A Planetary Software Reference Database

PI: Chase Million[a], Co-I: Michael St. Clair[a], Collaborator: Jennifer Seiler[b]
[a]Million Concepts, [b]Columbia University

Table of Contents **Page**

Overview

We propose to build, populate, maintain and promote a Planetary Software Reference Database (PSRD): a directory of software tools and source codes which are of specific use in planetary science research. The PSRD will serve two primary functions. First, (1) it will act as a centralized reference for planetary science-related software, with an emphasis on searchability and discoverability, dramatically reducing the difficulty of finding extant software and tools relevant to a specific research topic. This will reduce research costs by avoiding unnecessary duplication of software development labor and advance the state of planetary science by allowing investigators to more readily build upon the work of their colleagues. Second, (2) it will simplify the citation of planetary science software and source code, particularly in cases where there is no associated "software paper." We anticipate that this will increase the perceived legitimacy of software as a research product within the planetary science community and incentivize code creators to share their work. This will both bootstrap the salutary effects of (1) and generally increase the level of transparency and reproducibility within planetary science.

The creation and use of software is a critical component of the workflow of modern scientists in general and planetary researchers in particular. Almost all planetary scientists make use of the same pool of digital data from planetary science missions (as provided by, e.g., the Planetary Data System archive [1]). They therefore must interact with their data "in software" to some extent, like researchers in most modern scientific disciplines -- but unlike researchers in most modern scientific disciplines, they largely share the same data. This makes the value of generalized and shared software tools in planetary science even greater than in many other disciplines. And planetary scientists do frequently make use of generalized software tools developed for the community for basic search and retrieval, processing and visualization tasks (e.g., ISIS [2] or JMARS [3]) -- but they also develop a tremendous amount of custom software tailored to their specific research objectives. In almost all cases, this software is developed by and for the exclusive use of an individual researcher and his or her immediate collaborators and is never released to the community at large. This failure to release software as research products is potentially a source of unnecessary duplication of effort (wasted researcher time and resources). Also, because so much modern science happens "in code," the ubiquity of highly opaque, custom software makes it difficult for planetary science researchers to independently verify or build upon the results of their colleagues' research. In other words, although transparency about raw data inputs and outputs is a community standard, the community generally considers it acceptable to leave many intermediate steps as "black boxes."

The problem of duplication of effort among software developers within the planetary science community is both well-acknowledged and unaddressed. One of the summary recommendations to come out of the 2012 Planetary Data Workshop at the USGS, which brought together 150 data users and developers (for the first time in 29 years), was as follows: "Software developers acknowledged that there are many facilities addressing and solving similar problems in data access, delivery services and analysis methods. Using existing libraries and standards and sharing source code when possible would help to alleviate much of the overlap." [4]

A centralized, domain specific database of software, source code, and tools is the most plausible and efficient way to minimize duplicate developmental effort.

Background

Recent Discussions in Data Management and Reproducible Research

These problems are certainly not unique to planetary science. They have been the focus of intense discussion among many scientific communities in the past decade, often under the subject headings of “data management” and “reproducible research.” They have been the topics of entire conferences (such as “Working towards Sustainable Software for Science: Practice and Experiences” or the WSSSPE), papers (cited throughout this proposal), and even policy decisions (like the requirement that all NSF proposals submitted after January, 18, 2011 include an explicit data management plan). With the proposed PSRD project, the intention of the Investigators is not to contribute substantially to this discussion, but to draw upon it.

