Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 2709–2720
https://doi.org/10.54941/ahfe1008495

Measuring What Generative AI Actually Saves in Instructor-Led Training Development: A Matched-Deliverable Time-and-Quality Comparison

Richard Stone[1], Anthony Lavazza[1], Michael Jenkins[2], Claire Carter[1], Lixsa Barraza[1], and Meagan Ault[1]

[1]Stone Solutions + Research Collective LLC, Phoenix, AZ 85028, USA
[2]Knowmadics, Inc., Herndon, VA, 20171, USA

ABSTRACT

Instructor-led training (ILT) is a core delivery mode for workforce development, safety, equipment familiarization, and procedural instruction, and producing it is expensive. Generative Artificial Intelligence (AI) promises to cut that cost, and organizations want a number they can trust. The obstacle is measurement. The metrics that are easiest to capture, software-recorded editing time and time per slide, are unreliable, and the apparent advantage of AI can appear or vanish depending on what is counted. ILT slides are deliberately sparse because the instructor carries the content, so word count per slide says little about completeness, and recorded editing time counts only work done inside the slide file. This paper fixes the deliverable instead: a complete teachable module of defined instructional scope, including slides, instructor talking points, activities, and assessment items, sufficient for a qualified instructor to deliver. Development time is measured to that standard and normalized per learning objective and per minute of instruction. Three workflows produced the same module from the same source material: an experienced designer, a junior designer, and a designer working with generative AI followed by Subject Matter Expert (SME) review. The AI-assisted workflow reached a complete module in 5 person-hours against 32 and 56, but drafting fell 94 percent while review and revision rose to 65 percent of what remained. The contribution is a matched-deliverable method that lets organizations state AI productivity gains defensibly rather than rhetorically, with guidance for safe adoption in safety-critical domains.

Keywords: Generative AI, Instructor-led training, Instructional design, Productivity measurement, Safety training, Human-AI teaming

INTRODUCTION

Instructor-led training remains the default delivery mode wherever a task is physical, hazardous, equipment-dependent, or governed by a standard that requires demonstrated competence. Confined space entry, fall protection, aerial lift operation, trenching and excavation, and crane and rigging work are all taught in classrooms and on the job site by qualified instructors, and the courseware behind those sessions has to be built by someone. A designer

Received October 4, 2026; Revised November 16, 2026; Accepted November 21, 2026; Available online December 1, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

starts from regulatory text, manufacturer documentation, incident data, and site-specific procedures, and turns them into learning objectives, a lesson plan, an instructor guide, a slide deck, learner activities, and assessment items that a competent person can deliver and a compliance auditor can defend.

The cost of that work is well documented. A widely cited industry benchmark reported an average of 43 development hours per finished hour of instructor-led training, with a typical range from 22 to 82 hours depending on complexity and the amount of reusable source material (Chapman, 2010). At that ratio a single eight-hour safety course represents roughly 350 person-hours of design effort before any instructor stands in front of a class. Any tool that meaningfully reduces that number changes the economics of a training organization.

Generative AI is that tool, or is presented as one. Controlled studies in adjacent domains report substantial gains: professional writing tasks completed roughly 40 percent faster with higher rated quality (Noy and Zhang, 2023), consulting tasks inside the model's capability frontier completed faster and better while tasks outside it got worse (Dell'Acqua et al., 2023), customer support productivity up with the largest gains for the least experienced agents (Brynjolfsson, Li and Raymond, 2025), and software tasks completed in roughly half the time (Peng et al., 2023). Training organizations have read these results and want the equivalent figure for their own work.

The problem is that instructional design does not produce a single artifact whose quality can be scored in one pass. It produces a bundle of interdependent components whose sufficiency is judged by whether an instructor can teach from them. The effort metrics that authoring software records automatically do not measure that bundle, and they can be inflated or deflated by factors unrelated to the tool under study. This paper argues that the measurement problem has to be solved before any productivity claim about generative AI in ILT development is worth repeating, proposes a method that solves it, and applies that method in a controlled comparison.

