined participants tasked with a writing task. They partitioned the participants in three groups: GenAI, Search Engine and No-Tools. They found that sustained reliance on GenAI can lead to under-engagement and poorer neural, linguistic, and behavioural performance, even when participants later write without a tool. Bastani et al. [7, 8] similarly examined math students who had access to a GenAI tutor. They showed that without carefully designed guardrails, students underperform after removing the GenAI tutor.

2.3 GenAI Adoption Strategies

There has also been research into different strategies on how humans adopt GenAI. Gama and Magistretti [16] distilled three practical GenAI application strategies into a taxonomy:

Replace GenAI is adopted as a tool to improve existing processes, substitute human beings, and expedite external analyses. The motivation is often economic.

Reinforce GenAI is adopted as a lever to exploit new technological opportunities, empower existing processes, assist employees in their activities, and expedite external analyses.

Reveal GenAI is adopted as a sonar to unveil hidden technological opportunities and unshadow unforeseeable external situations.

This is similar to the MIT study from Randazzo et al. [33], which classified the engagement of consultants from the Boston Consulting Group (BCG) into three categories:

Cyborgs (Fused Co-Creation) Human and GenAI shape each other in a tightly fused decision process. These humans acquire new capabilities through GenAI.

Centaurs (Directed Co-Creation) The human steers the process while leveraging GenAI capabilities. The human maintains strategic control.

Self-Automators (Abdicated Co-Creation) Delegation of both task and decision to GenAI with minimal human judgment. These employees fail to gain new capabilities.

The Self-Automators are similar to the Replace strategy, while Centaurs and Cyborgs are different interpretations of the Reinforce strategy. Replace/Self-Automators are both unguided GenAI use with the negative consequence of "not learning". The Reinforce, Cyborg, and Centaur or Reveal strategies are not exposed to the same risk.

2.4 Forms of Debt

There are already various forms of debt researched, particularly in the field of Software Engineering, such as Technical Debt, Social Debt and Cognitive Debt.

Technical Debt is defined by Avgeriou et al. [5] as *"a collection of design or implementation constructs that are expedient in the short term but set up a technical context that can make future changes more costly or impossible. Technical Debt presents an actual or contingent liability whose impact is limited to internal system qualities, primarily maintainability and evolvability."* Technical Debt seems to be increasing with unguided use of GenAI [4, 41].

On the other side of the spectrum, Social Debt is defined by Tamburri et al. [38, 39] as *"a cumulative and increasing cost in the current state of things, connected to invisible and negative effects within a development community. These effects might need some digging in order to be found since they are connected to undesirable, often implicit characteristics in the organisational and social structure emerging in development communities. These characteristics produce an additional cost, e.g., increase the time needed for development."* Social debt and its sources are often harder to measure and detect, leading to difficulties in uncovering and resolving it.

The closest related debt is likely Cognitive Debt. As defined by [21], Cognitive Debt is *"a condition in which repeated reliance on external systems like LLMs replaces the effortful cognitive processes required for independent thinking. Cognitive debt defers mental effort in the short term but results in long-term costs, such as diminished critical inquiry, increased vulnerability to manipulation, decreased creativity. When participants reproduce suggestions without evaluating their accuracy or relevance, they not only forfeit ownership of the ideas but also risk internalising shallow or biased perspectives."* Competence Debt is closely linked to Cognitive Debt. Cognitive Debt describes the personal reliance on external tools, while Competence Debt focuses on organisational knowledge and competence gaps. They describe different scopes and therefore have different consequences. However, widespread Cognitive Debt in an organisation is a major factor for Competence Debt in the same organisation.

3 Methodology

We develop a conceptual framework as defined by Roel Wieringa [40] to describe constructs in working with GenAI within organizations. Our goal is to provide a basis for communicating about how different aspects influence each other; ultimately leading to competence debt. To create the framework we followed an iterative approach. We grouped and collected recent research and organised them into a meta view. To arrive at the conceptual framework, each of the authors familiarised themselves with the current research around GenAI and its impact. The first author then provided a first idea for a conceptual framework capturing the bigger picture and factors. This was then discussed and refined with all authors. Afterwards, the first author improved the conceptual framework to capture the discussed nuances. We repeated this process until we arrived at a framework that was coherent enough and able to explain the phenomena presented in the research. The framework was then reviewed by other researchers and improved with their feedback.

