How to Study and Support Labor of a Nascent Category? Reflection on the Investigation and Intervention of Scientific Software Work

C. Fan Du[1] and James Howison[2]

To vision and build towards the future of work, driven by the drastic evolution of technologies in workplaces, does not only involve the revamping of practice and profession within existing occupational categories; the emergent categories also call for attention, incubation, and efforts towards legitimation. Scientific software work, while critically driving forward the progress of scientific research, is one of those categories that have yet to be fully established. Both the products and work of building software stay relatively invisible within the sociotechnical system of scientific work (Alter, 2013; Winter et al., 2014). Scientific software as the infrastructural support of the increasingly digital scientific research, is often seen as the byproduct of the research process, and less acknowledged as the first-class citizens in the published science. Accordingly, contributors to scientific software are not given enough credit to demonstrate their impact on science. Being researchers who squeeze time from research for building software, sometimes even anonymous doctoral students who satisfice their sleep (Howison & Herbsleb, 2011), the group of scientific software contributors are far broader than few "research software engineers" who hold a formal job title with support oftentimes from large, computing-intensive research institutions. Even many of those research software engineers mobilize globally to advocate for more recognition of this nascent job category.

One pathway to more deserved visibility and credit for scientific software work is widely believed as to build software work into the current incentive system of science. While the system of literature citation and publication serves as the mainstream tally of scientific contribution, software work struggles to find its place. Initiatives such as journals that are dedicated to the publication of software work (e.g., The Journal of Open Source Software[3]; Journal of Open Research Software[4]; ACM Transactions on Mathematical Software[5]), are not scalable enough to countervail the systematic neglect of software work in science. Meanwhile, a constellation of efforts in standards setting, policy advocacy, and leading cultural change are dedicated to the standardization and implementation of software citation (e.g., The FORCE11 Working Group on Software Citation[6] and the leading efforts taken by software repositories and registries, etc.). Though, it remains challenging to translate the standards set at the macro level into each researcher's citation practice (Katz et al., 2019).

We argue that such collective efforts from both bottom up and top down are needed for implementing systematic change in science. Moreover, for channeling system-level endeavors into effective behavioral changes, we created more subtle and directed nudges via our design intervention. We have developed a specialized, interactive search engine, named CiteAs, with the basic support of helping users hunt for the relevant information for citing software and making standardized citation recommendations. By interacting with CiteAs, researchers who intend to cite software in their papers get the needed information at one search. For software

1,2 School of Information, The University of Texas at Austin

3 https://joss.theoj.org/

4 https://openresearchsoftware.metajnl.com/

5 https://dl.acm.org/journal/toms

6 https://www.force11.org/group/software-citation-implementation-working-group

creators, CiteAs will demonstrate a thorough list of provenances where it retrieves the required metadata for constructing a citation and highlight the missing items where researchers can start from for increasing the visibility of their software work.

To unpack more about our design intention, it would be helpful to discuss what it means to make software work more visible in a technical sense. The FORCE11 Software Citation Working Group has been spearheading the standard setting efforts for software citation. They established several general principles for citing software, ensuring the findability, accessibility, and the appropriate crediting of the artifact achieved by citation (Smith et al., 2016). From the perspective of information science, to meet such principles requires the availability of the detailed information about a piece of software, including its digital identifier, link to its repository, website, or storage space, and version numbers, etc. While it is laborious to prepare the software artifact with these metadata in a standard format and share them with the public online, it is also arguable who should carry that labor on their shoulder. If we purely rely on the volition of software developers, it means adding extra workload to their already underacknowledged work, leaving alone the fact that the availability and quality of metadata would be quite uncontrollable. However, we argue that the curation and sharing of software metadata align with the benefit of scientific software contributors, as this is a key move in augmenting the visibility of their software contribution to science within the system of citation. To enable others to cite your software work, human- and machine-readable metadata is the best technical precondition at present.