The results of a survey conducted by Stodden (2010) [5] compared the reasons given by computational scientists for sharing and not sharing research source code. The top reasons given for not sharing were (78%) “the time it takes to clean up and document for release,” (52%) “dealing with questions from users about the code,” and (45%) “the possibility that your code may be used without citation.” The top reasons reported by Stodden in favor of sharing were (91%) “encouraging scientific advancement,” (90%) “encouraging sharing and having others share with you,” (87%) “being a good community member,” and (84%) “improvement in the caliber of research.” Hong, et al. (2013) [6] lists reasons that researchers might want to publish papers covering their scientific software (i.e., “software papers”) “as a record of a particular research object: to advertise the work that has been done; to allow scrutiny of your work; so that other[s] can reproduce your methods; to enable reuse amongst other[s] in your research community; to build on your work to look at new kinds of studies; to allow its reuse for other purposes such as teaching, journalism and citizen science; to describe your software such that it can be preserved; to enable recognition and reward of your work.” We suggest that these points serve equally well as reasons that a researcher might want to share any code, even if it is not extensive or innovative enough to support a journal article -- provided of course that the contribution to the community can be referenced in the same manner as a paper.

Discoverability and Citability

A recurring theme in the reasons given both for and against sharing of software is that discoverability and citability are important. While there is considerable disagreement across the scientific community about the weight that should be given to software tools and source code as research products, there is essentially unanimous agreement that software tools and source code should be published and tools shared. [7-9, and many others] Stodden, et al. (2013) [10] assert that all 3rd party software used in published research should be cited, and that good software citation practice must includes a unique identifier. While they also suggest a specific citation format, we note that appropriate citation formats will vary by the context of the presentation of the work. The crucial factor is that the minimum information which forms the citation itself must be readily available: author(s), project title, publisher or affiliation, date, and a unique identifier. The PSRD will store these core elements (or as many as are available) and other metadata for each software entry, and feature tools to produce well-formatted, maximally informative (given available metadata) citations that match a number of common standards such as MLA, APA, and Chicago, or the file formats of popular citation management software. (Many online reference databases, from Google Scholar [11] to JSTOR [12], implement a similar citation scheme.)

To be clear, there are serious, ongoing, and unlikely to be soon-resolved discussions about what best practices for code citation should entail. How do you rank author contributions across large projects that include dozens or hundreds of individual developers, for example? While technically more feasible to track all individual contributions to a software project (unlike many traditional research projects), how should one weight the relative merits of a major feature implementation against a tricky bug fix against technically straightforward but tedious and critical writing of documentation? How should software libraries like Numpy or Pandas be cited when small fractions of the capabilities are ultimately used in any given project? (Knepley, et al., 2013 [13] suggests that sub-portions of software libraries should be cited individually.)

These problems are present in virtually any citation scheme -- they even exist in citation practices for traditional print references, as in the case of materials published by large consortiums. We cannot and are not attempting to resolve these issues. The goal of PSRD is simply to allow researchers to find software of potential utility to their own work, cite software clearly and uniquely, and allow their own software to be cited and found.

A Brief Overview of Scientific Software Databases

It is important to note that software tool reference databases such as PSRD are not a new idea. The concept of "a directory of links to resources" predates the modern internet. Most scientific fields have at least a few such software directories, generally of limited scope and with little claim to centralization. Some relevant examples include the NOAA's list of Other (Non-NWS) Software Links [14] and the Geotechnical & Geoenvironmental Software Directory (GGSD), [15] which has been around since 1996 and reports 10 million page hits per year.

The history of such efforts is long and dense, and it is outside the scope of this proposal to cover all directories of scientific software in detail. We're aware of very few examples of software tools or code directories in planetary sciences, however. The Mapping, Remote-sensing, Cartography, Technology, and Research (MRCTR) GIS Lab at the USGS Astrogeology Science Center also maintains a list of GIS tools and viewers [16]. The only attempt at creating a generalize directory of such software and tools within planetary science appears to be the Tool Registry maintained by the International Planetary Data Alliance (IPDA) [17] which has narrower scope than the PSRD. The IPDA Tools Registry limits eligible entries to tools developed and maintained by IPDA partner agencies, relies on user submissions rather than active curation, and does not include search or citation features.

