Lenhardt, W C et al 2014 Data Management Lifecycle and Software Lifecycle Management in the Context of Conducting Science. *Journal of Open Research Software,* 2(1): e15, pp. 1-4, DOI: http://dx.doi.org/10.5334/jors.ax

ISSUES IN RESEARCH SOFTWARE

Data Management Lifecycle and Software Lifecycle Management in the Context of Conducting Science

W. Christopher Lenhardt[*], Stanley Ahalt[*], Brian Blanton[*], Laura Christopherson[*] and Ray Idaszak[*]

This paper examines the potential for comparisons of digital science data curation lifecycles to software lifecycle development to provide insight into promoting sustainable science software. The goal of this paper is to start a dialog examining the commonalities, connections, and potential complementarities between the data lifecycle and the software lifecycle in support of sustainable software. We argue, based on this initial survey, delving more deeply into the connections between data lifecycle approaches and software development lifecycles will enhance both in support of science.

Keywords: data lifecycle; software development lifecycle; sustainable software

Introduction

Software, ranging from spreadsheets with formulas to very large and complex numerical models that simulate the earth system, is used throughout most phases of science from experiments to analyses. The utility of this software is based, in part, on whether it fulfills the scientist's need for which it was created. Science is also defined by the gathering and use of data through observation, measurement, analysis, and simulation, and this data almost invariably is collected, created, and analyzed with software. In the digital age, the value of scientific data is often assessed by its potential for reuse in other contexts particularly for other scientific analyses, applications, and decision support. Digital scientific data has a lifecycle that moves from inception to creation through use, re-use, and curation. Software too has a lifecycle from inception to creation to testing, deployment, use, evolution, and refactoring. Given the tight connections between scientific data and science software, it is illustrative to compare these two lifecycles to look for commonalities and potentials for synergies between them to create more sustainable software.

A comparable effort investigated what data management can learn from software development practices [1]. This paper will do the opposite, suggesting that the well-accepted and well-documented digital science data lifecycle management best practices should inform the development of sustainable scientific software. We describe the data lifecycle and the software lifecycle in more detail to identify commonalities, synergies, and discontinuities between the two approaches. We suggest that there are four initial intersection points that warrant additional investigation: metadata, preservation management, data flows into and out of science software, and standards. The goal of this paper is to start the dialog and build a consensus to delve into the subject in greater detail through future research.

Data Management Lifecycle

The goal of a data management lifecycle is to ensure that scientific data are collected with enough rigor to support the intended use, to support basic data management, to enable reuse and repurposing of the data, and to allow for the eventual long-term preservation and management of the data. With respect to scientific research, the overarching purpose is to support scientific discovery, verifiability, and reproducibility.

Numerous versions of varying complexity exist of an abstracted data management or data lifecycle [2]. However, they all share certain basic steps. These include, but are not limited to: Plan, Collect, Quality Control, Document, Preserve, Use. [See **Figure 1**. See also, http://www.dataone.org/sites/all/documents/DataONE_BP_Primer_020212.pdf and http://www.data-archive.ac.uk/create-manage/life-cycle.] The various steps associated with the data lifecycle are on some level self explanatory. However, the basic process involves the creation, quality control, storage, and analysis of the data required to support the science. At a deeper level, the lifecycle is concerned with providing the necessary information to effectively manage the data in a long-term archival setting and to support reuse beyond the original purpose for which the data were created. Throughout the process, additional metadata and documentation are added.

