# Beyond Open Source: A Call to Rethink Policies and Incentives for Sustainable Scientific Software Development

John Wu, LBNL, KWu@lbl.gov

Much of the high-performance computing (HPC) software ecosystem is built on top of free open-source tools (1). For example, many of the HPC systems use variants of Linux as their operating systems, Lustre as their file systems, GNU C compiler suites as their compilers, and LAPACK as their core math libraries. Aside from these foundational tools with wide usage outside of HPC systems, there are many more specialized tools critical for DOE scientific researches. These tools include high-performance simulation tools for physics, chemistry, biology, materials science, as well as data analysis tools for processing images from astronomical observations, x-ray light sources, neutron sources, and so on. These software tools are critically important for recent scientific discoveries including gravitational waves (2) and Higgs boson (3). These tools also rely on a large number other tools for their operations. A recent accounting of Spack packages (4) from the Exascale Computing Project (ECP) indicates that a handful of ECP simulation tools depend on thousands of other packages, most of which are open-source. The success of DOE science rely on these software packages. Continuing the development and maintenance of these tools require a thriving community of tools and developers. Open Source as a development paradigm has served us well so far. However, the developers of these Open Source Software usually could not depend on their software to make a living and therefore have to find ways to support themselves, which creates tension in the development community. This position paper is a call for more flexibility in valuing scientific software, which we believe would lead to more ways to support scientific software development.

## Software Sustainability: State of Art

Innovation in scientific software is highly valued, in contrast, the maintenance efforts are not receiving nearly as much attention. However, both development and maintenance efforts are critical to a thriving science community. In the above mentioned example from ECP, where ECP funds the work of a couple of dozen of software products, however, these software products relies on thousands of supporting packages. As these packages are used on new HPC systems and in new application scenarios, these supporting packages would require updates, corrections, and modifications. The key software sustainability challenge is "how to support these maintenance efforts required for these thousands of supporting packages?"

Software Sustainability is a long-standing research topic (5). Much of the published literature focuses on how to architecture the software system and similar technical issues. At the same time, there are also a number of studies pointing to the importance of non-software issues such as license and citation. For example, Midha and colleagues found that license choice affects both technical successes as well as market successes of an open source software project (6), and a number of authors have written about the importance of citation and credit of software tools (5). Another broad categories of issues sometimes discussed is community building (7). However, we believe the following observation from the Software Sustainability Institute really captures the core issue, "reward" (8):

> Many researchers are yet to be convinced of the importance of developing well-engineered software. Although we might disagree with this viewpoint, it's an understandable one, because the research community provides little - if any - reward for producing such software. A lack of reward leads researchers to choose quick fixes over a more considered, maintainable approach to development.

Therefore, our main thesis is to propose more research into different "reward" mechanisms.

## Exchange Platform and Evaluation of "Reward" Mechanisms

The existing software sustainability research touches on a number of aspects related to rewards. For example, license could be used to gain monetary rewards and citations is a form of reputation reward. Given that much of the scientific research are supported by grants with a limited ability to pay for software tools, there is strong bias preferring free open source license with no virial restrictions. However, this is not always the best option, even for esoteric software packages specifically designed for molecular dynamic simulations. There are many open source versions of molecular dynamics simulation packages (9), but also a number of commercial packages that perform the same tasks (10). In fact, we are aware of a senior scientist who recently resigned from a large open source development effort to start a commercial venture to market a set of molecular dynamics software. We take this as an evidence of the tension between divergent choices available at the moment. We don't believe the currently available choices are the only ones. Could we devise additional choices that might better serve the scientific community?

Based on the theoretical framework developed by Morgan et al (11), we propose to explore different reward mechanisms by conducting large-scale simulations. The simulator will track the operations of a large software trading/exchange platform with the ability to employing different utility assignment algorithm for software packages (12). The ideal outcome would be to develop a reward mechanism to enable the software community to self-sustaining. Most likely we might identify a number of mechanisms that might be able to achieve this ultimate goal, it would then be necessary to design a real software exchange to share the software development expertise and the resulting software tools. As with those platforms for the sharing-economy, we anticipate this exchange mechanism to help us further explore the reward mechanism for a self-sustaining scientific software community.

## References Cited

1. Bangerth W, Heister T. What makes computational open source software libraries successful? Computational Science & Discovery. 2013;6(1):015010.
2. Barish BC, Weiss R. LIGO and the detection of gravitational waves. Physics Today. 1999;52:44-50.
3. Spira M, Djouadi A, Graudenz D, Zerwas R. Higgs boson production at the LHC. Nuclear Physics B. 1995;453(1-2):17-82.
4. Gamblin T, LeGendre M, Collette MR, Lee GL, Moody A, Supinski BRd, et al. The Spack package manager: bringing order to HPC software chaos. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis; Austin, Texas: Association for Computing Machinery; 2015. p. Article 40.
5. Katz DS, Choi S-CT, Lapp H, Maheshwari K, Löffler F, Turk M, et al. Summary of the first workshop on sustainable software for science: Practice and experiences (WSSSPE1). arXiv preprint arXiv:14047414. 2014.
6. Midha V, Palvia P. Factors affecting the success of Open Source Software. Journal of Systems and Software. 2012;85(4):895-905.
7. Boettiger C, Chamberlain S, Hart E, Ram K. Building software, building community: lessons from the rOpenSci project. Journal of open research software. 2015;3(1).
8. Crouch S, Hong NC, Hettrick S, Jackson M, Pawlik A, Sufi S, et al. The Software Sustainability Institute: Changing Research Software Attitudes and Practices. Computing in Science & Engineering. 2013;15(6):74-80.
9. Jing X, Troullier NR, Dean D, Binggeli N, Chelikowsky JR, Wu K, et al. Ab Initio molecular dynamics simulations of Si clusters using the higher-order finite-difference-pseudopotential method. Physical Review B. 1994;50(16):12234-7.
10. Hafner J. Ab-initio simulations of materials using VASP: Density-functional theory and beyond. Journal of Computational Chemistry. 2008;29(13):2044-78.
11. Morgan L, Feller J, Finnegan P. Exploring value networks: theorising the creation and capture of value with open source software. European Journal of Information Systems. 2013;22(5):569-88.
12. Chli M, Wilde PD. The Emergence of Knowledge Exchange: An Agent-Based Model of a Software Market. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 2008;38(5):1056-67.