It may be worth examining the history of broad-scope software directories in the closely related field of astronomy as a possible predictor of what we can expect from an effort to do the same in planetary science. In a preprint submission to the upcoming 2nd Workshop on Sustainable Software for Science (Nov. 2014), the founders of the Astrophysical Source Code Library outlined their general approach to improving software discoverability in astronomy, their background research, and their strategy for future development [18]. They identify at least seven attempts to create centralized directories of astronomy-relevant software tools: (1) the Astronomical Software Directory Service (ASDS) [19], (2) Astroforge [20], (3) SkySoft [21], (4) the Astro-Code Wiki [22], (5) Astro-Sim [23], (6) AstroShare [24], and finally (7) the

Astrophysical Source Code Library (ASCL) itself, which is a 2010 revitalization of a directory which went dormant in 2002 [25]. The ASCL editors report that these previous attempts failed primarily for either lack of visibility and buy-in by the targeted community, unwillingness or lack of commitment by authors to make their software available, or termination of funding. In their current effort, they attempt to combat these failure points through aggressive marketing of their resource to the community, incentivizing code sharing (by improving citability), and exercising patience, or a long view to the cultural change that may be required in astronomy for software citation to become common practice.

We think that ASCL is doing a lot of things correctly, and we intend to build upon the lessons they've learned. One may ask the question of why bother to build a new software directory for planetary science at all, the PSRD, rather than use an existing option like the ASCL. The frank answer is that we think we can both improve on the model and adapt it to more specifically and better serve the needs of the planetary science community. ASCL does not, at present, have any complex search features, and while all entries in the ASCL are citable, the repository uses a custom unique identifier called the *ASCLID* rather than the industry-standard DOI described in more detail below. Because the ASCL has an emphasis on astrophysical applications, which are not the same as planetary remote sensing applications, many ASCL entries include astrophysical simulations and astronometric calibration or visualizations which are not relevant to planetary science and might dilute the utility of the directory for planetary researchers. And the ASCL does not, as part of their mission, promote the discoverability of either *commercial* software packages or incomplete code snippets and scripts (as opposed to fully formed software packages), both of which are often key and valuable pieces of software-enabled investigations by scientists.

There already exist a variety of web-based services widely used by planetary scientists and others that serve as stable software hosting solutions and can automatically assign DOIs to hosted software resources. GitHub [26] is the best-known general-purpose service of this type, and is broadly used in nearly every branch of software development, including computational science. Several services specifically focused on scientific software products and data outputs also exist -- notably, ResearchCompendia [27] and Figshare [28].

These are fundamentally *repositories* rather than *directories*. They are extremely valuable, but address different concerns. Generally speaking, they are aimed at the needs of software developers rather than end users. They aid collaboration, archival, targeted distribution, and even, in the case of ResearchCompendia, research reproduction and immediate successor projects. However, they do not attempt to address end users' discovery costs; in other words, they do not make it easier for planetary scientists to find and use already-developed software products that are immediately relevant to their current or potential research projects.

Directions Forward for Software Databases in Planetary Science

Planetary science needs a unique software directory targeted to the needs of the field for the same reason that it needs field-specific data archives, to directly and specifically meet the needs of that research community. **The PSRD's potential impact on NASA would arise from its novelty and utility as a citable, searchable, directory of planetary science-focused software.** PSRD will, by its existence, promote planetary science community norms and standards for the publication of software products, legitimizing them as valuable--and perhaps in some cases even primary--research products. Hopefully, integrating software products into existing institutionally-recognized norms and metrics about research, citation, and impact will further encourage the publication of software products. This will create a virtuous cycle in which investigators' increased willingness to publish and describe their scientific software makes software discovery, research reproducibility, and computational planetary science in general easier and more efficient.

Technical Approach

Scope of the Project

We propose to create a directory of software tools and source code of specific interest to researchers in planetary science. We will design, develop, and build a web portal which allows (1) users to submit content or content updates, (2) curators (i.e., the Investigators, at least initially) to review and approve eligible content and content updates, and (3) users to search, retrieve and *cite* the content based upon metadata parameters.