4 Conceptual Framework: Competence Debt through GenAI

In this section, we want to put all correlations and causal links that research has uncovered into a higher-level overview. GenAI can generate a lot of results extremely fast. However, the true quality of the results varies while the results often look like good quality at first glance. These results that look like good quality are referred to as workslop [5]. Quality control is now burdened twofold. On the

[5] https://hbr.org/2025/09/ai-generated-workslop-is-destroying-productivity

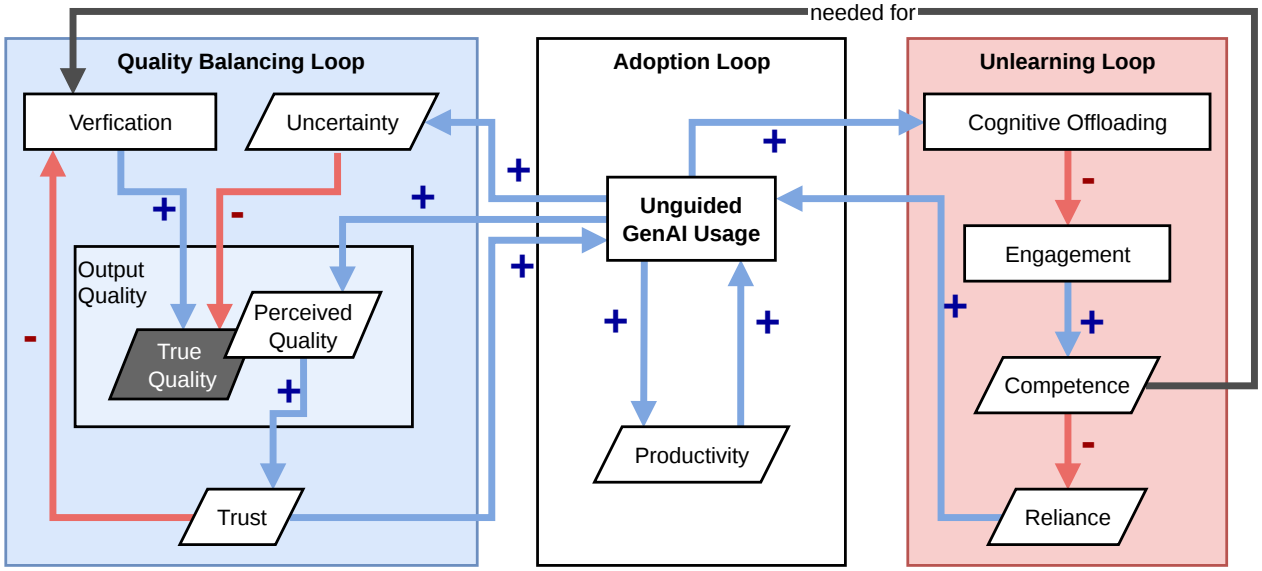

Figure 1: Conceptual Framework of Competence Debt with GenAI

one hand, many more results are produced. On the other hand, it gets increasingly harder to differentiate between good and bad quality quickly. Thus, the true quality of GenAI results is often only revealed later. For example, He et al. [19] already showed that the initial velocity of productivity of using Cursor also leads analysis warnings and higher code complexity, which leads to slowing down velocity long term.

Figure 1 illustrates our conceptual framework [40] of competence debt that may arise through unguided use of GenAI. The unguided GenAI usage triggers three loops: The adoption loop, the unlearning loop and the quality balancing loop. Each loop is made up of concepts that we identified through related literature as relevant to represent within the framework. In the figure, we use blue for strengthening and red for weakening influences for visual clarity. Additionally, we differentiate between human activities, for which we use rectangles and characteristics, for which we use parallelograms.

The Adoption Loop. Starting at the centre of the framework, **Unguided GenAI Usage** refers to how frequently employees delegate tasks to GenAI. Unguided refers to the use of GenAI without external guidelines or reflecting about how and when GenAI should be used. The adoption loop characterises how **Unguided GenAI Usage** leads to more **Productivity** [10, 27, 29]. This **Productivity** increase shows that the tools are useful, which then leads to more **unguided GenAI Usage** [2]. This is a self-reinforcing loop. The strength of this loop does depend on how well the GenAI fits the task; the better the fit, the stronger the loop.

