Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 529–537
https://doi.org/10.54941/ahfe1008285

Creating a Path to Mastery: Impacts of Machine-Learning Guidance on Novice Software Users

Melissa Peterson

San Jose State University Research Foundation, USA

ABSTRACT

Novice users of complex software often face a steep learning curve, and many disengage before reaching proficiency. This study examined whether machine-learning-guided, in-application scaffolding can lower that barrier. Using cognitive apprenticeship theory (Collins, 2012) as a framework, an online study with U.S. science teachers tested a novel creative software environment, the DaVinci Science Comic Maker (DSCM). In the experimental condition, a machine-learning algorithm adjusted the complexity of available tools and supports to each user's demonstrated skill level; the control condition offered no scaffolding and full tool complexity from the outset, mirroring typical unsupported software adoption. Participants created three comics within DSCM and completed a survey drawn from the Unified Theory of Acceptance and Use of Technology (UTAUT; Venkatesh et al., 2003). Three data sources were analyzed: artifact complexity and efficiency scores, clickstream data, and UTAUT responses. Experimental-group participants produced increasingly complex comics across trials ($F(2,18) = 17.60$, $p < .001$), while control-group complexity did not change ($F(2,20) = 0.36$, $p = .705$). The experimental group also used fewer clicks to complete its first comic. No significant between-group differences emerged on UTAUT subscales. These findings suggest that machine-learning-enabled cognitive apprenticeship scaffolding can support measurable skill growth among novice software users without external training, with implications for reducing adoption barriers in educational contexts and for the future design of complex creative software environments.

Keywords: Machine learning, Cognitive apprenticeship, Adaptive scaffolding, Novice software users, Human-computer interaction

INTRODUCTION

For novice software users, bridging the gap between current ability and future, skilled use of a tool can be daunting. Mastery may require thousands of hours; many novices disengage before they reach proficiency or adopt the tool. This paper explores a potential solution to that gap: a machine-learning-driven guide that adjusts to a user's demonstrated skill level and supports effective use of new software tools from the moment a user opens them. The guidance is grounded in cognitive apprenticeship and situated learning, providing scaffolding for procedural and design skills while users interact with the software.

Received August 26, 2026; Revised September 28, 2026; Accepted October 8, 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/

In situated learning theory, knowledge is inherently linked to the physical, social, and cognitive contexts in which that knowledge will be used (Schuh and Barab, 2008). This theory is relevant to the current research because informal learning happens in the workplace (Hoekstra et al., 2009), where skills are learned in the context in which they will be used. Further, when an expert mentor is included in this system, it can function as a cognitive apprenticeship system (Collins, 2012), providing more guidance to a learner rather than relying on self-directed learning or exploration learning. In this study, the AI functions as a non-ideal replacement for a human mentor. This differs from traditional learning systems like professional development workshops or tutorials because the learning is hands-on and focused on successfully creating something in the software, rather than learning to use the software more generally.

Recent work on scaffolded interfaces for professional creative software offers a close parallel to this approach. Liu and Sra (2025, 2026) introduced ScaffoldUI and its successor, TaskLens, both of which embed task-relevant tools, progressive complexity disclosure, and domain-concept labelling directly into professional 3D software (Blender), rather than relying on external tutorials or simplified standalone applications. Across two studies, their scaffolded interfaces significantly reduced perceived task load and improved task performance and concept learning for novice users compared to the default software interface. However, in both studies, users self-selected their own complexity level, and the authors explicitly identify system-driven interface adaptation, in which the software itself determines when a user is ready for more complexity, as an open direction for future work. The present study addresses this gap: rather than relying on user-selected complexity levels, the DSCM's machine-learning facilitator classifies user skill level from clickstream behavior and adjusts scaffolding accordingly, without requiring the user to judge their own readiness to advance.

