STATUS OF SOFTWARE PRESERVATION IN PLANETARY SCIENCE. C. C. Million¹, A. Brazier², T. King³, A. Hayes³. ¹Million Concepts (PO Box 119, 141 Mary St, Lemont, PA 16851; chase.million@gmail.com), ²Cornell University, ³UCLA.

Introduction: At the 2015 Workshop, we discussed the need for data users and software producers to discuss and agree on standards and policies related to executable digital data archiving in the absence of formal policies from our funding agencies. In the intervening two years, we have noted increasing community understanding of and support for software preservation, as well as evolving policies from NASA and related institutions on this topic. We will review the current status of software archiving in planetary science and suggest next steps.

NASA Planetary Science Division policy: The 2016 and 2017 NASA Research Opportunities in Space and Earth Sciences (ROSES) omnibus solicitations included new and updated requirements with respect to Data Management Plans (DMP) under many programs and subprograms. Of particular interest, the Planetary Science software is explicitly included as a class of data that may need to be archived. Specifically: “The DMP should also cover any other data and software that would enable future research or the replication/reproduction of published results. Software, [...] should be made publicly available when it is practical and feasible to do so and when there is scientific utility in doing so. [...] NASA expects that the source code, with associated documentation sufficient to enable the code’s use, will be made publicly available via GitHub (https://github.com/NASA-Planetary-Science), the PDS (for mission-specific code, when appropriate), or an appropriate community-recognized depository (for instance, the homepage of the code base for which a module was developed). Archiving software [...] does not require the proposer to maintain the code.” [1]

Additionally, the 2017 Planetary Data Archiving, Restoration and Tools (PDART) solicitation, which supports software tools development, includes (as it has in the past) a requirement that data be deposited “in the NASA PDS or an equivalent archive.” For the first time, the 2017 call clearly defines what qualifies as a “PDS equivalent” archive, specifically those which have features of (1) independence, (2) sustainability, (3) open accessibility, (4) searchability, (5) citability, (6) preeminence, and (7) standardization, with desirable but optional features of (8) peer review and (9) documentation. These terms are briefly defined in the solicitation. [2]

This policy is a significant step toward research reproducibility, but some notable issues remain. First, as we noted in 2015, Github is not an archive; it is a privately owned and controlled software project management tool that lacks the infrastructure, oversight, and Federal backing that are necessary to ensure the survival of data and does not meet the definition of “PDS equivalence.”

Also, source code and documentation (no matter how thorough) is unlikely to be sufficient to preserve long-term software usability, which in many cases will require preserving the operating environment in which the software system was designed and used. All software becomes un-runnable over time through changes to the software and hardware operating environment in a process called “bit rot,” eventually resulting in complete obsolescence such that any functionality is lost. To maintain the usability of software for decades or more requires “freezing” the operating environment as a virtual machine image. Neither current NASA policy nor archiving practice allows for this approach.

Planetary Data System policy: In 2016, the Planetary Data System Management Council issued a “Policy on Software Archiving.” It states, “Source code that is sufficiently documented may be submitted as documentation for or an example of some processing algorithm. It will be subject to the same review and standards requirements as any other document submitted for archiving.” and “PDS will not support or maintain software in the archive.” [3] This clarifies prior PDS policy in establishing that plain text source code is to be treated as a sort of non-prose documentation and seems to, by exception, reiterate that “executable” data such as virtual machines and source code that does not serve to document “some algorithm” (e.g. test cases, installation scripts, Docker files) are unwelcome.

Future work: Archiving of software is complicated by the realities of current technology, software engineering practices in research environments, and software licensing. Best practice continues to evolve rapidly. However current policies are inconsistent and likely to fail, in practice, to preserve the functioning of software over long terms. We will discuss a range of possible paths towards long term research software archiving in usable formats. Such paths may include compromises on what qualifies as an appropriate archive for research software and pursuit of advances in the state of the art of technology and archive practice.