In a series of studies conducted by Howison & Herbsleb (2011; 2013; Howison et al., 2014), by in-depth interviews and participant observation in relevant workshops, an overview of the scientific software ecosystem has been mapped out: key stakeholders in different ecosystem positions, their specific information needs, the distant interactions between stakeholder roles, and the status of software work across the ecosystem. From an ecosystem-level view, there exists a remote information exchange between software end-users, especially those who are expected to cite software and thus increase their visibility, and software producers, who are encouraged to share all the information needed by end-users for citing and accessing their products. Such exchanges are beneficial to both sides but there exists a gap in between. Previous studies identified that many software producers post their citation request either online or embed it in their software packages. But these citation requests are often out of sight of end-users who have the potential to cite them. One crucial design intention of CiteAs is to link the citation requests produced by these software developers and the software end-users who are likely to cite their software products. Thus, the available information flows to the envisioned direction.

Another design intention of CiteAs is to build a digital nudging environment (Schneider et al., 2018) for guiding the choices of software developers. To educate software developers to change their metadata management practice is costly and takes a long time to achieve the wanted effects. Meantime CiteAs presents a choice environment on its interface and influences user decisions at the moment: A chain of citation metadata provenances will be displayed and the missing items will stand out. In this way software producers will be prompted what exact metadata item they can improve to make their software more visible.

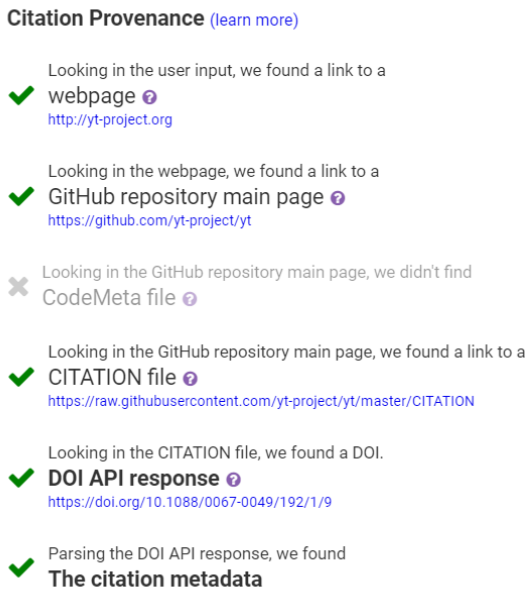

Figure 1: The demonstration of citation provenance as a nudging environment

At the core of the design deliberation of CiteAs lies the principles of mechanism design. That is, to reorganize and coordinate actions and activities by the goal-oriented design of incentive mechanisms (Hurwicz & Reiter, 2006). It is a field in economics with the basic assumption that human actors behave rationally. In our case, when we assume that most of the scientific workers are motivated by self-interest and the increased visibility and demonstrated impact of their work, rationally their decision for the software citation metadata practice would be influenced by the visual cues presented by CiteAs. Broadly speaking, by pushing forward the implementation of software citation, the incentive system of science would ultimately integrate software work; the status of scientific software work would then be established and improved.

Overall, taking a sociotechnical system view in design envisioning enables us to capture the gap in the interaction between different ecosystem roles and positions and further identify the entry point of technology intervention. In our conception, CiteAs serves as a two-sided system that bridges the gap in information flows and creates a behavioral chain with real impact within the ecosystem. We are now working on building more interactive and intelligent features into CiteAs, including a software recognizer that deploys text mining capabilities to detect informal mentions of software in academic writing. Then it is envisaged that cues would be given to CiteAs users prompting them to formalize the mention into standard software citations.