WHY THE EASY METRICS MISLEAD

Two metrics dominate informal productivity claims in training organizations: how much content was produced, usually proxied by slide count or words per slide, and how long the file was open, harvested from the authoring application's own metadata. Neither measures development effort to a teachable standard. Instructor-led slides are intentionally sparse, because the deck is a visual anchor and the content lives in the instructor's delivery and the accompanying guide. This is consistent with multimedia learning research, which finds that redundant on-screen text competes with spoken narration for the same processing channel (Mayer, 2009; Sweller, Ayres and Kalyuga, 2011). A slide carrying one image and four words may represent

ten minutes of instruction; a slide dense with bullet text may represent a designer who has not yet decided what the instructor will say.

The module used in this study illustrates the problem. Its deck runs to 103 slides, but 94 are distinct: nine are exact duplicates produced by progressive reveals, where an agenda or a comparison table is built one element at a time across consecutive slides. The deck carries roughly 3,500 words, an average near 34 per slide, which reflects a decision about how much the instructor speaks rather than how much instruction the module contains. A metric that rewards slide text penalizes sound instructor-led design, and a metric that counts slides credits the same content several times.

Recorded editing time fails differently. PowerPoint stores a cumulative TotalTime value in a file's core properties, and because it accumulates automatically it is quoted as if it were a labor record. It is not. It counts time in the slide file only, so it misses SME review, source research, instructor guide authoring, activity design, and assessment writing. Across three production safety courses built by this organization, SME review alone accounted for between 22 and 60 percent of attributed development effort, none of it visible to the counter. Nor can the counter distinguish editing from a file left open overnight, and one inherited from a shared branded master carries other people's sessions forward. The exclusion is worst for AI comparisons specifically, because an AI-assisted workflow moves effort out of the slide file and into prompting, source review, and correction. Timing has to be anchored to a defined deliverable and captured across the whole workflow, not harvested from one application's metadata.

THE MATCHED-DELIVERABLE METHOD

The method fixes the output rather than the input. Instead of asking how much was produced in a given time, it asks how much time was required to produce a defined thing, and it defines that thing in terms an instructor and an auditor would both accept.

Defining a Complete Teachable Module

A complete teachable module is a unit of instruction with a defined scope that a qualified instructor could deliver without further development. The definition is operationalized as a completeness checklist (Table 1), derived from the developing organization's instructional design standard operating procedure so that the study measures the deliverable the organization already commits to producing rather than one invented for the experiment. A module is complete when every component is present, every learning objective is traceable to at least one content element and one assessment item, and an SME has signed off on technical accuracy against the governing source documents. Until all conditions are met the clock is still running.

Table 1: Completeness criteria for a teachable module.

Component	Criterion for Completeness
Learning objectives	Measurable, aligned to the governing standard or procedure, and covering the defined scope
Lesson plan and timing	Sequenced outline with time allocations that sum to the target instructional duration
Slide deck	Visual anchor for every lesson segment; sparse by design; branded template applied
Instructor talking points	Speaker notes or guide sufficient for a qualified instructor unfamiliar with the draft to deliver each segment
Learner activities	At least one activity per major objective with instructions, materials list, and debrief prompts
Assessment items	Knowledge check items mapped to each objective, with answer key and rationale
Source traceability	Each technical claim traceable to a cited regulation, standard, or manufacturer document
SME sign-off	Technical accuracy reviewed and accepted by a qualified subject-matter expert

The checklist is deliberately strict about traceability and sign-off. In safety-critical content a module that reads well but cites the wrong standard is not complete; it is a liability. Treating SME acceptance as part of the deliverable ensures that any time the AI-assisted workflow saves in drafting is netted against the time the SME spends correcting it.

Normalization

Modules differ in length and scope, so raw hours to completion are not directly comparable across modules. Two normalizations are used. Development hours per learning objective captures effort per unit of instructional intent; development hours per minute of delivered instruction captures effort per unit of classroom time and maps onto the industry benchmark ratio (Chapman, 2010). Reporting both guards against a workflow that inflates objective counts or pads delivery time to improve one ratio. Within a single matched comparison normalization is a reporting convenience; across organizations it is what makes results portable.

Study Design

The study is a matched-deliverable comparison of three development workflows producing the same module from identical inputs.

