Lightning Talk:"I solemnly pledge" A Manifesto for Personal Responsibility in the Engineering of Academic Software

Alice Allen, Cecilia Aragon, Christoph Becker, Jeffrey Carver, Andrei Chis, Benoit Combemale, Mike Croucher, Kevin Crowston, Daniel Garijo, Ashish Gehani, et al.

HAL Id: hal-01367344

https://hal.inria.fr/hal-01367344

Submitted on 15 Sep 2016

Lightning Talk: "I solemnly pledge"

A Manifesto for Personal Responsibility in the Engineering of Academic Software

Alice Allen[1], Cecilia Aragon[2], Christophe Becker[3], Jeffrey C. Carver[4], Andrei Chis[5], Benoit Combemale[6], Mike Croucher[7], Kevin Crowston[8], Daniel Garijo[9], Ashish Gehani[10], Carole Goble[11], Robert Haines[11], Robert Hirschfeld[12], James Howison[13], Kathryn Huff[14], Caroline Jay[11], Daniel S. Katz[15], Claude Kirchner[16], Kateryna Kuksenok[2], Ralf Lämmel[17], Oscar Nierstrasz[5], Matthew Turk[15], Rob V. van Nieuwpoort[18], Matthew Vaughn[13], Jurgen Vinju[19]

[1]University of Maryland, USA
[2]University of Washington, Seattle, USA
[3]University of Toronto, Canada
[4]University of Alabama, USA
[5]University of Bern Switzerland
[6]IRISA, Rennes, France
[7]University of Sheffield, UK
[8]Syracuse University, USA
[9]Technical University of Madrid, Spain
[10]SRI, Menlo Park, USA
[11]The University of Manchester, UK
[12]Hasso-Plattner-Institut, Potsdam, Germany
[13]University of Texas, Austin, USA
[14]University of California, Berkeley, USA
[15]University of Illinois, Urbana-Champaign, USA
[16]INRIA, Le Chesnay, France
[17]Universität Koblenz-Landau, Germany
[18]Netherlands eScience Center, Amsterdam, The Netherlands
[19]CWI Amsterdam, The Netherlands

Abstract— **Software is fundamental to academic research work, both as part of the method and as the result of research. In June 2016 25 people gathered at Schloss Dagstuhl for a week-long Perspectives Workshop and began to develop a manifesto which places emphasis on the scholarly value of academic software and on personal responsibility. Twenty pledges cover the recognition of academic software, the academic software process and the intellectual content of academic software. This is still work in progress. Through this lightning talk, we aim to get feedback and hone these further, as well as to inspire the WSSSPE audience to think about actions they can take themselves rather than actions they want others to take. We aim to publish a more fully developed Dagstuhl Manifesto by December 2016.**

Index Terms—software, manifesto.

I. INTRODUCTION

Software is fundamental to academic research work, both as part of the method and as the result of research. With the advent of artifact evaluation committees of conferences, journals that include source code and running systems as part of the published artifacts, as well as the increasing push to reproducibility, we foresee that software will only increase in importance as part of the academic process.

In June 2016, 25 people gathered at Schloss Dagstuhl for a weeklong Perspectives Workshop [1] to produce a roadmap towards future professional software engineering for software-based research instruments and other software produced and used in an academic context (i.e., for research, not administration). The group was carefully picked to be broad in its range of disciplines (including Astronomy, Social Sciences, Biology, Chemistry, Computer Science, Physics, and Humanities), roles (including computer science researchers, general and specialist research software engineers and systems administrators) and career stages (from PIs and institute heads to PhD students and postdocs). Despite its ambiguous title, "Engineering Academic Software," the workshop was on the Engineering *of* Academic Software (not software for engineers).

A Dagstuhl Perspectives Workshop results in a Manifesto. The open science and research software communities have been very active in creating manifestos: the Science Code Manifesto [2], Karlskrona Manifesto for Sustainability Design [3], the Reproducibility Manifesto [4], and Principles for Software Citation [5], FAIR data [6], and so on. Why do we need another Manifesto?

First, our manifesto is to be less about what others should do, and more about what we, as *individuals*, should do. That is, it is more in the style of a *personal responsibility pledge* like those on open access [7] and open peer review [8]. It is easy for us to declare "the community should do X", "funding panels should do Y" and "promotion committees should do Z", while conveniently forgetting that we are the community, we are the panelists, we are the committee members.