The PSRD will serve principally as a directory *aimed at the software discovery needs of planetary scientists*. The PSRD will maintain a variety of data necessary for scientists to discovery and cite the software it indexes with maximum efficiency. It will also focus on *planetary science research-focused searchability*. It will be a *well-described*, *curated* database. Descriptions of linked software products will include the software's history, known use cases, available documentation, target and output data types, and so on. To the extent possible, these will also be described by a wide number of searchable tags which will be generated by curators, end users, or the portal software itself. In the service of being a targeted and curated resource, not any and all submissions or software will be accepted for inclusion, but the criteria will be broad. If the software or source code meets these criteria for inclusion (in our opinion), then it will be eligible for inclusion in the PSRD.

Software will be eligible for inclusion in the Planetary Software Reference Database if it

1. **addresses domain-specific needs of planetary science researchers and**
2. **is available through the internet for use by every member of the community.**

While eligibility is predicated on availability, it does not depend on the license of the available software / code or the cost structure under which it can be obtained, though it will be possible to filter results based upon such parameters. Similarly, while, as a NASA funded project, our primary objective would necessarily be to support NASA mission derived science, our assessment of eligibility for the inclusion of software will not be restricted to only software which supports NASA mission data. Agency-funded planetary science researchers frequently collaborate with non-US colleagues and make use of data from non-US missions. The two major categories of computational science software which will *not* be eligible are (1) tools that are broadly useful to computational science but in no way planetary science-specific (like text editors and core programming language modules such as Numpy [31]) and (2) tools which are not available to all members of the community (such as software developed for the exclusive use of specific research groups or mission / instrument teams, or software restricted by ITAR).

Specific examples of code which would be eligible for inclusion in the PSRD include

- Tools or reading and displaying planetary spacecraft data, e.g. JMARS, ds9 [29]
- Tools for the calibration of planetary spacecraft data, e.g. ISIS
- Modeling software, e.g. GCM or radiative transfer models
- Useful scripts or code snippets, e.g. a script to copy a subset of PDS data to a local disk based on user-defined parameters
- Language-specific software, e.g. IDL, Matlab or Python tools, functions and scripts
- Geospatial Information System (GIS) tools, e.g. GDAL [30]

Project Database and Interface Capabilities

Software will be tagged and searchable by the following parameters, when applicable and available: broad class of functionality or area of utility (e.g. data visualization), narrower sub-domain (e.g. 3d visualization), license type, fee structure if any, availability of source code for review or reuse, language or languages of the source code, supported operating systems, and specific classes of functionality such as individual missions, or instruments. Based on use patterns and available metadata, we may, of course, modify or add to these search terms over the course of the project. Potential additional search terms could include a rough metric of the level of documentation available for a specific project (none, some, a lot), or the perceived (to the curators) maturity of the project (and it's likelihood of, e.g., maintaining backward compatibility). We intend to include a number of *non-searchable* metadata fields when the information is available such as a list of individuals on the software team, the prefered citation format of the software team, a lists of other research products which either describe or utilize the software entry, and aggregate citation or usage statistics. Basic user reviews (one of five stars, etc.) or discussion board functionality may seem like obvious additions to this system, but we are explicitly not including any feature for community feedback for a number of reasons: (1) it risks being simply a bad implementation of inauditable "software peer review" at a time when standards for this practice are still being explored, (2) it is our intention to create a directory / database resource, not a social / community forum, (3) we expect software citation rates, once those citations are trackable, to speak for themselves.

To use a well known example, ISIS3, the most recent version of the Integrated Software for Spectrometers and Imagers (ISIS) created and maintained by the USGS Astrogeology Science Center, would be broadly classified under the head of data processing, with sub-classifications in geometric and radiometric calibration, released under a permissive license (nearly public domain), and available for free. The source code is available for review and reuse, largely written in C++, and designed primarily for *nix environments. ISIS supports data from a wide variety of missions including Lunar Orbiter, Apollo, Voyager, Mariner 10, Viking, Galileo, Magellan, Clementine, Mars Global Surveyor, Cassini, Mars Odyssey, Mars Reconnaissance Orbiter, MESSENGER, Lunar Reconnaissance Orbiter, Chandrayaan, Dawn, and Kaguya. We consider ISIS3 to be well documented and at relatively high maturity with little chance of breaking backward compatibility (though future versions, like ISIS4, might possibly do so). The ISIS3 documentation includes a large list of contributors who would be included in the PSRD reference, and the team prefers to receive credit through the following acknowledgement: "This research has made use of the USGS Integrated Software for Imagers and Spectrometers