Conditions

Three conditions produced the module from identical inputs, all using the same authoring suite and all requiring SME review before the module could be declared complete. An experienced instructional designer with eight years

of ILT experience represents the current best-case human baseline. A designer with two years represents the typical staffing case in a growing training organization and gives a second point on the experience curve, which matters because prior work finds AI gains concentrate among less experienced workers (Brynjolfsson, Li and Raymond, 2025; Noy and Zhang, 2023). The AI-assisted condition pairs a designer with four years of experience with a general-purpose large language model under a defined prompt protocol.

MATERIALS

All conditions receive the same package: the governing regulatory text and consensus standards for the topic, the organization's branded master template, a fixed set of learning objectives, a target instructional duration, and the completeness checklist. The module used for this study covers operator-level use of the Honeywell BW Ultra portable multi-gas detector within a confined space entry course. Its governing sources are the Honeywell BW Ultra user manual and OSHA 29 CFR 1910.146. The module is organized in eight sections: the device and the gases it monitors, safety considerations and operating limits, screen symbology, alarms and response, daily operations including activation, bump test, zero, and calibration, configuration and connectivity, user maintenance, and course review. Target instructional duration is 90 minutes. A 30-minute hands-on practical follows the classroom module and is outside the scope of this comparison.

The AI-assisted condition additionally receives a prompt template set covering objective refinement, outline generation, talking-point drafting, activity design, knowledge-check generation, and first-pass source cross-reference, so that the condition tests a repeatable workflow rather than an individual's prompting skill.

PROCEDURE AND TIME CAPTURE

Designers log time contemporaneously at the task level using a fixed taxonomy: source review, objective and outline work, slide production, instructor guide authoring, activity design, assessment authoring, formatting, prompting and AI output review, SME review response, and revision. Application metadata is collected in parallel for comparison but is not used as the measure. Each condition passes the same checkpoints: draft submitted for SME review, corrections returned, revisions submitted, and completeness gate applied. Review cycles are counted, and SME review time is captured separately and included in total development time, because it is a real cost and because it is where the AI-assisted workflow is expected to incur the effort that drafting saves.

The AI-assisted condition runs inside a persistent workspace holding the organization's instructional design standard operating procedure, the branded slide template, and the course source documents as standing context, so none of it is restated in individual prompts. The model first drafts learning objectives and a section outline from the SOP and the sources; that outline then serves as the prompt for the slide deck. This is why the prompting and

slide production figures in Figure 2 are small: the designer is directing a workflow whose standards and layout are already fixed, and the effort that remains sits in reading the sources beforehand and correcting the output afterward.

The condition also differs in how SME feedback is applied. The SME returns corrections in the same form for all three conditions; in the AI-assisted condition the designer converts them into a revision prompt and the model applies them, rather than editing by hand. Revision time therefore includes prompt construction and verification of the model's edits, and all of it is counted. The model used was Anthropic Claude Opus 5. No controlled unclassified information, export-controlled material, or client-proprietary source document was placed into it.

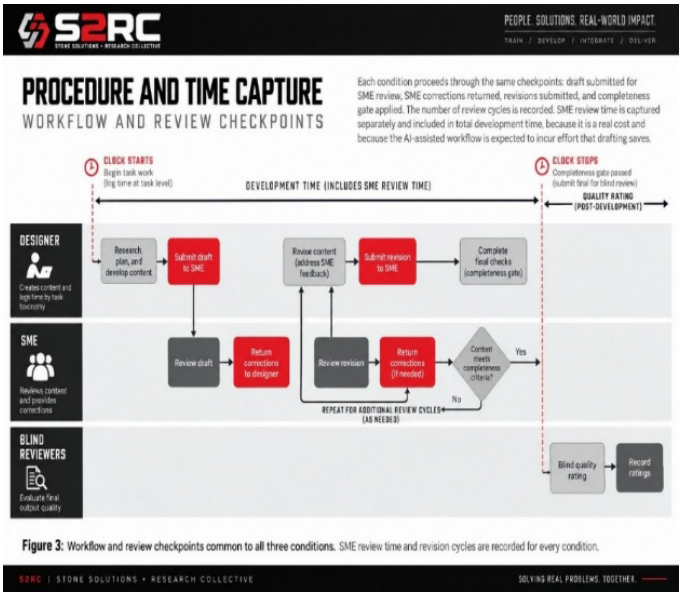


Figure 1: Workflow and review checkpoints common to all three conditions. SME review time and revision cycles are recorded for every condition.