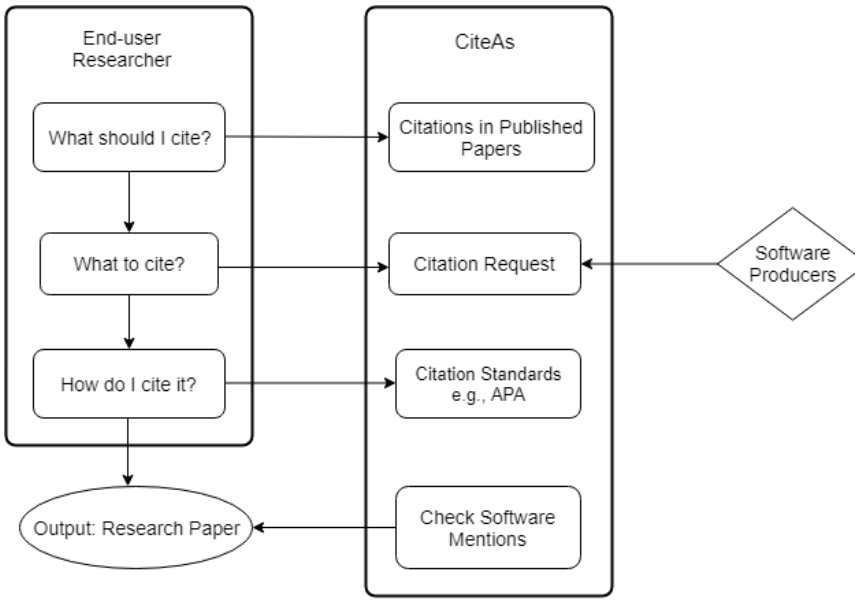

Figure 2: CiteAs is designed for bridging the gaps in software citation

The movement of software citation seeks to implement change in the existing scientific incentive system. The aim of developing CiteAs is to translate the vision of system-level change into the action of scientific workers that supports and benefits their work and status. To achieve effective intervention, we have been interviewing our targeted users and inquiring into the point and moment when end-user researchers decide to cite or not cite a piece of software in their work, or to which extent scientific software producers have adopted the recommended metadata practice. One challenge that we have encountered, is that software end-users and producers are not well-established categories with clear-cut boundaries, and their practice varies widely. The potential factors that affect their citation behavior, hypothetically their academic discipline, the professional affiliation, occupational role, training experience, and so forth, are multifarious and thus the systematic behavioral patterns are elusive. This reality makes the aggregate effect of individual behaviors and intervention hard to grasp. Another challenge we are facing is how to make our design interventions sustainable. We speculate that a sustainable intervention with the sociotechnical system vision needs to be achieved by the integration into ecosystem infrastructure as a functional component. In a nutshell, to investigate an emerging category of workers is more exploratory rather than systematic in nature; and our technology intervention aims at introducing behavioral change in line with the incentives for scientific work and the ultimate goal of repurposing the scientific reputation system.

Reference

Alter, S. (2013). Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future. *Journal of the Association for Information Systems*, 72–121.

Howison, J., & Herbsleb, J. D. (2011). Scientific Software Production: Incentives and Collaboration. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work* (pp. 513-522).

Howison, J., Deelman, E., McLennan, M. J., Ferreira da Silva, R., & Herbsleb, J. D. (2015). Understanding the Scientific Software Ecosystem and Its Impact: Current and Future Measures. *Research Evaluation*, *24*(4), 454–470. https://doi.org/10.1093/reseval/rvv014

Howison, J., & Herbsleb, J. D. (2013). Incentives and Integration in Scientific Software Production. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work - CSCW '13*, 459. https://doi.org/10.1145/2441776.2441828

Hurwicz, L., & Reiter, S. (2006). *Designing Economic Mechanisms*. Cambridge University Press.

Katz, D. S., Bouquin, D., Hong, N. P. C., Hausman, J., Jones, C., Chivvis, D., Clark, T., Crosas, M., Druskat, S., Fenner, M., Gillespie, T., Gonzalez-Beltran, A., Gruenpeter, M., Habermann, T., Haines, R., Harrison, M., Henneken, E., Hwang, L., Jones, M. B., … Zhang, Q. (2019). Software Citation Implementation Challenges. *ArXiv:1905.08674 [Cs]*. http://arxiv.org/abs/1905.08674

Smith, A. M., Katz, D. S., & Niemeyer, K. E. (2016). Software Citation Principles. *PeerJ Computer Science*, *2*, e86. https://doi.org/10.7717/peerj-cs.86

Schneider, C., Weinmann, M., & vom Brocke, J. (2018). Digital Nudging: Guiding Online User Choices Through Interface Design. *Communications of the ACM*, *61*(7), 67–73. https://doi.org/10.1145/3213765

Winter, S., Berente, N., Howison, J., & Butler, B. (2014). Beyond the Organizational 'Container': Conceptualizing 21st Century Sociotechnical Work. *Information and Organization*, *24*(4), 250–269. https://doi.org/10.1016/j.infoandorg.2014.10.003