This study tested this system with science teachers in the United States. This population was chosen as the focus of this study for two reasons: first, teachers are asked to quickly integrate new technology into their work every year with little to no support (Gray and Lewis, 2021), and second, in the computer science literature, teachers are an audience described as having a lower capacity for learning complex computer skills and coding (Murray, 2016), although it is unclear whether this has been empirically investigated. Thus, teachers, as a group, may have a strong need for interventions of this kind.

Two research questions were posed for this study, as follows:

1. What is the effect of AI-facilitated cognitive apprenticeship scaffolds on science teachers' learning of novel software?
2. What is the effect of AI-facilitated cognitive apprenticeship scaffolds on science teachers' performance expectancy, effort expectancy, attitudes, and self-efficacy while learning to use novel software?

METHODS

An online study was conducted with 21 science teachers in the United States to test in-application cognitive apprenticeship scaffolding as an intervention for novice users of the novel creative software environment DaVinci Science Comic Maker (DSCM). Participants were recruited through professional networks and social media and were asked to commit to 35-45 minutes of science comic making. They then participated in a brief eligibility survey to confirm that they were full-time, currently working science teachers in grades 4-10 in the US, and then went through informed consent procedures. Participants created three comics within DSCM and completed a final survey that included four subscales (Performance Expectancy, Effort Expectancy, Attitude Towards Use, and Self-Efficacy) from the Unified Theory of Acceptance and Use of Technology (16 Likert-style questions, rated 1-7) (UTAUT; Venkatesh et al., 2003) (see Figure 1).

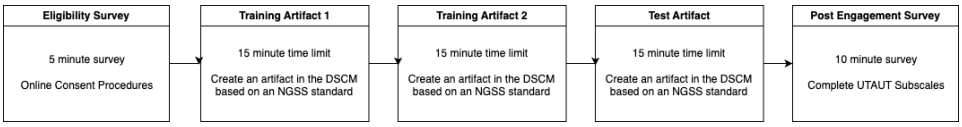


Figure 1: Experimental tasks and timing.

In-application scaffolding was controlled by a machine-learning skill-classification algorithm that provided skill-based support rather than a rigid curriculum of external tutorials: as users demonstrated they had learned the software, they were reclassified from Novice to Intermediate to Advanced, and received less support and access to more complex tools within the environment (see Table 1).

This scaffolded (experimental) condition was compared with a control condition in which participants received no support and access only to the most complex tools—designed to mimic the "normal," often unsupported experience of learning new creative or development software when outside help, tutorials, or training time are not available. Participants answered three 7-point Likert-style opinion questions to measure whether groups matched in attitudes towards comics and technology, and were then randomly assigned to control and experimental groups upon logging into the DSCM.

Three sets of experimental data were collected: (1) complexity and efficiency scores from the artifact-comics created within DSCM; (2) clickstream data, comprising the timing and type of mouse actions within DSCM (e.g. button click, pick up an asset, drop an asset); and (3) UTAUT survey data. Artifact complexity and efficiency scores were computed using a simple counting metric, and those scores were verified by an intra-rater reliability protocol. Complexity was a count of the number of assets (comic panels, images, or text), plus a count of the number of edits made to those assets. Efficiency was then measured as the complexity score divided by the total number of clicks used to create the artifact.

Each artifact was scored when collected, and then again during final analysis. The percentage agreement between the two times was 77.7%,

with an allowed error of two points. Final numbers were verified for mismatches with a third coding after a delay. Percentage agreement is a reasonable measure of agreement that is neither overly conservative nor permissive (Zhao et al., 2022) and is acceptable for most fields (Graham et al., 2012).

Artifact scores were analyzed using ANOVA for within-group analysis and t-tests for between-group analysis to identify evidence of learning over time and skilled performance; the remaining clickstream and UTAUT data were analyzed using between-group t-tests.