* Renaissance Computing Institute (RENCI), University of North Carolina at Chapel Hill, Chapel Hill, USA
clenhardt@renci.org

Corresponding author: W. Christopher Lenhardt

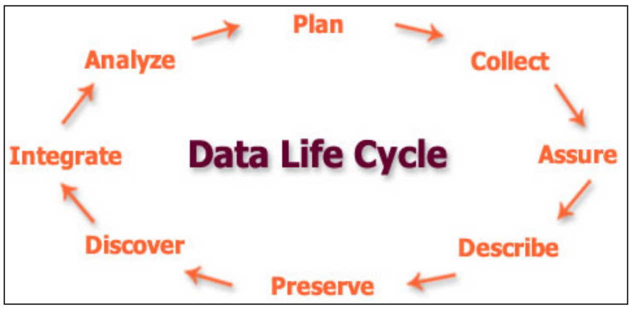

Figure 1: Data Life Cycle. © DataONE, available at http://www.dataone.org/best-practices [3].

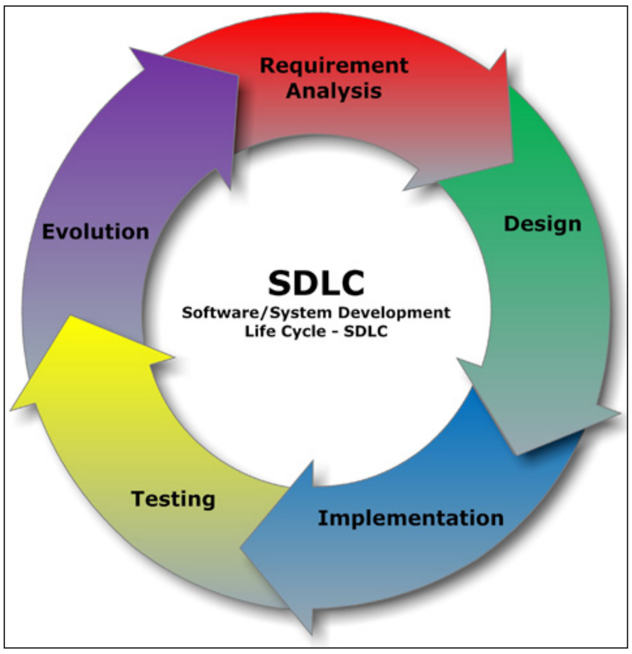

Figure 2: The Software Life Cycle, available at http://commons.wikimedia.org/wiki/File:SDLC_-_Software_Development_Life_Cycle.jpg (uploaded by user Cliffydcw).

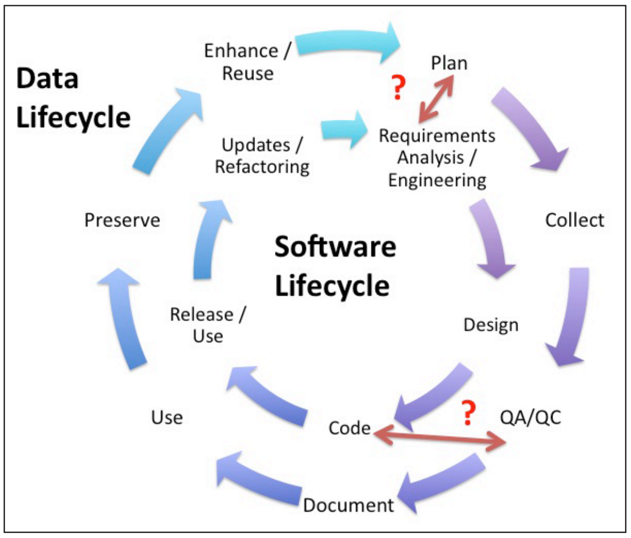

Figure 3: Comparison of High Level Lifecycle Phases for Science Data and Science Software.

The data management lifecycle is well documented and has led to a number of accepted best practices and standards. Two such examples are the Metadata Encoding and Transmission Standard (METS) [4] and the Open Archival Information System Reference Model (OAIS, ISO 14721: 2012) [5]. See also the ESIP Federation Data Management Short Course, http://commons.esipfed.org/datamanagementshortcourse [6].