The Unlearning Loop. This loop characterises how unguided GenAI usage leads to unlearning skills. **Unguided GenAI Usage** leads to an increase in **Cognitive Offloading** [21, 22, 36]. Users offload information, decision-making, analysing to a GenAI tool, often because it is easier. The increased **Cognitive Offloading** can decrease active **Engagement** [18, 32]. Because users offloaded the cognitive burden to the tool, they spent less energy and time actively engaging with the material. However, active **Engagement** is needed to increase **Competence** since learning requires active engagement [11, 14, 23]. The more **Competence** users have in a specific skill for a task, the less users are **Reliant** on systems that perform the same task [7, 12]. For example, using GPS leads to a decrease of spatial transformation abilities, which are skills needed for navigation [35]. High **Reliance** leads to some dependency on such systems, which then likely leads to the system being used more, i.e. more **GenAI Usage** [32]. For example, students that solved a task with GenAI are not able to fix errors in their solution without using GenAI [13, 32]. We arrived back at our starting point and this then triggers the loop again.

The Quality Balancing Loop. This loop characterises how the quality of results balances the tool usage delayed. Output quality consists of the **True Quality** of the result and the **Perceived Quality**. Since the **Perceived Quality** is often very good in **GenAI** results because GenAI can mask their result quality [44], the user **Trusts** GenAI more [43]. More **Trust** in the output of a GenAI also leads to more **GenAI Usage**. This loop triggers first and we conjecture that this loop is short cutting the actual quality balancing loop. Due to the stochastic nature of GenAI, higher **Unguided GenAI Usage** leads to higher **Uncertainty** [26]. We conjecture that the **Uncertainty** inherent to GenAI may reduce the **True Quality** of results. The more a user **Trusts** the output of a system, the less they **Verify** this output's true quality [34]. The more (intense) humans **Verify** results, the more likely the **True Quality** of results is higher [24]. Unfortunately, only a human with high **Competence** can do fitting **Verification**. This loop can balance the unguided GenAI usage, but due to verification being a time-intensive process

for more complex tasks, the high speed of the content production of GenAI and the high perceived quality, the true quality of the GenAI output is only revealed later. This leads to the latency.

As trust in GenAI grows and human competence decreases, thorough verification could also decrease. This means errors or "workslop" go uncorrected (and unnoticed), further concealing the accumulating competence debt until a failure occurs.

Competence Debt. The increased speed of the adoption loop accelerates the unlearning loop, which leads to an erosion of skill while the quality balancing loop is short cutted due to the masking of the true quality. This means that competence debt increases—the gap between what is created in an organisation (assets, decisions, obligations, practices, codebases) and how well the responsible people can reliably explain, predict, and change it. Through unguided GenAI usage competence debt could be also noticeable when outputs are generated and implemented, whose effects cannot be traced at a later point in time. This accumulates into lower traceability and possible dismantling resulting in additional work and economic losses. Previously, the unlearning loop was balanced by the quality balancing loop. However, since GenAI is producing results (of seemingly good quality) at a way faster rate than humans can evaluate, the quality balancing loop now has a latency. Negative consequences are not immediately visible, but speed and convenience are. This means that competence debt is collected at a faster rate with delayed consequences. The problem with competence debt is that its long-term effects only become apparent after some time, which means that the effects of competence debt are hard to adequately anticipate.

5 Competence Debt through Unguided GenAI Usage: The Effects in Organisations

In this section, we want to discuss the (negative) effects unguided GenAI has on organisations and the different dimensions of competence debt effects.

Our proposed conceptual framework helps explain why companies face emerging risks, i.e. reliance on uncertain systems, reduced active engagement and thus less critical thinking and de-skilling. Following our framework, if unguided GenAI usage in an organisation increases, both the adoption loop and the unlearning loop are activated. Both increase the usage and reliance on GenAI and the erosion of competence. At the same time, the increased unguided GenAI usage leads to a decrease in quality through the quality balancing loop, which is not noticed immediately. The temporal asymmetry of the quality balancing to the unlearning and adoption loop leads to increased competence debt.

GenAI's democratisation of skills can be a double-edged sword. GenAI seems to democratise various skills which were previously limited to specific human experts, such as programming [29]. Non-experts use GenAI-based tools unguided, and productivity is increasing (the adoption loop). This can lead to non-experts creating outputs which they cannot verify, as they simply lack the competence to do so (Figure 1, Competence is needed for Verification). For example, in software engineering, the non-experts don't know if the produced code has security issues [30], bugs [30], is maintainable or is of good quality [19]. These non-experts can neither discern the content quality of the results nor verify the quality completely. At the same time, through the unguided GenAI usage, they are also not improving their competence (the unlearning loop).

GenAI can be used to support learning, but is not automatically helpful. However, there is potential that, with sufficient intrinsic motivation, GenAI can be used to acquire skills through self-teaching faster and more easily. However, it seems that with unguided GenAI usage, this is often not the case [8, 21, 36]. For example, in software engineering, most non-experts can get caught in a "copy-paste-please-fix" loop [32]; Students continuously supply the GenAI-based system with an error message without checking the results themselves. The unlearning loop is often time-consuming and ineffective, and seems to not easily break without sufficient guardrails.