MATERIALS: DSCM SCAFFOLDING AND DESIGN

The DSCM was built using the Godot engine in GDScript and exported to HTML5 for online use. The AI facilitator was created using the scikit-learn Python machine learning library. Both of these parts (i.e., the AI facilitator and the DSCM interface) communicate with a MongoDB database via application programming interfaces (APIs), which allowed data to be fed to the AI to dynamically update the application interface.

An important feature of the DSCM is its incorporation of an AI facilitator. The AI facilitator is designed to support DSCM users by providing specific scaffolds. The scaffolds cover two distinct skill domains: design skills for science comics and technology skills for using the DSCM. Table 1 describes these scaffolds in detail.

Table 1: Scaffolds to support skills by user level.

User Level	Technology Scaffold	Design Scaffold
Novice	Panel template provided for user to fill in, limited tool options	Master exemplar provided as a design guide
Intermediate	Simple tools available for user to build comics	Subject specific prompts provided to guide user designs
Advanced	Complex tools available for user to build comics	Asset libraries provided with no external prompts to guide user designs

The AI facilitator uses clickstream data to categorize the user into one of three software user types: novice, intermediate, or advanced. In other words, the user is dynamically categorized by the AI facilitator as more skillful as their clickstream changes over time. As the user becomes more skillful and moves up through the software user types, the AI facilitator provides less scaffolding and more options related to the content and which tools are available within the environment. Model clickstream data from the development process and clickstream data from early beta testing were used to initially train the AI facilitator to categorize users.

The DSCM uses a direct-manipulation interface (Shneiderman, 1997) rather than a menu-based interface. This drag-and-drop style interface allows users to reuse the same skills across multiple tasks. This is an important

consideration in software design, as the ability to reuse skills makes software more learnable, since reusability aligns with the principles of predictability, generalizability, and consistency (Dix et al., 2003). Further, as found by Roldán-Álvarez et al. (2016), direct manipulation interfaces are quicker to learn, and the skills learned are more robust over time than those learned with a menu-based interface, even a familiar one.

In the context of the DSCM, the direct manipulation interface is shown in Figure 2: users can decide what they wish to add to the comic, and then drag the asset into the workspace, positioning it as desired with the mouse. All elements of the comic can be manipulated in this way, with the exception of panels, which are placed automatically in the novice and intermediate interfaces, and are placed using a coordinate system in the advanced and control interfaces. As discussed above, this design made the DSCM easier to learn than traditional menu-based software.

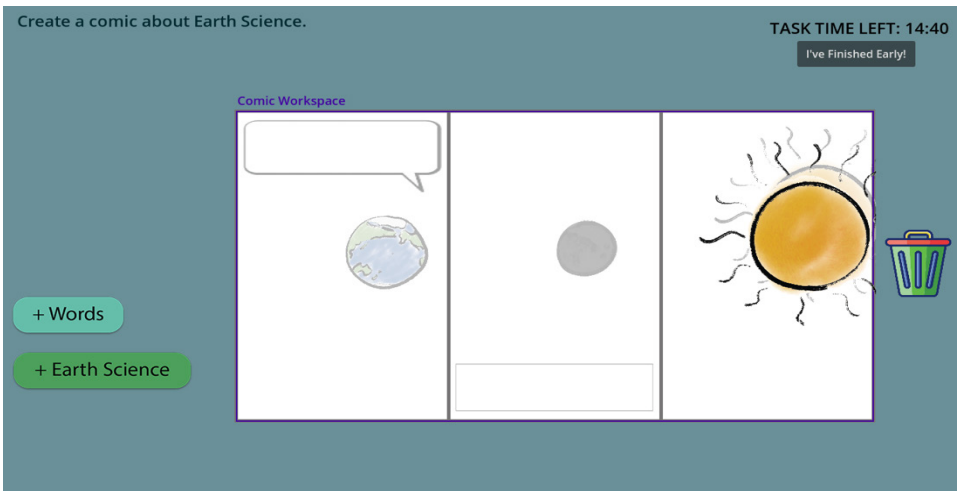


Figure 2: Direct manipulation interface with novice scaffolds.

