Revibing HCI: Agentic Reimplementation as Community Practice

Yoonjoo Lee
lyoonjoo@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Tae Soo Kim
taesoo.kim@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Kevin Pu
kevin.pu@mail.utoronto.ca
University of Toronto
Toronto, Ontario, Canada

Mira Dontcheva
mirad@adobe.com
Adobe Research
Seattle, Washington, USA

Bjoern Hartmann
bjoern@eecs.berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Toby Jia-Jun Li
toby.j.li@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Brad A Myers
bam@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Jeffrey Nichols
jwnichols@apple.com
Apple
Seattle, Washington, USA

April Yi Wang
april.wang@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Xuhai Xu
xx2489@columbia.edu
Columbia University
New York, New York, USA

Eytan Adar
eadar@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Abstract

Agentic AI coding tools can now reimplement, or *"revibe,"* the systems described in research papers. *Revibing* could give the HCI community what it has long lacked: prior systems that can be run, act as strong baselines, and be extended. However, before *revibing* becomes common practice, our community must shape norms around how *revibes* are conducted, how they are verified, and what widespread *revibing* means for how HCI research is practiced, shared, and valued. Our workshop brings together the UIST community to discuss and form these early norms. Participants will *revibe* a classic UIST system in a hands-on session, brainstorm evaluation approaches for *revibed* artifacts, and discuss what *revibing* means for research methods, open science, and the future of HCI system contributions. The workshop aims to foster a community around *revibing* by consolidating guidance and outputs into a shared community document.

1 Workshop Motivation and Goal

Agentic AI coding tools are reshaping the process through which research systems are created. Given a natural language description, these tools can now produce entire code repositories [6, 13, 16], allowing HCI researchers to rapidly transform ideas into functional interactive systems. Beyond this, agentic coding tools also present a unique opportunity to re-engage with existing literature. Recent work by Adar et al. [1] demonstrated that coding agents can *revibe* recent UIST systems—i.e., produce faithful reimplementations from only the paper describing them. This opens up the question of *revibing* the many kinds of artifacts that the HCI community produces, from studies and algorithms to datasets and design probes. This emerging capability can fundamentally shift how we treat prior literature and how HCI research is conducted: past papers become living baselines rather than static citations.

Revibing provides new means to address a long-standing concern in HCI: research is rarely replicated and built upon. Replications are scarce [4, 14], as few as 2-3.5% of CHI authors open source their code [2, 12], and reimplementing prior systems is costly [8], leaving new systems evaluated in isolation through demonstrations [7] or usability studies [3]. Meanwhile, other communities are already building on these agentic coding tools to replicate papers [9–11]. Interactive systems raise challenges that these efforts do not capture, such as the faithfulness of interactive behavior and evaluation with users. This will require the HCI community to define its own norms and practices around this new capability.

This workshop brings together the UIST community, one that cares about both system building and evaluation and is well-positioned to shape early norms around agentic reimplementation, to reflect on the potential, limitations, and possible risks of *revibing* past artifacts. Specifically, this workshop has three goals:

Build shared, hands-on understanding of *revibing*. Through a live *revibing* session and showcase, participants collectively experience the potential and limitations of agentic reimplementation by producing concrete artifacts that ground the day's discussions.

Develop and test evaluation approaches for *revibed* artifacts. *Revibed* systems can deviate from the original in subtle ways, and *revibeability* rubrics [1] are an early answer whose alternatives remain untested. Participants design and discuss candidate approaches and consolidate them into a shared catalog.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Articulate what *revibing* means for HCI research. This focus reflects concrete challenges such as examining the research methods that *revibing* can enable, including: evaluating new systems against reimplemented baselines; and what authors should share in (and with) their papers to keep their work *revibeable* (e.g., rubrics, videos, screenshots). There are many ways that HCI generally [5, 15], and UIST specifically, creates knowledge. The workshop will provide an opportunity for participants to examine where and how *revibing* can work, given varying sub-disciplinary objectives (e.g., an interaction technique *revibe* may have very different goals than a prototyping system). The more high-level workshop question might be: how will new agentic capabilities impact our thinking about what a UIST contribution might be?

2 Workshop Schedule and Plan

Our one-day workshop consists of interactive group activities that move from building *revibes* in the morning to evaluating and reflecting on them in the afternoon.

2.1 Preparatory Activities

Prospective participants will apply through a short form with a few open-ended questions: which classic HCI system paper they would want to see *revibed*, one idea for evaluating a *revibed* system, and any 'seed' topics for discussion. Accepted participants will additionally submit a single slide for their lightning introduction. The organizers will use the paper nominations to select the walk-through paper for the live *revibing* session, and the proposed ideas and questions to seed the evaluation jam and breakout topics.

2.2 Workshop Activities

2.2.1 Welcome and Framing (9:00–9:20). Organizers open the workshop by introducing the goals of the day and the schedule. An organizer and author of the work that introduced *revibing* [1] will give a short talk on the concept and demonstrate existing *revibes*.

2.2.2 Lightning Introductions (9:20–9:40). Each participant gives a lightning introduction (30 seconds), presenting themselves and one open idea or question about *revibing* using the slide submitted before the workshop.

2.2.3 Live Revibe Demo and Workshop (9:40–11:10). We present a step-by-step walkthrough of *revibing* an existing UIST software system paper, selected from participants' survey responses to illustrate both the successes and current limitations of the approach. Participants will follow along in small groups (2–3 people). The demo begins by reimplementing the paper's system using AI coding agents. While the agents execute, participants will brainstorm ways to "remix" the classic paper with variations that elicit fresh research ideas. As a fallback for inaccessible models or slow *revibes*, we will offer *'pre-revibed'* codebase(s) on GitHub. Groups will spend the remainder of the session adding their own extensions to the system.