GenAI enables skill masking. Additionally, humans often infer skills via different signs; for example, a more eloquent person will be seen as more knowledgeable, since these properties, i.e. knowledge and eloquence, often correlate. We also infer the competence of a person from the quality of the result, i.e. the more competent a person, the better the quality of their produced results. However, with GenAI, it takes significantly less work to produce results that look polished. These polished-looking results mask the skill gap [42]. If skill gaps are masked, then other persons, such as mentors or high-skill employees, might not be aware of skill gaps and encourage low-skilled employees less. Therefore, it can slow down learning, and the effects of the unlearning loop are not noticed.

GenAI can trick non-experts. GenAI is good at masking its quality issues and hallucinations [37, 44]. Every expert in a specific topic who asks a GenAI-based system, such as ChatGPT, for information on their topic is aware that these systems often contain quality issues, and these experts often spot these quality issues. However, at the same time, the same expert does not spot these quality issues in the result of a GenAI-based system on a topic they are not knowledgeable about.

Aversion to algorithms among older high-skill employees might lead to too little scrutiny within companies towards potentially GenAI-produced outputs. Another factor contributing to the widening of the competence gap due to GenAI is an inherent aversion to algorithms among older people, who rather tend to hold senior positions [6]. This puts companies at risk of a lack of awareness of GenAI due to personal aversions among those in higher positions, making it even easier to mask skills. Due to the aversion, older high-skill employees might have a limited understanding of GenAI; they themselves have problems recognising results produced by GenAI as such, which means that juniors can once again mask their skills.

Quality checks relying mainly on high-skill employees means high GenAI usage burdens high-skill employees with more work. Experts, i.e. people with high competence, are needed to check the quality (Figure 1, Competence is needed for Verification). However, through the increased productivity (adoption loop), these experts will struggle to keep up with the verification, which leads to the latency of the quality balancing loop. Additionally, we have an asymmetry in the number of experts who can check the quality of results versus the number of non-experts producing results. Due to GenAI being faster [44], lower-skill employees can produce more work with a risk of producing workslop (Figure 1,

Unguided GenAI Use increases Uncertainty decreases True Quality). This, in turn, either produces more work for high-skill employees because more verification is needed [41]. This then enforces the negative effects of the asymmetry or risks economic loss. Furthermore, the effects of this asymmetry compound over time, with unguided GenAI usage eroding the acquisition of skill in users (unlearning loop) while experts are leaving the workforce.

Stopping junior hiring or replacing juniors because of GenAI will lead to a bottleneck. Companies that primarily focus on trying to replace human employees with GenAI, going as far as offboarding, will struggle more with the consequences described above, for example, Duolingo, Microsoft and Salesforce[6]. The replacement/self-automator strategy [16, 33] seems to focus primarily on customer support, coding, and data processing departments and led to decreased costs [9]. However, the strategy has already shown its negative effects in the form of backlash and regrets[7]. Organisations that maximise GenAI substitution of employees will experience short-term gain by decreasing costs and increasing productivity (the adoption loop). Long term, if we follow our framework, these companies will see a decline in competence (unlearning loop), which will lead to a quality collapse (quality balancing loop with no one to fix it). Since the balancing loop of quality triggers later, the debt of the short-term gain also only surfaces later. These companies lose important growth in implicit company knowledge because they essentially removed the junior-senior mentorship. When high-skill employees retire or switch companies, their implicit and explicit knowledge is lost [9].

6 How Can Organizations Prevent Competence Debt While Using GenAI?

In this section, we want to suggest methods that can be used to avoid competence debt and other unwanted consequences.

Measure GenAI success via metrics that value quality and competence. As shown by Meimandi et al. [25], GenAI success is often not measured on human-centered measures. Additionally, we are also missing measurements of GenAI success beyond the increased productivity. Productivity is an imperfect proxy metric for success. Therefore, relying on productivity as a measure for success too heavily, organisations can unintentionally reward quantity over quality. This would also lead to an increase of unguided GenAI usage per our conceptual framework as adoption loop is boosted by higher productivity. If we instead elevate competence-measuring metrics and quality metrics, we introduce moderating factors in both the quality balancing and the unlearning loop. However, as a prerequisite to creating other metrics, we also need a good understanding of our current workflows, data, and quality properties of results. A starting point for fitting metrics could be the characteristics in the parallelograms in Figure 1. For example, we could measure uncertainty by running a GenAI system a set number of times and use emedd Essentially, we need to start measuring success such that quality and learning get more emphasis, not only output or productivity. For example, automated quality evaluation is very hard to achieve. For example, current code quality checks rely other software engineers to review the code in a merge request. The concrete review guideline and definition of done is often decided and improved over time inside the software development team.