RESULTS

The analysis for RQ1 tested the hypothesis that the experimental group would improve their complexity and efficiency scores over time; therefore, repeated measures ANOVAs were performed on these two measures across the three artifacts within each group. Efficiency scores did not improve across trials for either group, $F(2,20) = 0.62$, $p = .546$ for the control group, and $F(2,18) = 1.81$, $p = .192$ for the experimental group. Complexity scores changed only for the experimental group.

Figure 3 plots the mean complexity for each artifact by group. This plot shows that the experimental group improved their complexity over time, as hypothesized, given access to more tools and assets. The control group, also as expected, did not seem to improve across artifacts.

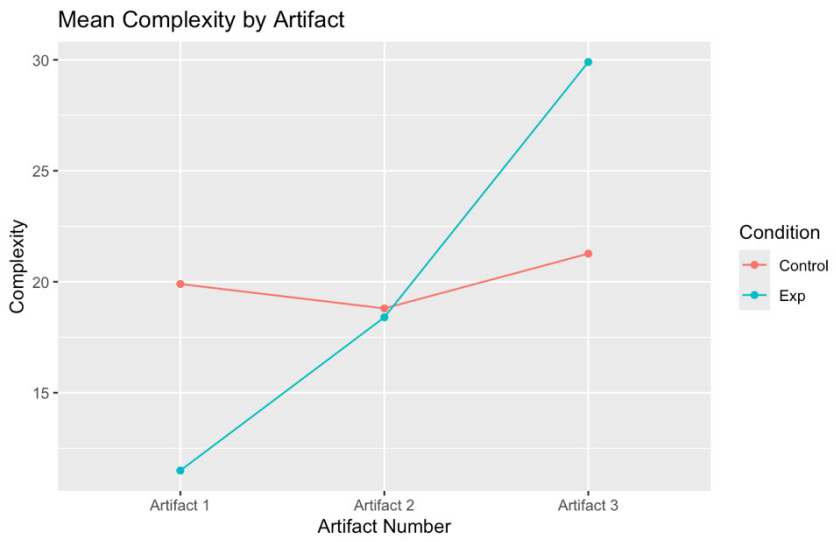


Figure 3: Mean complexity by artifact.

The ANOVA for artifact complexity showed that participants in the experimental group created increasingly complex comics across their three trials within DSCM, $F(2,18) = 17.60$, $p < .001$. A post hoc pairwise t-test using Tukey's Honestly Significant Difference (HSD) was performed to determine significant differences in complexity between artifacts within the experimental group (see Table 2). This showed that Artifact 3 was significantly more complex than either Artifact 1 or 2. Although the mean complexity of Artifact 2 was higher than that for Artifact 1, this difference was not significant.

Table 2: Pairwise Tukey's HSD T-tests of artifact complexity (experimental group only).

Artifact Comparison	Artifact	M	SD	*df*	*t*	*p*
Artifact 1 - Artifact 2	1	11.50	2.84	27	−1.90	0.16
	2	18.40	10.60			
Artifact 1 - Artifact 3	1	11.50	2.84	27	−5.07	<0.001***
	3	29.90	8.81			
Artifact 2 - Artifact 3	2	18.40	10.60	27	−3.17	0.01**
	3	29.90	8.81			

Note: ** $p \leq 0.01$; *** $p < 0.001$

In contrast, the control group ANOVA showed that they produced comics of consistent complexity across trials, with no increase over time, $F(2,20) = 0.36$, $p = .705$.

Group differences in clickstream data (also used for RQ1) were limited to the experimental group, which used significantly fewer clicks to create its first artifact (see Table 3) – an effect attributable to the smaller number of interface elements available during the scaffolded learning experience.

No significant differences emerged between groups on any UTAUT subscale, providing no support for affective changes based on condition during the study.

Table 3: Analysis of artifact 1 clicks by group.