Second, our manifesto places emphasis on the scholarly value of academic software. For some of our group, engineering academic software is chiefly a means to an end – to pro-

duce robust and reliable software as an instrument as part of a wider research investigation. For others the software is the end in itself – the software is the research. In both cases the software has scholarly merit.

II. THE DRAFT PERSONAL MANIFESTO

Currently at 20 there are too many pledges so we are actively working on reducing the number and simplifying the message. At the WSSSPE4 meeting, through this lightning talk, we aim to get feedback and contribute to this process.

Table I presents our pledge list organized into three broad areas: (1) recognition of academic software, (2) academic software development processes and (3) the intellectual content of academic software. Each pledge should be actionable by an individual. Each pledge has a story that will be developed in the full manifesto.

TABLE I. MANIFESTO DRAFT PLEDGES

	A. Recognition of academic software
1	I will properly cite software used to produce my research results
2	I will point out improper or missing citations to software when I am reviewing publications.
3	I will make explicit how to cite the software I make available.
4	I will recommend software experts for funding agencies to include in their review processes.
5	I will invite developers of software that enables my research to be co-authors on my papers.
6	I will recognize software contributions in hiring and promotion within my institution.
7	I will recognize software contributions at conferences, e.g. dedicated sessions, and prizes.
8	I will support and publish in journals that recognise software contributions.
9	I will contribute to sustaining the software I rely on for my research.
	B. Academic software development processes
10	I will develop software as open source right from the start whenever possible.
11	I will document my academic software for users with instructions and examples.
12	I will package, release and archive versions of my software
13	I will consider and document the sustainability of my research software.
14	I will publish how I organize and run my software projects
15	I will match software engineering practices I recommend to the needs and resources of projects.
16	I will help scientists improve the quality of their software without passing judgment.
	C. The intellectual content of academic software
17	I will acknowledge that source code is a legitimate part of the academic discourse.
18	I will publish the intellectual contributions of my research software.
19	I will distinguish the intellectual contribution of my software from its service contribution.
20	I will examine the source code of academic software contributions and encourage others to do so as well.

The pledges A and B (1-16) are targeted at both developers and users of academic software; pledges C (17-20) are more focused on developers. Pledges to recognize software are self-evidently responsibilities to be borne by users. Perhaps more implicit are the benefits to users embedded in the software development processes. Open source (10) and documentation (11) enables feedback and potential community engagement during critical design stages, and examples and instructions by their definition must embed an understanding of use cases and usability pitfall. Improving transparency of process, and analytic consideration of future use cases (12, 13, 14) are valid user-centered approaches and (15, 16) directly speak to assessing user needs and crafting appropriate interventions.

III. NEXT STEPS

This is still work in progress. We recognize that 20 pledges is roughly twice as many as desirable, and that the pledges need to be succinct and easy to understand and adopt, more in the style of the Reproducibility Manifesto [4]. We are currently (1) looking to merge pledges where feasible and (2) exploring the use of roles to structure them; for example "When I develop software ...When I write research papers ...When I evaluate colleagues ..." and so on. At WSSSPE we intend to run an online vote on the pledges structured on roles.

We aim to develop the full manifesto by December 2016 and publish it as a Dagstuhl Manifesto [9]. We intend that community groups promote its contents among researchers and research software engineers, and we use it to influence decision makers to enable our respective communities to execute these pledges with moral and financial support.

REFERENCES

[1] Engineering Academic Software, Dagstuhl Perspectives Workshop 16252, 19–24 June 2016, http://www.dagstuhl.de/16252 last accessed 07-Jul-2016

[2] Science code manifesto http://sciencecodemanifesto.org/

[3] Karlskrona Manifesto for Sustainability Design http://sustainabilitydesign.org/

[4] Reproducibility manifesto http://lorenabarba.com/gallery/reproducibility-pi-manifesto/

[5] A.M. Smith, D.S. Katz, K.E. Niemeyer, FORCE11 Software Citation Working Group. (2016) Software citation principles. PeerJ Preprints 4:e2169v3 https://doi.org/10.7287/peerj.preprints.2169v3

[6] M.D. Wilkinson et al. The FAIR guiding principles for scientific data management and stewardship (2016) Scientific Data 3, Article number: 160018 (2016) doi:10.1038/sdata.2016.18

[7] Open access pledge http://www.openaccesspledge.com/

[8] Open science peer review oath http://f1000research.com/articles/3-271/v2

[9] Dagstuhl manifestos http://www.dagstuhl.de/en/publications/dagstuhl-manifestos