Reinforcement [16] / Cyborg / Centaur Strategy [33]. Organisations that use guided GenAI to augment their workflows, such as AWS[8], are not impacted by competence debt as much. Instead of (mis)using GenAI to replace the perceived costs of juniors, organisations use GenAI to lower their investment costs in junior employees and speed up their senior employees. These organisations shift the easily checkable work, such as boilerplate code, routine customer questions, to the GenAI and freeing mental capacities of their workers to handle higher-cognitive tasks, like rare customer questions and difficult software architecture design. Since only easily checkable work is moved to GenAI, the employees can assess the quality of results more quickly, leading to faster balancing. Long-term, this strategy faces little competence debt and is sustainable.

Introduce guardrails and training to ensure mindful use of GenAI. We further need guardrails, for example, in the form of policies and training for employees and organisations that take into account different context factors, such as the experience of the user in the task domain, which they wish to solve with the help of a GenAI system. If companies create GenAI-supported workflows for tasks and educate people on the correct use of GenAI, i.e., mindfully validating GenAI outputs and being critical of outputs, their skill level could at least hold [8].

How to best use GenAI depends on multiple context factors: *The experience of the user.* A junior employee is not yet experienced enough to check GenAI output sufficiently; a senior employee can check it by themselves with relatively low risk. Therefore, low-skilled and high-skilled employees should have different access to a GenAI system. For example, in software engineering, seniors are allowed to "vibe code" while juniors can only review their code, ask questions or produce documentation. *The fidelity of the result.* A prototype has lower fidelity and thus needs lower quality, they are a good sandbox to learn, whereas a product has high fidelity and less room for errors. The longer a solution is maintained, the fewer GenAI systems should be used. *The criticality of the system.* The more critical a system, the higher the need for correctness and quality. Therefore, the more critical the product, the fewer GenAI systems should be used. *The available resources.* The more time-critical a task is, the more likely humans will use GenAI. Additionally, the more time-critical, the more likely it is that quality assurance is skipped. *The company's awareness of GenAI, its modes of operation, functions, and outputs.* Low awareness favours individual employees within the company and tempts them to sell GenAI-generated output as their own. If awareness is high, the potential to use GenAI as a sparring partner increases, enabling even better results to be achieved based on the output.

Concretely, we recommend as guidelines:

- Organisations should implement **metrics that capture the result quality and skill retention** of their employees.

[6] https://tech.co/news/companies-replace-workers-with-ai
[7] https://www.vice.com/en/article/this-company-replaced-workers-with-ai-now-theyre-looking-for-humans-again/
[8] https://talent500.com/blog/aws-ceo-matt-garman-says-replacing-junior-developers-with-ai-is-a-mistake-he-explains-why-junior-talent-is-vital-and-how-ai-should-empower-not-replace/

- Organisations should **communicate guidelines and a framework for action** on the use of GenAI within the company. For example, senior employees return work and point out improvements, so that the effort of fixing outputs lies on the person generating it. This would combat workslop and build up a healthy mindset of quality responsibility.
- Organisations should consider implementing **mandatory GenAI training** for all employees. This should be carried out by external personnel who are neutral towards the company to convey the benefits and risks without bias. The training should also include a fundamental understanding of the architecture in order to anchor the functioning of GenAI in the company and beyond in the long term. Furthermore, training could be mandatory to unlock GenAI tools.
- Constrain when GenAI can be used depending on context factors. For example, **allow GenAI generation for prototyping**, but not for production code, or constrain GenAI use such that **junior developers may only use GenAI as a tutor/reviewer** (explain, debug, improve) and reserve generation privileges for senior developers who can discern quality.

7 Future Directions and Limitations

This paper proposes a conceptual framework to support reasoning about the effects and interactions of unguided GenAI usage by combining relationships reported in mostly recent scientific literature, such as the short-term productivity increase reported by He et al [20] and the skill erosion evidence of Shen and Tamkin [36]. The framework is intended as an organising lens rather than a causal model with guaranteed completeness, identifiability, or predictive power.