Measure	Group	*n*	*M*	*SD*	*df*	*t*	*p*
Clicks	Control	11	66.27	41.93	19	2.45	0.024*
	Exp	10	27.70	28.25			

Note: $*p < .05$

DISCUSSION

This study showed that user capacity for creating complex artifacts improved when provided with AI-facilitated scaffolding for a novel creative software environment. This is particularly interesting because the control group did not improve on any measures, so their initial skill level remained constant across all three trials. While efficiency scores and clickstream-related measures showed no improvement for either group, the experimental group's initially lower number of clicks to create an artifact suggests the interface worked as designed by providing a simpler starting point before moving to more complex capabilities. However, the study found no evidence that this changed teachers' attitudes toward the software or the learning process. Future work should investigate this finding in more depth, as attitudes are a significant predictor of long-term software adoption (Venkatesh et al., 2003).

The potential impacts of this scaffolding approach may be far-reaching. For software development and educational technology companies, dynamic, skill-based scaffolds of this kind may substantially lower barriers to entry for new users, as they do not require users to leave the software to find training. They may also help to retain more users over the long term. As supports are tailored to each user, they may offer an individualized experience that is more effective for skill growth than group workshops or training (Phillips et al., 2020). In classrooms, technologies of this kind may allow teachers to generate individualized content for students more quickly, without needing to vet external sources or learn to engineer prompts for generative AI.

LIMITATIONS AND FUTURE RESEARCH

The most obvious limitation of this study is the low sample size, despite extensive recruitment efforts. This may be due to one of the main problems this study addresses: teachers lack time for additional tasks beyond their required classroom work, and already work unpaid hours every day (Steiner et al., 2023). A low sample size is a significant limitation, reducing the generalizability of the results presented here and affecting the types of analyses that could be conducted.

This study was an early exploration of machine-learning-enabled, in-application software skills support; at the time of writing, few studies had addressed this space, constituting a gap in the literature that warrants

further research. For instance, while this study showed differences between the control group without scaffolds and the experimental group with AI-facilitated scaffolds, it did not specifically examine whether the AI-facilitation was effective. In other words, whether the effect was due to scaffolds such as templates, prompts, and increased affordances, to the AI providing dynamic changes, or to both working together. A further study would be advisable to determine which factors in the experimental group directly contributed to the observed increase in learning. That study could also address the potential for user variability as a covariate of performance, which was beyond the scope of the current effort. Another interesting avenue for future research is to combine the current solution with large-language-model-based GenAI to potentially provide more accurate skill classifications, as this has been shown to increase classification accuracy in other domains, such as literacy (Palma and Soto, 2025).

CONCLUSION

The results presented here are encouraging. While they do not show an improved experience for scaffolded learners across all measures, they do indicate significant effects of machine-learning-enabled cognitive apprenticeship guidance on software learning. This is an important finding: it suggests that novice users of creative software can be supported by an ML-driven adaptive interface and successfully use novel software immediately, without external training—a benefit that is especially valuable in educational institutions, where funding and time for software training are often limited. This study ultimately demonstrated that, as designed, the system warrants further exploration as a tool to support novices in learning to use novel creative software, both in schools and at work.

ACKNOWLEDGMENT

This study was determined exempt by the Institutional Review Board of the University of Hawai‘i at Mānoa (Protocol # 2023-00183).

REFERENCES

Collins, A. (2012) ‘Cognitive apprenticeship’, in Sawyer, R.K. (ed.) The Cambridge Handbook of the Learning Sciences. Cambridge: Cambridge University Press, pp. 47–60. Available at: https://doi.org/10.1017/CBO9780511816833.005

Dix, A., Finlay, J., Abowd, G. and Beale, R. (2003) Human-Computer Interaction. 3rd edn. Harlow: Pearson Education/Prentice Hall.