Software Lifecycle

The software lifecycle (See **Figure 2**) [7] focuses on the development, maturation, and enhancement of a particular piece or set of software. This lifecycle pertains in its most basic form to the definition and creation of a piece of software. Similar to data lifecycles, there are various models of differing complexity reflecting different underlying methodologies for developing software. [8] These approaches, such as various forms of agile, waterfall, and spiral, are most formally applied to the context of commercial or production software. This raises another potential distinction, which is how to describe science software, as opposed to commercial software, which is often not developed according to software development best practices.

Linking the Two Lifecycles and Areas for Additional Investigation

The effective management of scientific data and science software share certain common goals such as supporting reproducible science and supporting potential reuse. The high level steps related to each lifecycle share a certain symmetry. However, there are points where it might be natural to connect the lifecycles such as during the quality assessment portion (see red arrows), or it might be useful to look for ways to connect the cycles more explicitly. See **Figure 3**.

We argue that delving into this lifecycle comparison in greater detail will yield interesting insights to promote sustainable software. For example, consider metadata for data. There are many different types of metadata, metadata standards, metadata tools for authorship, management, and search and retrieval. The more metadata, the more the value-add for the data and the data should be easier to manage and reuse. However, questions still remain. For example, what level of metadata is enough? Can we leverage science data standards in the context of developing science software?

Based on this preliminary analysis we suggest there are at least four different avenues for exploration that will contribute to the discussion of sustainable scientific software. The first is metadata, the second is data management principles, the third is the explicit connections of data flows into and out of science software, and the fourth is standards.

Metadata

As data moves through the data management lifecycle, there is increasing value added through the creation of additional metadata and documentation. The categories of metadata and documentation added suggest similar types of additional structured content that could or should be added during the science software development process. A recent article on climate model metadata underscores the need to improve the systematic documentation of model metadata in order to facilitate model intercomparison and future use. The implications of this should be explored in the context of sustainable science software [9].

Preservation Management Principles

Examining in more detail the actual principles behind science data management practices will also help to inform the science software development process. For example, best practices related to putting the data into a repository and also performing archival management will be relevant to how science software is released and managed. These will be particularly important in the context of validation of science software. Conversely, established open source community methods for software will inform how to better achieve open science with open data.

Data in / Data out

It is a reasonable assertion to state that much of what science software does is work with data. The software is used to perform data manipulation, data fusion, data transformation, data analyses, visualization, and for modeling. Not only does science software work with data, but scientific software often generates data. This is particularly true in the area of modeling. Therefore conducting a more systematic investigation of how data standards, and best practices might be used to facilitate the connection of data to science software might prove fruitful. For example, can the code be used to generate data that is 'self-documented'. Can machine-readable data and model code be connected in a way that facilitates sustainable software *and* data curation.

Standards

Internationally recognized standards abound for both data and software. Data standards include metadata standards such as ISO 19115, 14721, 16363. ISO 19115 pertains to geographic information metadata, ISO 14721 encompasses the Open archival information system (OAIS) -- Reference model, and ISO 16363 focuses on audit and certification of trustworthy digital repositories. The goal of these standards is to develop appropriate documentation for and to support the management of digital data for the long-term. Complimentary software related ISO standards include ISO 25010 relating to systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE), and ISO/IEC/IEEE 29119 Software Testing. The goal of software-related standards is to ensure the functionality and reliability of software so that the software will function as intended for end-users. ISO standards, particularly on the data side are encoded in machine-readable forms which include the value added content. Software standards at times include things like data models, and specifications related to quality. These types of standardized approaches to key aspects of both data lifecycle and software lifecycle should be linked.