(ISIS).” They also have a large list of references containing descriptions of their tools--a list which would be included in the PSRD--, and a huge number of planetary science results have been assisted by the ISIS software (some subset of which could well be included). [32] It is worth noting that the ISIS team has done an unusually thorough job of documentation and that many of these parameters / meta-data fields will not be defined for most projects referenced by the PSRD.

The question of how to host software and source code in a continuously maintainable and citable way--sometimes referred to as the problem of “software sustainability”--is under active discussion across all scientific disciplines right now and doesn’t show signs of being resolved very soon. We will avoid addressing many of these issues by not hosting code directly but only linking to projects hosted elsewhere. We will address the issue of “link rot” where links fail to be valid over time by implementing automated scripts which check link validity and for updates or changes to projects. These scripts will alert the curators to the existence of changes which may necessitate updates or revisions to the database.

Another essential tool for addressing “link rot” is the use of Digital Object Identifiers (DOIs). DOIs are unique codes assigned to electronic resources and maintained by the International DOI Foundation the registration authority for the ISO standard [33] for the DOI system. The role of a DOI in helping users and curators find and maintain access to electronic resources is analogous to the function that ISBN and ISSN codes have for readers and curators of physical books. Indexes of DOIs contain metadata about their associated objects, including current URLs. However, because they rely on stable codes with associated metadata, DOIs do not “rot” like URLs or similar static location markers. (If an electronic resource stops being available anywhere online, of course, DOIs won’t make it available, any more than an ISBN can make an out-of-print book available. They are still much more robust solutions than simple URLs, publisher catalog codes, etc.) We will assign DOIs to projects which do not already have them. We will also automatically generate properly formatted citation text (for dropping into proposals, posters, and publications) for software in the database. We will store a reference to the DOI for projects when it already exists and, when it does not, we will generate and register one. **A researcher will be able to write a useful code snippet or script, put it on a website, register it with the PSRD and it will immediately become not only a discoverable resource for the community, but a citable research product.**

<u>Community Outreach Aspects</u>

As past attempts have shown, awareness and adoption are key to the success of tools like the PSRD. We’re going to combat death by obscurity with a plan to publicize the work at several major planetary science conferences over the course of the project, periodic submissions to industry-wide publications (such as Planetary Exploration Newsletter [34]), flyers and handouts at conferences, and journal articles. We will be able to track the impact of the PSRD directly through web page view statistics and community software submissions and indirectly through citation rates of software (via DOI) in our system. Note that we expect adoption of our project to be roughly geometric--use will drive more use.

The portal will include a mechanism by which members of community can submit software and code (their own or others’) for inclusion in the database. Such submissions will be vetted by the investigators before inclusion. The investigators will also actively seek out new entries through web searches, interaction with the community, and ongoing reviews of conference proceedings and professional publications.

We will also practice what we preach and make the source code for the web site, script, and database management components as well as the database contents themselves freely available to the community under a permissive license from the beginning of the project. (The PSRD software itself will be the very first entry in the PSRD.)

Work Plan

Team

PI Million will perform all project management functions including the responsibility to ensure that project goals are met in a timely fashion and to high standards. He will guide and contribute to all stages of the project, perform development / programming functions when necessary, and serve as a researcher and curator of PSRD content. The PI will be committed for 0.1 FTE (208 hours) in the first year, during major development, and 0.05 FTE (104 hours) in the second and third years of performance.

Co-I St. Clair will help develop and maintain the PSRD design requirements--to include both the interface and database--and serve as the primary researcher and curator of PSRD content. St. Clair will be committed for 0.1 FTE (208 hours) in all three years of performance.