There are several limitations to this framework. First, GenAI research and capabilities evolve rapidly, so additional loops, characteristics, and constructs will likely emerge and should be incorporated over time. Second, we cannot claim causal completeness; the strength and direction of interactions can be context-dependent, e.g. task type, domain type, and user expertise. Third, the model is currently built up from scientific reports of the interaction of individual components, but the second order interactions across components are only hypotheses. To validate and refine the framework as a whole, we need comprehensive real-world data on unguided GenAI usage, which is currently scarce. Furthermore, most empirical data we base our conceptual framework on is only from short-term studies. Together with the delayed aspect of the quality balancing loop, it is necessary to validate the long-term cumulative effects of competence debt through longitudinal studies.

Recognising these limitations, we envision our conceptual framework as a living tool. We conjecture that the conceptual framework can help researchers and practitioners describe consequences of GenAI usage and identify measures against negative consequences from a holistic perspective. To iteratively refine and empirically ground the model, we propose multiple research directions to iteratively refine and empirically ground our conceptual framework.

In a first step, we need to identify appropriate metrics to measure the individual components of our conceptual framework. Existing metric frameworks, like the DX AI measurement framework by Noda and Tacho [1], or the SPACE framework for developer productivity by Forsgren et al. [15], could serve as inspiration. One concrete example is to investigate the revisions necessary to contributions created with(out) help of GenAI. This would estimate the effect of workslop by measuring and monitoring the amount of comments, suggestions and time until approval in merge requests, and how does this differ between code that was created with no GenAI, unguided GenAI and guided GenAI usage. Such quantitative investigations enable us to measure the individual loops and their strengthening effects over a longer time period in an organization. This would also be the basis to determine the effect sizes of the different strengthening/weakening connections in the framework.

As related work on measuring developer experience and productivity emphasizes [1, 15, 17], it is also crucial to investigate how individual context of a software project and the involved individuals alters the effects and interaction in our framework. An orthogonal approach to gain insights into context would be to combine the analysis with qualitative interviews of the respective software developers, to gain some insights such as their stance to GenAI and their skill level, and how these affect the loops within our conceptual framework.

To show the value of our inferred recommendations, such as comprehensive guidelines as guardrails, comparative studies of guided and unguided GenAI usage are needed. For example, by studying how in the long term (unguided) GenAI usage effects average time-to-defect or time-to-defect-fixing. To measure the impact on learning and quantify the unlearning loop, we propose to use controlled experiments in a software development task, similar to Kosmyna et al. [21]. Three groups of developers— one is not allowed any additional tools besides an IDE, the second is allowed unguided use to a GenAI system and the third is allowed GenAI use with concrete guardrails. Beyond measuring productivity and code quality, it is crucial to measure in how far skill was developed and/or retained between the groups.

To further study the latency of the quality-balancing loop, beyond He et al.'s results [20], we propose to measure and monitor the presence of code smells in organizations working with GenAI and correlate them with development velocity to identify if there are negative effects.

Together, these research directions would strengthen the proposed conceptual framework into a data-driven model that is capable of providing custom and contextualized insights in organizations to inform policy, tooling, and training strategies.

8 Conclusion

GenAI is transforming the way we work, and we must understand how various factors interact and the potential consequences. One of these consequences is competence debt, i.e. short-term productivity gains at the cost of a widening gap between what an organisation produces and the expertise required to sustain and maintain that production. Drawing on recent findings in literature, we proposed a conceptual framework that captures the dynamics of unguided GenAI use in three loops: the adoption loop, the unlearning loop and the quality-balancing loop. These loops jointly explain how unguided GenAI use can create a self-reinforcing spiral that compromises quality and skill acquisition. We use the conceptual model

to discuss illustrative examples of effects on organizations and infer recommendations on how to mitigate the negative effects of unguided GenAI use leading to competence debt. While we focus on software engineering, we conjecture that insufficiently moderated use of GenAI in similar way impacts creation in society as a whole.

To leverage the benefits of GenAI in the future-of-work, we need to critically reflect on how it impacts learning and skill acquisition processes and how to moderate its use to ensure organisations still grow human expertise. We further need to evaluate how we measure to incorporate result quality and skill retention metrics into performance assessments for GenAI-based systems in organisations. From a future-of-work perspective, the challenge is not merely whether GenAI improves productivity, but how to structure human oversight, autonomy, and learning. By recognising the different dynamics, organisations (and society) can ensure humans remain in the driver's seat and not become passengers.

References

[1] Noda A. and Tacho L. 2025. Measuring AI code assistants and agents: The DX AI measurement framework. https://getdx.com/research/measuring-ai-code-assistants-and-agents/ accessed:2026-02-12.