Measures

The primary dependent variable is development time to a complete teachable module, in person-hours, inclusive of SME review and revision, reported raw and normalized per learning objective and per minute of instruction. Development time is logged at the task level using a fixed taxonomy and is not taken from application metadata, for the reasons given above.

Secondary measures are drawn from records the organization's standard operating procedure already requires, so they can be collected without instrumenting designers for the study: time by task category; the number of SME review cycles before acceptance; and the SME correction log, in which each correction is recorded against a slide or item and coded as technical

error, omission, standard misattribution, or editorial. Technical error and standard misattribution are reported separately, because in safety-critical content they are the categories with consequences.

Technical accuracy is measured through the source traceability record rather than rater judgment. The SOP requires a source register listing each controlling document with its revision, and requires every technical claim to be traceable to a register entry at section or paragraph level. Two quantities follow: the proportion of technical claims traceable at the point of SME acceptance, and the number whose cited source was wrong rather than missing. The second matters most for AI-assisted development, because a model produces citations that look correct, and a missing citation is visible to a reviewer in a way that a plausible wrong one is not. Completeness is measured against the criteria in Table 1, and the module is not counted as finished until the SME accepts it, the source register is complete, and the correction log is closed.

With one module per condition, results are reported descriptively rather than as null-hypothesis tests, and the study is framed as a controlled case comparison and method demonstration rather than a powered experiment. Time figures are reported to the nearest hour as logged; correction counts are reported as counts, not rates, because the denominator differs across conditions.

RESULTS

Development time to a complete teachable module, including SME review and revision, is reported in Table 2 for the three conditions. All three modules reached the completeness gate defined in Table 1.

Table 2: Development time to a complete teachable module by condition.

Condition	Total Hours	Hours per Objectivew	Hours per Minute of Instruction	SME Review Cycles
Experienced human	*32*	*6.4*	*0.53*	*1 (No Edits)*
Junior human	*56*	*11.2*	*0.933*	*4*
AI-assisted	*5*	*1*	*0.083*	*2*

The AI-assisted workflow reached a complete module in 5 person-hours against 32 for the experienced designer and 56 for the junior designer, a reduction of 84 percent against the experienced baseline and 91 percent against the junior. Normalized per minute of instruction it required 0.083 hours against 0.53 and 0.93. In the benchmark's own units the three conditions correspond to ratios of about 32:1, 56:1, and 5:1 per finished hour of instruction, against an industry average of 43:1 (Chapman, 2010): the experienced designer beat that benchmark, the junior designer did not, and the AI-assisted workflow was roughly an order of magnitude below it.

Review cycles moved in the opposite direction from drafting time. The experienced designer's draft was accepted on first submission with no

edits; the AI-assisted draft required two cycles and the junior designer's four. The AI-assisted condition did not eliminate review effort, but it needed less correction than the junior draft and, because drafting was fast, far less total time. The 5-hour figure includes both cycles and the associated revision, so the comparison is on a matched deliverable, not first-draft speed.

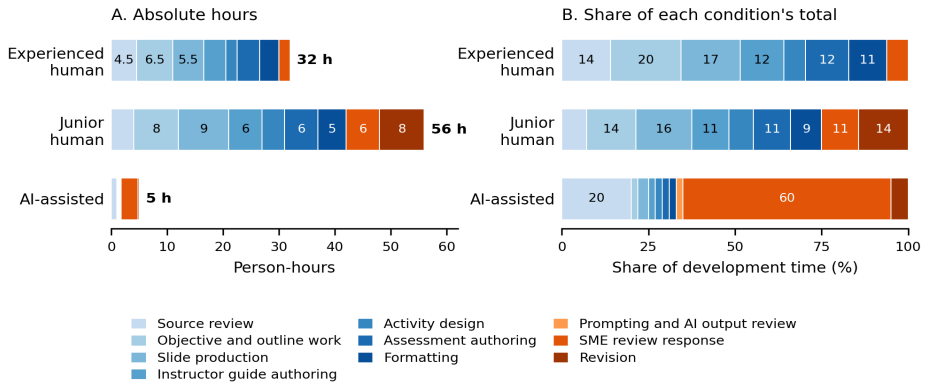


Figure 2: Development time by task category and condition. Panel A shows absolute person-hours; Panel B shows each condition's own composition, because at 5 hours against 56 the AI-assisted bar is unreadable on an absolute scale. Hours for the experienced and AI-assisted conditions are from project time tracking; the junior condition's task split is estimated.