Collaborator Seiler, in addition to serving as the point of contact with ResearchCompendia, on which project she is a cofounder, for the archival of our project (see below), will serve as an occasional advisor on matters related to software based research sustainability, reproducibility, and citability.

Timeline

Following software project management best practices, in the first three months of the period of work, the Investigators will develop a number of project description and design documents. The first of these will be a Concept of Operations (ConOps) design document broadly following the outline of IEEE Standard [35]. A ConOps is a document which describes the characteristics of a proposed system from the point of view of the users of that system. In this case, the users would include both members of the public or community as well as the curators (i.e. Investigators) themselves. In addition to capturing much of the information already described in this proposal with respect to things like parameters for eligibility for inclusion of entries, the classes of metadata by which entries can be searched, and strategies for combating link rot and promoting citability, the ConOps will also include items such as interface mockups, discussion of user experience (UX) considerations, technical circumstances in which it will be deployed, and any possible performance or quality metrics which should be optimized against (e.g. Whether speed is a higher priority than aesthetics).

Using the ConOps as a guide, we will then develop a Software Requirements Specification (SRS) document. The SRS is a technical description of the software system to be developed, laying out the functional and non-functional requirements. Whereas the Conops is a user-oriented document, the SRS is a developer-oriented document which will serve as the basis to hire a professional web developer and graphic designer to produce the core PSRD site, functionality, and underlying database management system as we’ve described them.

Website development, validation, and testing is scheduled for the fourth to ninth months of the period of performance. The bulk of major site functionality and design will be performed professional web developers to ensure that it is robust, reliable, extensible, usable, and built to industry standards. At the same time as major web development is taking place, the Investigators will be creating and integrating the necessary tools, scripts, and utilities which go beyond basic web and database design, including mechanisms for correctly formatting metadata into user-ready citations, for automatically assigning DOIs when needed, for tracking changes to linked web pages, and for tracking citation rates of curated codes. All development will be performed openly, with the code hosted publicly on a site such as Github under a permissive license.

The following two and a half years activity will be composed of maintenance and growth activities. St. Clair will populate the web database from user submissions and, drawing upon his background as an academic researcher in a wide range of technological domains, search for eligible resources in industry publications, conference proceedings, and targeted web exploration. During this time, the PI will advertise the work through conference presentations and flyers, announcement submissions to field specific publications, guest blogging, and journal articles. Anticipating the unforeseeable in terms of bugs and must-have features or interface changes, we've also allocated a small amount of money to for a professional web developer and graphic designer in these later years. Both investigators are also qualified to perform software modifications and bug fixes, and will do so as needed.

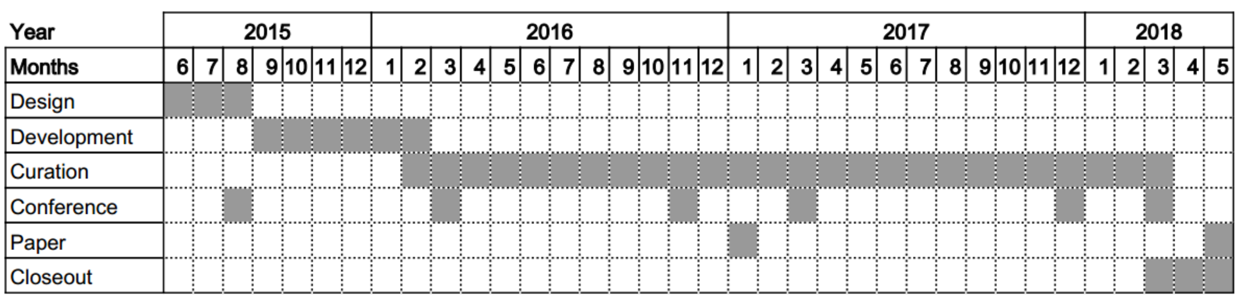

Figure 1. Task matrix over three years given the start date of June 1, 2015.