Graham, M.J., Milanowski, A.T. and Miller, J.B. (2012) Measuring and Promoting Inter-Rater Agreement of Teacher and Principal Performance Ratings. Washington, DC: Center for Educator Compensation Reform. Available at: https://files.eric.ed.gov/fulltext/ED532068.pdf

Gray, L. and Lewis, L. (2021) Use of Educational Technology for Instruction in Public Schools: 2019–20. Washington, DC: National Center for Education Statistics, Institute of Education Sciences. Available at: https://nces.ed.gov/pubs2021/2021017.pdf

Hoekstra, A., Korthagen, F., Brekelmans, M., Beijaard, D. and Imants, J. (2009) 'Experienced teachers' informal workplace learning and perceptions of workplace conditions', Journal of Workplace Learning, 21(4), pp. 276–298. Available at: https://doi.org/10.1108/13665620910954193

Liu, Y. and Sra, M. (2025) 'Designing scaffolded interfaces for enhanced learning and performance in professional software', arXiv [cs.HC]. Available at: https://doi.org/10.48550/arXiv.2505.12101

Liu, Y. and Sra, M. (2026) 'TaskLens: Generating task-conditioned scaffolded interfaces for learning professional creative software', in Proceedings of the 2026 Designing Interactive Systems Conference. DIS '26: Designing Interactive Systems Conference, New York, NY, USA: ACM. Available at: https://doi.org/10.1145/3800645.3813081

Murray, T. (2016) 'Coordinating the Complexity of Tools, Tasks, and Users: On Theory-based Approaches to Authoring Tool Usability', International Journal of Artificial Intelligence in Education, 26(1), pp. 37–71.

Palma, D. and Soto, C. (2025) 'Combining large language models with linguistic features for the readability complexity assessment of texts', in AHFE International. AHFE 2025 Hawaii International Conference Human Factors in Design, Engineering, and Computing, AHFE International. Available at: https://doi.org/10.54941/ahfe1007028.

Phillips, A., Pane, J.F., Reumann-Moore, R. and Shenbanjo, O. (2020) 'Implementing an adaptive intelligent tutoring system as an instructional supplement', Educational Technology Research and Development, 68, pp. 1409–1437. Available at: https://doi.org/10.1007/s11423-020-09745-w

Roldán-Álvarez, D., Martín, E., García-Herranz, M. and Haya, P.A. (2016) 'Mind the gap: Impact on learnability of user interface design of authoring tools for teachers', International Journal of Human-Computer Studies, 94, pp. 18–34. Available at: https://doi.org/10.1016/j.ijhcs.2016.04.011

Schuh, K.L. and Barab, S.A. (2008) 'Philosophical Perspectives', in Spector, J.M. et al. (eds) Handbook of Research on Educational Communications and Technology. 3rd edn. New York: Routledge, pp. 67–82. Available at: https://doi.org/10.4324/9780203880869-8

Shneiderman, B. (1997) 'Direct manipulation versus agents: Paths to predictable, controllable and comprehensible interfaces', in Bradshaw, J.M. (ed.) Software Agents. Cambridge, MA: MIT Press, pp. 97–106.

Steiner, E.D., Woo, A. and Doan, S. (2023) All Work and No Pay—Teachers' Perceptions of Their Pay and Hours Worked: Findings from the 2023 State of the American Teacher Survey. Santa Monica, CA: RAND Corporation (RR-A1108-9). Available at: https://www.rand.org/pubs/research_reports/RRA1108-9.html

Venkatesh, V., Morris, M.G., Davis, G.B. and Davis, F.D. (2003) 'User acceptance of information technology: Toward a unified view', MIS Quarterly, 27(3), pp. 425–478. Available at: https://doi.org/10.2307/30036540

Zhao, X., Feng, G.C., Ao, S.H. and Liu, P.L. (2022) 'Interrater reliability estimators tested against true interrater reliabilities', BMC Medical Research Methodology, 22(1), p. 232. Available at: https://doi.org/10.1186/s12874-022-01707-5