Figure 2 shows where the hours went. The seven drafting categories, source review through formatting, accounted for 30.0 hours in the experienced condition and 42.0 in the junior condition, against 1.65 in the AI-assisted condition: a reduction of 94 percent against the experienced baseline. Review and revision did not fall proportionally with drafting effort. Together, SME review and revision accounted for 3.25 of the AI-assisted condition's 5 hours, or 65 percent of total development time, compared with 6 percent for the experienced designer and 25 percent for the junior designer. Of this, approximately 3.0 hours were associated with SME review and feedback. The additional review burden was driven largely by instructional-design issues, particularly recommendations concerning lesson sequencing, classroom flow, and how information should be presented to learners. Thus, the AI-assisted workflow substantially reduced production effort but increased the proportion of time devoted to expert review of how the course would actually be taught.

Source review was the largest AI-assisted drafting category at 1.0 hour, a fifth of that condition's total: the designer still had to read the governing documents, and that step did not compress. Prompting and review of model output took a tenth of an hour, small enough that the constraint on this workflow is the reading before it and the correction after it, not the interaction with the model.

Time for the experienced and AI-assisted conditions was recorded in the organization's issue tracker as the work was performed. The junior

condition's task-level split was estimated from a tracker data pull rather than logged category by category, so its total is firm and its internal distribution approximate. The contrast the figure carries runs between the AI-assisted condition and the experienced baseline, both tracked directly.

DISCUSSION

Where Gains are Real and Where They Move

Generative AI is strongest on components that are structured, repetitive, and bounded by the source material: drafting objectives from a standard, producing an outline, generating knowledge-check items, writing facilitator prompts, and formatting. These are the tasks prior literature places inside the capability frontier (Dell'Acqua et al., 2023). It is weakest on instructional flow, on judging what belongs on the slide versus in the notes, and on technical accuracy where the governing standard has exceptions, thresholds, and jurisdictional variation. The matched-deliverable method surfaces this because it counts SME correction time against the AI-assisted condition: a workflow that produces a draft in an hour and requires four hours of correction has not saved four hours, it has moved them, and possibly to a more expensive person.

The practical framing is augmentation. The designer remains the author of instructional flow and the arbiter of what the module needs; the AI accelerates first drafts and routine components and shifts the designer's time toward design, sequencing, and quality assurance. This matches the reallocation reported in writing tasks, where AI substituted for rough-drafting effort and moved workers toward editing and idea generation (Noy and Zhang, 2023).

The trial measures that reallocation rather than assuming it. The AI-assisted workflow reached a complete module in 5 person-hours against 32 and 56 for the experienced and junior human conditions, respectively. Drafting effort fell 94 percent relative to the experienced-human baseline, but the work shifted toward quality assurance: SME review and revision accounted for 3.25 hours, or 65 percent of total AI-assisted development time. Much of this additional SME effort involved recommendations to improve instructional sequencing, classroom flow, and how content would be presented during delivery.

The 65 percent review-and-revision share observed here should not necessarily be interpreted as a fixed cost of AI-assisted development. As prompt protocols mature and the system is provided with stronger examples of accepted instructional sequencing, classroom flow, and course structure, first-pass outputs may better reflect organizational instructional expectations and reduce subsequent SME review effort. Repeated incorporation of SME feedback into prompt templates and reference examples could further improve first-pass alignment. This study did not measure that learning effect, however, and longitudinal research across repeated course-development cycles is needed to determine whether the review burden decreases as the workflow matures.

Safety-Critical Content and the Limits of AI Review

In safety training, technical accuracy and regulatory compliance are not quality dimensions among others; they are gating conditions. A model can produce a first-pass cross-reference between module content and a cited standard, which is useful for catching omissions early, but it is not assurance. Compliance assurance is the responsibility of a qualified human reviewer, and the method encodes that by making SME sign-off part of the deliverable definition. Organizations that report AI productivity gains without including SME review time are reporting a number that excludes the step that makes the content usable.