Counter Arguments

While this paper takes the position that there is much to be gained by systematic examination of the data and software lifecycles together, there may be circumstances where the notion of linking the two may not be fruitful or is unwarranted. For example, there are plenty of applications the primary purpose of which is not science-related, e.g. gaming, office productivity, communications and social media. Data gathered from these types of applications may have a scientific purpose, but the primary goal of these is not to support science. Therefore, addressing software sustainability as a function of lifecycle will most likely be misplaced. Second, commercial software will be difficult to assess from a development lifecycle perspective and cross-reference to a data lifecycle perspective. Corporations that develop software for profit, for reasons of competitive advantage, may consider information related to their software development practices proprietary making comparisons to data lifecycles difficult. Finally, there is an issue of scale. Is it reasonable to treat a code snippet or script the same way we should treat software that supports large science data systems, such as Earth observing satellites or IPCC climate models? Probably not when it comes to a comparison of a data lifecycle with a software lifecycle in this context. However, good software engineering should be applied at the most basic level of code development. Similarly, data curation should also be considered even for the most simple science data collection activities.

These counterarguments not withstanding, given the trends to use more nontraditional and more disparate types of data as inputs for scientific analyses, for example analysis of social media content or other transactional data, and given that fact that most scientific data is "born digital", it may be increasingly difficult to find clear cut cases where one should not consider the implications of data management lifecycles and software lifecycles. The crucial distinction in this context would seem to be whether or not the purpose of the software or outputs are to be used scientifically. Once that line is crossed, then scientific rigor suggests a higher standard when it comes to these types of issues.

Why does this matter?

The goal of this paper was to start a dialog on examining the commonalities, connections, and potential complementarities between the data lifecycle and the software lifecycle. The goal was not to present a comprehensive examination and analysis. This paper is intended as a thought piece to explore whether there was sufficient justification to look into this approach in greater detail. We think there is potential for much more on this front. While this paper addresses the question largely from the data management perspective, it is not intended to imply that data management has all the answers or that data management cannot learn from software management. Quite the contrary, we should be doing both. As with science data curation, doing a better job of promoting the sustainability of science software is important for a number of reasons, including the support of scientific reproducibility, software reuse, and longer term viability of science software. However, there are two final outcomes linked to creating more sustainable science software. First, sustainable software should enable new science and second, sustainable software should make the scientist's job easier.

Acknowledgements

This work is funded in part by National Science Foundation (NSF) award 1216817 Conceptualization of a Water Science Software Institute. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the NSF. The authors would also like to thank the reviewers for their constructive comments.

Reference

1. **Schopf, J M** JCDL '12 Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries. New York, NY, USA: ACM, pp. 153-156. DOI: http://dx.doi.org/10.1145/2232817.2232846
2. **Ball, A** 2012 *Review of Data Management Lifecycle Models.* Bath, UK: University of Bath. Available at: http://opus.bath.ac.uk/28587/
3. **DataONE** Available at: http://www.dataone.org/best-practices
4. **Metadata Encoding and Transmission Standard (METS)** Available at: http://www.loc.gov/standards/mets/mets-home.html
5. **Open Archival Information System (OAIS) Reference Model** Available at: http://public.ccsds.org/publications/archive/650x0m2.pdf
6. **Federation of Earth Science Information Partners (ESIP)** 2012-2013 Data Management for Scientists Short Course. Various authors. Available at: http://commons.esipfed.org/datamanagementshortcourse
7. **Wikimedia Commons** Available at: http://commons.wikimedia.org/wiki/File:SDLC_-_Software_Development_Life_Cycle.jpg
8. **Munassar, N M A** and **Govardhan, A** 2010 A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science Issues*, 7(5): 94-101. Available at: http://www.ijcsi.org/papers/7-5-94-101.pdf
9. **Guilyardi, E,** et al 2013 Documenting climate models and their simulations. *Bulletin of the American Meteorological Society*, 94(5): 623–627. DOI: http://dx.doi.org/10.1175/BAMS-D-11-00035.1

How to cite this article: Lenhardt, W C, Ahalt, S, Blanton, B, Christopherson, L and Idaszak, R 2014 Data Management Lifecycle and Software Lifecycle Management in the Context of Conducting Science. *Journal of Open Research Software,* 2(1): e15, pp. 1-4, DOI: http://dx.doi.org/10.5334/jors.ax

Published: 9 July 2014