Given that *awareness* of the resource is a prerequisite for *adoption*, we've allocated for substantial amounts of travel to professional conferences to promote the PSRD through poster presentations, talks, and direct community interaction. The PI will attend a Planetary Data Users Workshop at the USGS in Flagstaff AZ, a followup to the 2012 event, not yet scheduled but anticipated to occur near the beginning of the period of performance in 2015 [36]. The PI will also attend the Lunar and Planetary Sciences Conference in each year during the period of performance (2016, 2017, 2018), anticipated to be held yearly in the future, as it has in the past, during March at The Woodlands, TX. Both investigators will attend the American Geophysical Union (AGU) fall conference in Decmeber of 2017. Both Investigators will attend the Division of Planetary Sciences of the American Astronomical Society (DPS/AAS) meeting scheduled to take place in Pasadena, CA in November, 2016.

We also plan to produce two peer-reviewed publications describing this work. The first, following shortly after the completely of construction, design, and a short period for population, will be the announcement / “software paper” describing the project as a whole, its design goals and parameters, and the specific implementation. The second paper, released near the end of the period of work, will provide an update on the project with specific emphasis on “lessons learned.”

A solid foundation can be laid in three years, but it may take as long as a decade or more of continuous availability and utility for a service like this to be widely used. While the requested funding will propel this resource into existence and sustain it through its critical early years, the PI, in his role as the president and AOR for his institution, therefore makes a commitment that Million Concepts will ensure that PSRD remains available online for at least five years following the proposed period of work.

<u>Archival</u>

As required by the solicitation, the products of this effort will be archived with “PDS or an equivalent archive.” Our formal work products or deliverables are (1) the PSRD website and portal software source code, and (2) the contents of the software database itself. These are not classes of products that the Planetary Data System (PDS) either can or will archive [37, 38], so we have obtained permission to archive these products with ResearchCompendia (RC). Cofounded by Collaborator Seiler, RC is a platform which supports the goals of “reproducible research” by “hosting data, code, and methods in a form that is accessible, trackable, and persistent.” [27] They also have an agreement with Columbia University to ensure that, should the RC initiative end, projects hosted there will continue to be available through the University indefinitely. [See: Letter of Support from Collaborator Jennifer Seiler] Though our project and data products are both outside of the scope of the PDS mission, the PDS Engineering Node has agreed to integrate references to the PSRD into the PDS so that users of the PDS will be able to more easily find this resource. [See: Letter of Support from Daniel Crichton] The web site / portal source code produced in the course of this project will remain indefinitely available on a public software repository (e.g., Github) under a permissive license which allows modification and reuse.