Limitations

The study compares one module built by one designer per condition, so results are indicative rather than statistically general, and the reported reduction is a demonstration of the method on a single matched deliverable, not an expected value. Designer experience is confounded with condition, so the comparison against the experienced designer is the more conservative one. A single SME reviewed all three modules, in the order experienced, junior, AI-assisted, so the AI-assisted module was reviewed by the reader most familiar with the content. Quality was assessed through SME acceptance and the correction record, not independent blind rating, so the study reports whether each module was accepted and what had to be fixed. The junior condition's task-level distribution was estimated rather than logged by category. The AI-assisted workflow depends on a persistent workspace preloaded with the SOP, template, and sources, a setup cost paid once rather than per module, so an organization without it would see less of this reduction. The prompt protocol reflects a specific model generation and will age. The module is a short, equipment-focused unit built from one manufacturer manual and one regulatory citation; modules synthesizing multiple standards with exceptions and jurisdictional variation should incur more correction, and this trial does not test that case. No controlled unclassified or otherwise restricted source material was used, so the trial does not measure the overhead a compliant deployment would add.

PRACTICAL GUIDANCE FOR ADOPTION

Organizations that want a defensible productivity figure for generative AI in ILT development can apply the following without replicating the full study.

Define the deliverable before anyone starts a clock, using Table 1 as a starting point, and treat SME sign-off as part of it. Report hours per learning objective and per minute of instruction alongside raw hours, against a published benchmark such as the 43:1 ILT ratio (Chapman, 2010) so the figure has external context. Capture time with a contemporaneous task-level log rather than application metadata. Publish the prompt templates internally so gains are attributable to a repeatable process rather than one designer.

Require SME review before any module is declared complete, and log the corrections rather than only the cycles, coded as technical error, omission, standard misattribution, or editorial; the count of technical errors and misattributions is the only number that tells an organization whether review can safely be shortened. Make technical accuracy auditable rather than rated: keep a source register for each course, require each technical claim to be traceable to an entry at section or paragraph level, and never carry forward a citation an AI tool produced without opening the source. In safety-critical domains, restrict AI-generated content to draft status until human review, prohibit AI-only compliance sign-off, and document which model and template version produced each draft. Source material carrying controlled or proprietary data cannot run on a general-purpose commercial model, and the approval and hosting overhead that requires should be counted against the productivity figure rather than assumed away.

CONCLUSION

The question organizations ask is how much generative AI saves in building instructor-led training. The honest answer begins with a different question: saves relative to what deliverable, measured how. Slide text density and application-recorded editing time both fail, in ways that can flatter or penalize AI depending on which is read. Fixing the deliverable as a complete teachable module, timing the full workflow to that standard including SME review, and normalizing per objective and per minute of instruction produces a defensible figure. Applied to one confined space module it produced 5 hours against 32 and 56, with drafting down 94 percent and review and revision rising to 65 percent of what remained. The method also makes visible where gains are real and where effort has merely moved. In safety-critical instruction that visibility matters as much as the number.

ACKNOWLEDGMENT

Generative AI disclosure: Claude (Anthropic) was used in preparing this manuscript. It assisted with reviewing the manuscript for internal consistency and unsupported claims. All data, study design decisions, and technical content are the authors' own, and the authors reviewed and take responsibility for the final text. This use is separate from the AI-assisted workflow evaluated in the study.

REFERENCES

Brynjolfsson, E., Li, D. and Raymond, L. (2025). Generative AI at work. The Quarterly Journal of Economics, 140(2), pp. 889–942.

Chapman, B. (2010). How long does it take to create learning? Research study. Chapman Alliance LLC. Available at: http://www.chapmanalliance.com/howlong

Dell'Acqua, F., McFowland, E., Mollick, E., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F. and Lakhani, K. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. Harvard Business School Working Paper 24-013.

Mayer, R.E. (2009). Multimedia learning. 2nd ed. New York: Cambridge University Press.

Noy, S. and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. Science, 381(6654), pp. 187–192.

Peng, S., Kalliamvakou, E., Cihon, P. and Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv preprint arXiv:2302.06590.

Sweller, J., Ayres, P. and Kalyuga, S. (2011). Cognitive load theory. New York: Springer.