[2] Dennis A Adams, R Ryan Nelson, and Peter A Todd. 1992. Perceived usefulness, ease of use, and usage of information technology: A replication. *MIS quarterly* (1992), 227–247.

[3] Aldeida Aleti. 2023. Software testing of generative ai systems: Challenges and opportunities. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE, 4–14.

[4] Ahmed Aljohani and Hyunsook Do. 2025. PromptDebt: A Comprehensive Study of Technical Debt Across LLM Projects. In *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering*. 371–382.

[5] Paris Avgeriou, Philippe Kruchten, Ipek Ozkaya, and Carolyn Seaman. 2016. Managing technical debt in software engineering (dagstuhl seminar 16162). *Dagstuhl reports* 6, 4 (2016), 110–138.

[6] Maria Barbul and Irina Bojescu. 2023. Generations' perception towards the interaction with AI. *R. Pamfilie, V. Dinu, C. Vasiliu, D. Pleșea, L. Tăchiciu eds* (2023), 539–546.

[7] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2024. Generative AI can harm learning. *The Wharton School Research Paper* (2024).

[8] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2025. Generative AI without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences* 122, 26 (2025), e2422633122.

[9] Erik Brynjolfsson, Bharat Chandar, and Ruyu Chen. 2025. Canaries in the coal mine? six facts about the recent employment effects of artificial intelligence. *Stanford Digital Economy Lab. Published August* (2025).

[10] Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. 2025. Generative AI at work. *The Quarterly Journal of Economics* 140, 2 (2025), 889–942.

[11] Michelene TH Chi and Ruth Wylie. 2014. The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist* 49, 4 (2014), 219–243.

[12] Chun-Wei Chiang and Ming Yin. 2021. You'd better stop! Understanding human reliance on machine learning models under covariate shift. In *Proceedings of the 13th ACM Web Science Conference 2021*. 120–129.

[13] Ethan Dickey, Andres Bejarano, and Chirayu Garg. 2023. Innovating computer programming pedagogy: The AI-lab framework for generative AI adoption. *arXiv preprint arXiv:2308.12258* (2023).

[14] K Anders Ericsson, Ralf T Krampe, and Clemens Tesch-Römer. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological review* 100, 3 (1993), 363.

[15] Nicole Forsgren, Margaret-Anne D. Storey, Chandra Shekhar Maddila, Thomas Zimmermann, Brian Houck, and Jenna L. Butler. 2021. The SPACE of developer productivity. *Commun. ACM* 64, 6 (2021), 46–53. doi:10.1145/3453928

[16] Fábio Gama and Stefano Magistretti. 2025. Artificial intelligence in innovation management: A review of innovation capabilities and a taxonomy of AI applications. *Journal of Product Innovation Management* 42, 1 (2025), 76–111.

[17] Michaela Greiler, Margaret-Anne D. Storey, and Abi Noda. 2023. An Actionable Framework for Understanding and Improving Developer Experience. *IEEE Trans. Software Eng.* 49, 4 (2023), 1411–1425. doi:10.1109/TSE.2022.3175660

[18] Sandra Grinschgl, Frank Papenmeier, and Hauke S Meyerhoff. 2021. Consequences of cognitive offloading: Boosting performance but diminishing memory. *Quarterly Journal of Experimental Psychology* 74, 9 (2021), 1477–1496.

[19] Hao He, Courtney Miller, Shyam Agarwal, Christian Kästner, and Bogdan Vasilescu. 2025. Does AI-Assisted Coding Deliver? A Difference-in-Differences Study of Cursor's Impact on Software Projects. *arXiv preprint arXiv:2511.04427* (2025).

[20] Junda He, Jieke Shi, Terry Yue Zhuo, Christoph Treude, Jiamou Sun, Zhenchang Xing, Xiaoning Du, and David Lo. 2025. From code to courtroom: Llms as the new software judges. *arXiv preprint arXiv:2503.02246* (2025).

[21] Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. 2025. Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. *arXiv preprint arXiv:2506.08872* 4 (2025).

[22] Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. 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*. 1–22.

[23] Brooke N Macnamara, Ibrahim Berber, M Cenk Çavuşoğlu, Elizabeth A Krupinski, Naren Nallapareddy, Noelle E Nelson, Philip J Smith, Amy L Wilson-Delfosse, and Soumya Ray. 2024. Does using artificial intelligence assistance accelerate skill decay and hinder skill development without performers' awareness? *Cognitive Research: Principles and Implications* 9, 1 (2024), 46.