References and Citations

[1] McMahon, Susan K. "Overview of the planetary data system." Planetary and Space Science 44.1 (1996): 3-12.
[2] Torson, J.M., and Becker, K.J. (1997) ISIS - A Software Architecture for Processing Planetary Images, Lunar Planet. Sci. XXVIII, Houston, Texas, Abstract #1219. (Software available at http://isis.astrogeology.usgs.gov/index.html)
[3] Gorelick, N. S., et al. "JMARS: A multimission data fusion application." Lunar and Planetary Institute Science Conference Abstracts. Vol. 34. 2003. (Software available at http://jmars.asu.edu/)
[4] Gaddis, L.R., Hare, T. and Beyer, R., 2013, Summary and Abstracts of the Planetary Data Workshop, USGS Open-File Report XXX, 193 pp.
[5] Stodden, Victoria C. "The scientific method in practice: Reproducibility in the computational sciences." (2010).
[6] Hong, Neil Chue, Brian Hole, and Samuel Moore. Software Papers: improving the reusability and sustainability of scientific software. Technical Report 795303, figshare, 2013. http://dx. doi. org/10.6084/m9. figshare. 795303.
[7] Chue Hong, N 2012, 'Doing Science Properly in the Digital Age' Paper presented at Digital Research 2012, Oxford, United Kingdom, 10/09/12 - 12/09/12, .
[8] Ince, Darrel C., Leslie Hatton, and John Graham-Cumming. "The case for open computer programs." Nature 482.7386 (2012): 485-488.
[9] Shamir, Lior, et al. "Practices in source code sharing in astrophysics."Astronomy and Computing 1 (2013): 54-58.
[10] Stodden, Victoria, and Sheila Miguez. "Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research." Social Science Research Network 2322276 (2013).
[11] "Google Scholar." Retrieved 9/17/2014. (http://scholar.google.com/)
[12] “JSTOR.” Retrieved 9/17/2014. (http://www.jstor.org/)
[13] Knepley, Matthew G., et al. Accurately Citing Software and Algorithms Used in Publications. Technical Report 785731, figshare, 2013. http://dx. doi. org/10.6084/m9. figshare. 785731, 2013.
[14] "Other (Non-NWS) Software Links." NOAA. Retrieved 9/17/2014. (http://www.nws.noaa.gov/im/softb.htm)
[15] Spink, Tim. “Geotechnical and Geoenvironmental Software Director.” Retrieved 9/17/2014. (http://www.ggsd.com/)
[16] “GIS Tools.” USGS Astrogeology Science Center. Retrieved 9/17/2014. (http://astrogeology.usgs.gov/facilities/mrctr/gis-tools)
[17] Law, Emily, and the IPDA Member Agencies. “IPDA Tools Registry.” IPDA. Retrieved 9/17/2014. (http://planetarydata.org/services/registry)
[18] Allen, Alice, and Judy Schmidt. "Looking before leaping: Creating a software registry." arXiv preprint arXiv:1407.5378 (2014).
[19] Hanisch, Robert J., Harry Payne, and Jeffrey JE Hayes. "An Astronomical Software Directory Service." Astronomical Data Analysis Software and Systems III. Vol. 61. 1994.

[20] Brunner, R.J.; Remijan, M. 2005, Astronomy catalog searching and software hosting
[21] Baffa, C., Giani, E., & Checcucci, A. 2004, in Astronomical Society of the Pacific Conference Series, Vol. 314, Astronomical Data Analysis Software and Systems (ADASS) XIII, ed. F. Ochsenbein, M. G. Allen, & D. Egret, 157
[22] Moore, B. 1999, AstroSim - A five year European program which provides research support for activities in computational astrophysics
[23] Brinkmann, S. 2007, astro-sim.org is up!
[24] Shortridge, K. 2009, in Astronomical Society of the Pacific Conference Series, Vol. 411, Astronomical Data Analysis Software and Systems XVIII, ed. D. A. Bohlender, D. Durand, & P. Dowler, 3
[25] Allen, Alice, et al. "Practices in Code Discoverability: Astrophysics Source Code Library." arXiv preprint arXiv:1202.1028 (2012).
[26] “Github.” Retrieved 9/17/2014. (https://github.com/)
[27] Stodden, V; Seiler, J.; Miguez, S.; “Our Mission and Goals.” ResearchCompendia. Retrieved 9/17/2014. (http://researchcompendia.org/)
[28] “Figshare.” Retrieved 9/17/2014. (http://figshare.com/)
[29] Joye, W. A., and E. Mandel. "New features of SAOImage DS9." Astronomical Data Analysis Software and Systems XII. Vol. 295. 2003.
[30] Warmerdam, Frank. "GDAL-Geospatial Data Abstraction Library." http://www. gdal. org/ (2005).
[31] Oliphant, Travis. "NumPy." (2007). (Software available at http://www.numpy.org/)
[32] Anderson, J., et al. “ISIS 3 Documentation: System Architecture” Retrieved 9/17/2014. (http://isis.astrogeology.usgs.gov/documents/SystemArchitecture/SystemArchitecture.html)
[33] ISO 26324
[34] Sykes, M; Lane, M.; Benecchi, S; “Planetary Exploration Newsletter.” Retrieved 9/17/2014. (http://planetarynews.org/)
[35] IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document, 1362-1998, 2007.
[36] Justin Hagerty, personal communication, June 2014.
[37] PDS Policy on Formats for PDS4 Data and Documentation (2014-06-30) (PDS4 only)
[38] Daniel Crichton, personal communication, September 2014.