2.2.4 Revibes Showcase (11:10–12:00). In 3 to 5 minutes, each group will present a short walkthrough of their resulting *revibes*, any challenges and opportunities they encountered, and any remixes they made. Participants vote on each other's demos in two categories: the most faithful *revibe* and the most interesting remix.

2.2.5 Evaluation Jam (13:30–14:30). This two-part session focuses on validating *revibes*: **Ideation (30 mins):** Seeded by organizer-provided contexts and methods, participants will reflect on their morning *revibes* to brainstorm validation approaches (e.g., rubrics [1], behavioral tests, task walkthroughs, simulated users, performance user tests). Groups will also be asked to explore how evaluation changes depending on the type of artifact being *revibed* and the goals of the *revibe*, capturing their ideas on post-it notes. **Catalog Building (30 mins):** Participants will gather at a shared wall to cluster and group their post-its alongside the seeded situations. They can also comment on other groups' notes. Organizers will consolidate these insights into a shared evaluation catalog for the workshop's closing synthesis.

2.2.6 Breakout Meta Discussions & Cross-Table Exchange (14:30–16:30). This two-hour block steps back from the hands-on work to discuss *revibing*'s broader impact on the field. The session consists of two one-hour rounds. In each, participants join an organizer-moderated table for a 40-minute discussion on a given theme, followed by a 20-minute cross-table exchange where organizers summarize their group's insights (2–3 minutes each). Between rounds, participants shuffle to new tables to cross-pollinate ideas with different peers. Potential themes may include:

- **What gets *revibed*.** How does *revibeability* shift across artifact types (e.g., systems, algorithms, datasets, interaction techniques), and can *revibing* keep aging artifacts runnable?
- **Writing for *revibeability*.** What should authors share to make work *revibeable* (e.g., *revibeability* rubrics as supplements), and what does this mean for open science?
- ***Revibing* as a research method.** How can *revibing* aid research (e.g., rapid system variations, living baselines), and when is a *revibe* a valid comparison?
- ***Revibing* as pedagogy.** How does *revibing* change how we teach systems research, from reimplementation assignments to *revibe*-and-extend projects?
- **Limitations of *revibeability*.** Where do *revibes* break down, and what negative effects follow (e.g., dis-incentivizing the release of original artifacts)?
- **Future of HCI systems contributions.** If systems can be rebuilt in days, what is a systems paper's lasting contribution, and do empirical contributions gain new significance?

Themes may be added or replaced based on participants' pre-workshop survey responses. Participants will be provided with a shared document to take notes on their discussions.

2.2.7 Closing Synthesis (16:30–17:00). Finally, organizers will work with participants to consolidate the day's outputs to outline early recommendations on how to conduct and evaluate *revibes*, how papers and supplemental materials should change to foster higher-quality *revibes*, limitations, and open questions.

3 Post-Workshop Plans and Goals

We will consolidate and publish the community document drafted during meta discussions on the workshop website as a co-authored blog post. Additionally, the website will present an interactive showroom of the *revibed* artifacts and evaluation catalog from the workshop, which will also be hosted in a public GitHub repository.

References

[1] Eytan Adar, Yoonjoo Lee, Ning-Er (Nina) Lei, Q. Vera Liao, and Weirui Peng. 2026. Revibing Code from Papers: Reimplementing HCI Artifacts. https://bit.ly/4vpywWj.

[2] Florian Echtler and Maximilian Häußler. 2018. Open Source, Open Science, and the Replication Crisis in HCI. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8.

[3] Saul Greenberg and Bill Buxton. 2008. Usability Evaluation Considered Harmful (Some of the Time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 111–120.

[4] Kasper Hornbæk, Søren S Sander, Javier Andrés Bargas-Avila, and Jakob Grue Simonsen. 2014. Is Once Enough? On the Extent and Content of Replications in Human-Computer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3523–3532.

[5] Scott E. Hudson and Jennifer Mankoff. 2014. *Concepts, Values, and Methods for Technical Human–Computer Interaction Research*. Springer New York, New York, NY, 69–93. doi:10.1007/978-1-4939-0378-8_4

[6] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. In *The Twelfth International Conference on Learning Representations (ICLR)*.

[7] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–17.

[8] Dan R Olsen Jr. 2007. Evaluating User Interface Systems Research. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. 251–258.

[9] Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. 2025. Paper2Code: Automating Code Generation from Scientific Papers in Machine Learning. *arXiv preprint arXiv:2504.17192* (2025).

[10] Zachary S Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. 2024. CORE-Bench: Fostering the Credibility of Published Research Through a Computational Reproducibility Agent Benchmark. *arXiv preprint arXiv:2409.11363* (2024).

[11] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. 2025. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848* (2025).

[12] Chat Wacharamanotham, Lukas Eisenring, Steve Haroz, and Florian Echtler. 2020. Transparency of CHI Research Artifacts: Results of a Self-Reported Survey. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[13] Huanting Wang, Jingzhi Gong, Huawei Zhang, and Zheng Wang. 2025. AI Agentic Programming: A Survey of Techniques, Challenges, and Opportunities. *arXiv preprint arXiv:2508.11126* (2025).

[14] Max L Wilson, Paul Resnick, David Coyle, and Ed H Chi. 2013. RepliCHI: The Workshop. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. 3159–3162.

[15] Jacob O Wobbrock and Julie A Kientz. 2016. Research Contributions in Human-Computer Interaction. *Interactions* 23, 3 (2016), 38–44.

[16] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems*, Vol. 37. 50528–50652.