[24] Shane McIntosh, Yasutaka Kamei, Bram Adams, and Ahmed E Hassan. 2016. An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering* 21, 5 (2016), 2146–2189.

[25] Kiana Jafari Meimandi, Gabriela Aránguiz-Dias, Grace Ra Kim, Lana Saadeddin, and Mykel J Kochenderfer. 2025. The Measurement Imbalance in Agentic AI Evaluation Undermines Industry Productivity Claims. *arXiv preprint arXiv:2506.02064* (2025).

[26] Maria Nordström. 2022. AI under great uncertainty: implications and decision strategies for public policy. *AI & society* 37, 4 (2022), 1703–1714.

[27] Shakked Noy and Whitney Zhang. 2023. Experimental evidence on the productivity effects of generative artificial intelligence. *Science* 381, 6654 (2023), 187–192.

[28] Mario Passalacqua, Robert Pellerin, Esma Yahia, Florian Magnani, Frédéric Rosin, Laurent Joblot, and Pierre-Majorique Léger. 2025. Practice with less AI makes perfect: partially automated AI during training leads to better worker motivation, engagement, and skill acquisition. *International Journal of Human–Computer Interaction* 41, 4 (2025), 2268–2288.

[29] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590* (2023).

[30] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. 2023. Do Users Write More Insecure Code with AI Assistants?. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2785–2799. doi:10.1145/3576915.3623157

[31] James Prather, Brent N Reeves, Juho Leinonen, Stephen MacNeil, Arisoa S Randrianasolo, Brett A Becker, Bailey Kimmel, Jared Wright, and Ben Briggs. 2024. The widening gap: The benefits and harms of generative ai for novice programmers. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 469–486.

[32] Christian Rahe and Walid Maalej. 2025. How Do Programming Students Use Generative AI? *Proceedings of the ACM on Software Engineering* 2, FSE (2025), 978–1000.

[33] Steven Randazzo, Hila Lifshitz-Assaf, Katherine Kellogg, Fabrizio Dell'Acqua, Ethan R Mollick, and Karim R Lakhani. 2024. Cyborgs, centaurs and self automators: Human-GenAI fused, directed and abdicated knowledge co-creation processes and their implications for skilling. *The Wharton School Research Paper* (2024).

[34] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M Howard, and Alan R Wagner. 2016. Overtrust of robots in emergency evacuation scenarios. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 101–108.

[35] Ian T Ruginski, Sarah H Creem-Regehr, Jeanine K Stefanucci, and Elizabeth Cashdan. 2019. GPS use negatively affects environmental learning through spatial transformation abilities. *Journal of Environmental Psychology* 64 (2019), 12–20.

[36] Judy Hanwen Shen and Alex Tamkin. 2026. How AI Impacts Skill Formation. *arXiv preprint arXiv:2601.20245* (2026).

[37] Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas W Mayer, and Padhraic Smyth. 2025. What large language models know and what people think they know. *Nature Machine Intelligence* 7, 2 (2025), 221–231.

[38] Damian A Tamburri, Philippe Kruchten, Patricia Lago, and Hans Van Vliet. 2013. What is social debt in software engineering?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 93–96.

[39] Damian A Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2015. Social debt in software engineering: insights from industry. *Journal of Internet Services and Applications* 6, 1 (2015), 10.

[40] Roel Wieringa. 2014. *Design science methodology for information systems and software engineering*. Springer.

[41] Feiyang Xu, Poonacha K. Medappa, Murat M. Tunc, Martijn Vroegindeweij, and Jan C. Fransoo. 2026. AI-Assisted Programming Decreases the Productivity of Experienced Developers by Increasing the Technical Debt and Maintenance Burden. arXiv:2510.10165 [econ.GN] https://arxiv.org/abs/2510.10165

[42] Lixiang Yan, Samuel Greiff, Ziwen Teuber, and Dragan Gašević. 2024. Promises and challenges of generative artificial intelligence for human learning. *Nature human behaviour* 8, 10 (2024), 1839–1850.

[43] Yue Zhang and Pascal Reusch. 2025. Trust in and Adoption of Generative AI in University Education: Opportunities, Challenges, and Implications. In *2025 IEEE Global Engineering Education Conference (EDUCON)*. 1–10. doi:10.1109/EDUCON62633.2025.11016490

[44] Zora Zhiruo Wang, Yijia Shao, Omar Shaikh, Daniel Fried, Graham Neubig, and Diyi Yang. 2025. How Do AI Agents Do Human Work? Comparing AI and Human Workflows Across Diverse Occupations. *arXiv e-prints* (2025), arXiv–2510.

Received 15 October 2025