Lightning Talk: A Proposal for the Measurement and Documentation of Research Software Sustainability in Interactive Metadata Repositories

Stephan Druskat
Department of German Studies and Linguistics
Humboldt-Universität zu Berlin
Berlin, Germany
stephan.druskat@hu-berlin.de

***Abstract*—This paper proposes an interactive repository type for research software metadata which measures and documents software sustainability by accumulating metadata, and computing sustainability metrics over them. Such a repository would help to overcome technical barriers to software sustainability by furthering the discovery and identification of sustainable software, thereby also facilitating documentation of research software within the framework of software management plans.**

***Index Terms*—Metadata, repositories, technical sustainability, information search and retrieval, measurement, software management plans.**

The identification and discovery of sustainable research software present technical barriers to software sustainability [1, p. 16f.]. This paper proposes to tackle these barriers by accumulating and quantifying software metadata in interactive metadata repositories (IMRs) to enable identification of sustainable software through *sustainability measurement*, and simplified discovery of suitable software through *documentation of software and its sustainability*.

In analogy to the discussion about data sustainability and funders' mandate of data management plans (DMPs), software management plans (SMPs) would be an obvious tool for securing sustainability of newly developed software. DMPs feature data repositories as natural targets for research data releases as part of the data lifecycle documentation, and analogously, software repositories should be an integral part of software lifecycles as detailed in SMPs. In addition to source code repositories and distribution repositories for deliverables, there are also already a large number of research data repositories which archive research software source code and deliverables along with some metadata (cf. re3data.org). However, while the latter may provide support for assessing the suitability of a software for the intended research application, they do not provide a lot of information about, not to mention metrics of, the software's sustainability. The same is true for existing *metadata* repositories dedicated to research software, e.g., SciencePAD, the EGI Applications Database, and the DiRT Directory. There are also other platforms providing metadata for software, e.g., GitHub, Open Hub, and Zenodo. Some of the metadata they provide – a rather small portion in the case of GitHub and Zenodo, some more in the case of Open Hub – can indeed be used to discover some sustainability aspects of the software they cover. However, this facet is not presented explicitly in either of the platforms. In any case, considering the available tools and platforms, users that are interested in acquiring a comprehensive and in-depth understanding of how sustainable a software is will have to interpret and extrapolate to a high degree.

However, a repository can be a suitable means to provide documentation and measurement of software sustainability, specifically by utilising software metadata. In order to do so, it must: accumulate the metadata for a software that is relevant for determining both its suitability for a research application (to allow for straightforward discovery of software), and its sustainability; compute sustainability metrics; document the metrics and provide means to interactively evaluate and refine some of them. Such a repository would also benefit projects which will be subject to funding agencies' potential mandate of SMPs, as it represents a natural means of documenting the sustainability of software developed within the project.

The conception of the envisioned IMR poses at least three theoretical challenges, namely (1) how to give "software sustainability" an operationalisable definition, (2) how to define quantifiable criteria for sustainability, and (3) how to design algorithms for computing comprehensible and reproducible metrics for sustainability. While detailing (3) is beyond the scope of this paper and is left for future research, (1) and (2) are approached briefly in the following sections.

"Software sustainability" is an ambiguous concept, with different agents approaching it from different angles. Reference [2] focuses on the ecological sustainability of software, discussing software engineering only briefly. Reference [3] focuses on the development process rather than the product. Reference [4] – specifically discussing research software – names training, availability, community recognition for developers, dedicated job descriptions, and funding as defining factors for sustainability. A systematic approach to a definition of sustainability in the context of software engineering is presented in [5], which utilizes a basic definition of sustainability as "preserving the function of a system over a defined

timespan" [5, p. 1184] for the refinement of detailed aspects of sustainability for and in software engineering. However, while [5]'s approach would be operationalisable, it uses a generic, multi-dimensional definition of sustainability (cf. [6, p. 4]) and applies it to *engineering processes* rather than to *features of products* and is therefore not directly adaptable for IMRs.

A definition of sustainability that can be operationalized for the purposes of an IMR must in fact be based on the definition of "technical sustainability" from [7, p. 470f.]: "Technical [sustainability] refers to longevity of information, systems, and infrastructure and their adequate evolution with changing surrounding conditions." The two key points of this definition, *longevity* and *evolution*, can be concretized by transforming another existing global concept of "sustainability". Following [8], an enhancement of a three-dimensional general model of sustainability ("three column model"), we can preliminarily define "software sustainability" via its goals: *The three goals of software sustainability are* (1) *ensuring the existence of the software,* (2) *preserving the potential for productive operation of the software,* (3) *creating and retaining possibilities for further development and adaptation of the software.* This definition, finally, should be employable for defining and categorizing suitable criteria for measuring the sustainability of research software.

An IMR must accumulate quantifiable metadata pertaining to all three sustainability goals. The parameters for the metrics to be computed over these metadata must be based on criteria that are themselves categorized along the lines of the sustainability goals. One starting point for the compilation of a list of criteria is [9], which defines criteria for "sustainability", "maintainability", and "usability" to facilitate the evaluation of software products (e.g.: "Documentation is on the project web site", "software has an open source licence", "source code is commented"). As the definition for sustainability given above includes all three of these notions (usability arguably counting towards goal (2)), the items given for the criteria in [9] can be transformed into parameters for computing sustainability metrics for an IMR, although some items may require further definitory work to make them quantifiable. Further criteria that are not included in [9] – e.g., pertaining to human resources available for a software project – can be identified introspectively, elicitated or crowd-sourced, and made quantifiable.

In order to leverage fully the available metadata for a software, and at the same time preserve the reproducibility of the sustainability metric and secure it against manipulation, it is expedient to use more than one metric for the representation of sustainability. E.g., a "hard" metric could be computed over metadata that is both quantifiable and objective (such as whether a software is open source or not), a "semi-hard" metric over metadata that is objective but hard to quantify (e.g., use of a programming language or build system that in itself may be more or less sustainable), and a "soft" metric taking into account qualitative and subjective metadata (e.g., whether a UI is intuitive). Both the "hard" and "semi-hard" metrics would be reproducible and would likely be hard to tamper with, as they are based on objective metadata. The use of these three metrics would additionally allow for user interaction with the metadata, i.e., users could review and evaluate metadata, and these evaluations could in turn be used to detect bad data and re-compute metrics where necessary. User contributions in the form of, e.g., votes on or grading of specific metadata points would inherently be constituting the "soft" metric, while other forms of contributions, such as documenting the use of a software (e.g., in conjunction with a reference to work documenting the use of the software), could contribute to either of the "harder" metrics as well. Interactivity can also be used for gamification purposes to attract users, and provide software originators with further incentive to submit their data, e.g., by releasing standings tables and issuing metrics graphics for use on websites, etc.

The accumulation of the metadata can be performed by different means: direct input by the originator of the software, harvesting data from existing repositories via, e.g., GitHub and Open Hub APIs, dedicated crawling of source code repositories, etc. The latter two methods can also be used for verification and a preliminary quantification of the input. The metadata must be available in the repository as structured text in a well-defined format, e.g., based on XML or JSON.

In summary, an interactive metadata repository, field-specific or not, that measures and documents the sustainability of software can be a valuable tool not only for the discovery of research software, but also for the identification of sustainable – and therefore preferable – software, and as an integral part of the implementation of SMPs.

In future research, the author intends to further develop the idea of such IMRs within the framework of a PhD thesis.

References

[1] S. Hettrick, "Research software sustainability: Report on a Knowledge Exchange workshop," The Software Sustainability Institute, Tech. Rep., 2016. [Online]. Available: http://repository.jisc.ac.uk/6332/

[2] J. Gröger and M. Köhn, "Nachhaltige Software. Dokumentation des Fachgesprächs "Nachhaltige Software" am 28.11.2014," Umweltbundesamt, Dokumentationen 07/2015, June 2015. [Online]. Available: http://www.umweltbundesamt.de/en/publikationen/nachhaltige-software

[3] K. Tate, *Sustainable Software Development: An Agile Perspective*. Boston, Mass.: Addison-Wesley, 2005.

[4] C. Goble, "Better software, better research," *IEEE Internet Computing*, vol. 18, no. 5, pp. 4–8, 2014.

[5] B. Penzenstadler, "Towards a definition of sustainability in and for software engineering," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 1183–1185. [Online]. Available: http://doi.acm.org/10.1145/2480362.2480585

[6] B. Penzenstadler and H. Femmer, "A generic model for sustainability with process- and product-specific instances," in *Proceedings of the 2013 Workshop on Green in/by Software Engineering*, ser. GIBSE '13. New York, NY, USA: ACM, 2013, pp. 3–8. [Online]. Available: http://doi.acm.org/10.1145/2451605.2451609

[7] C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability design and software: The Karlskrona Manifesto," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, May 2015, pp. 467–476.

[8] J. Jörissen, J. Kopfmüller, V. Brandl, and M. Paetau, *Ein integratives Konzept nachhaltiger Entwicklung*, ser. Wissenschaftliche Berichte FZKA. Karlsruhe: Forschungszentrum Karlsruhe, 1999, no. 6393. [Online]. Available: http://www.itas.kit.edu/pub/v/1999/joua99a.pdf

[9] M. Jackson, S. Crouch, and R. Baxter, "Software evaluation: criteria-based assessment," *Software Sustainability Institute*, 2011.