Towards a Software Pillar for Open Science
brining software to the limelight

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

May 2022
Outline

1. Software and Open Science
2. Building the software pillar of Open Science: assessing the needs
3. Focus on ARDC and infrastructures
4. Call to action
Software is a pillar of Open Science

Software powers modern research

[…] software […] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don’t have the software, you don’t have the data

Christine Borgman, Paris, 2018

Nota Bene

software may be a tool, a research outcome and a research object

access to the source code is essential!

Preserving (the history of) source code is necessary for reproducibility

The links in the picture are important

Missing pillar: software (source code)
Software *Source Code* is Precious Knowledge

Harold Abelson, *Structure and Interpretation of Computer Programs* (1st ed.) 1985

“Programs must be written for people to read, and only incidentally for machines to execute.”

---

### Apollo 11 source code (excerpt)

<table>
<thead>
<tr>
<th>P63SPOT3</th>
<th>CA</th>
<th>BIT6</th>
<th># IS THE LR ANTENNA IN POSITION 1 YET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXTEND</td>
<td>RAND</td>
<td>CHAN33</td>
</tr>
<tr>
<td></td>
<td>EXTEND</td>
<td>BZF</td>
<td>P63SPOT4</td>
</tr>
<tr>
<td></td>
<td>CAF</td>
<td>CODE508</td>
<td># ASTRONAUT: PLEASE CRANK THE</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>BANKCALL</td>
<td># SILLY THING AROUND</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>GOPERF1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>GOTOP00H</td>
<td># TERMINATE</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>P63SPOT3</td>
<td># PROCEED SEE IF HE’S LYING</td>
</tr>
<tr>
<td>P63SPOT4</td>
<td>TC</td>
<td>BANKCALL</td>
<td># ENTER INITIALIZE LANDING RADAR</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>SETPOS1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>POSTJUMP</td>
<td># OFF TO SEE THE WIZARD ...</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>BURNBABY</td>
<td></td>
</tr>
</tbody>
</table>

---

### Quake III source code (excerpt)

```c
float Q_rsqrt(float number)
{
    long i;
    float x2, y;
    const float threehalves = 1.5F;
    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this can be removed
    return y;
}
```

---

Len Shustek, Computer History Museum 2006

“Source code provides a view into the mind of the designer.”
Source code is *special* (software is not data)

Software *evolves* over time
- projects may last decades
- the *development history* is key to its *understanding*

Complexity
- *millions* of lines of code
- large *web of dependencies*
  - easy to break, difficult to maintain
  - *research software* a thin top layer
- sophisticated *developer communities*

The human side
- design, algorithm, code, test, documentation, community, funding
  - and so many more facets ...
UNESCO, Inria, Software Heritage invite 40 international experts to meet in Paris. The call is published on Feb 2019.

“[We call to] promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, sharing good practices and recognising in the careers of academics their contributions to high quality software development, in all their forms.”

Open and promote research software source code

- actions (selection)
  - charter for research software policy
  - recognize software development (see announcement of the 2021 prize)
  - coordinate communities of practice
  - connected ecosystem of research outputs

- recommendations (selection)
  - archive in Software Heritage
  - standardise and use SWHID
  - build a national catalog of research software
  - leverage ADAC network

See official announcement

The "Collège Logiciel" of the National Committee on Open Science (CoSO) is now live!

R. Di Cosmo  roberto@dicosmo.org  (CC-BY 4.0)
Outline

1. Software and Open Science
2. Building the software pillar of Open Science: assessing the needs
3. Focus on ARDC and infrastructures
4. Call to action
## A plurality of needs

<table>
<thead>
<tr>
<th>Researchers</th>
<th>Laboratories/teams</th>
<th>Research Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>• archive and reference software used in articles</td>
<td>• track software contributions</td>
<td>• know its software assets</td>
</tr>
<tr>
<td>• find useful software</td>
<td></td>
<td>• technology transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• impact metrics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• funding strategy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• career evaluation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• produce reports</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• maintain web page</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• verify, reproduce, improve results</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Description</td>
<td>Note</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Archive</td>
<td>Research software artifacts must be properly archived</td>
<td>make sure we can retrieve them (reproducibility)</td>
</tr>
<tr>
<td>Reference</td>
<td>Research software artifacts must be properly referenced</td>
<td>make sure we can identify them (reproducibility)</td>
</tr>
<tr>
<td>Describe</td>
<td>Research software artifacts must be properly described</td>
<td>make it easy to discover and reuse them (visibility)</td>
</tr>
<tr>
<td>Cite/Credit</td>
<td>Research software artifacts must be properly cited (not the same as referenced!)</td>
<td>to give credit to authors (evaluation!)</td>
</tr>
</tbody>
</table>
Outline

1. Software and Open Science
2. Building the software pillar of Open Science: assessing the needs
3. Focus on ARDC and infrastructures
4. Call to action
Forges are *not* archives!

2015: the first big bad news

Google Code and Gitorious.org shutdown: ~1M endangered repositories
- broken links in the web of knowledge (my papers too)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250,000+ repositories (including research software)

2021: … in Academia too

- october 2021: Inria’s old gforge is unplugged
  - breaks the build chain of the OCaml package manager (Opam)

Bottomline

we need a universal archive of software source code: now we have one!
Software Heritage in a nutshell

Collect, preserve and share all software source code

Preserving our heritage, enabling better software and better science for all

Reference catalog
find and reference all software source code

Universal archive
preserve all software source code

Research infrastructure
enable analysis of all software source code
The largest software archive, a shared infrastructure

Software Heritage

- Source files: 12,032,627,304
- Commits: 2,536,918,821
- Projects: 173,242,749

- Directories: 9,946,192,395
- Authors: 47,334,620
- Releases: 31,763,605
Addressing the four needs (see ICMS 2020 for details)

Archive (12B+ files, 170M+ projects)

- save.softwareheritage.org
- deposit.softwareheritage.org

Reference (20 billion SWHIDs)

Intrinsic, decentralised, cryptographically strong identifiers, SWHIDs

Now supported in SPDX 2.2, Wikidata etc.

Describe

- **Intrinsic metadata** from source code
- Contributed the Codemeta generator

Cite/Credit

- Contributed *software citation* style
  biblatex-software, v 1.2-2 now on CTAN
Best practices for ARDC

Archiving and referencing

For all source code used in research (yes, even small scripts!)

- ensure it is archived in Software Heritage (see save code now)
- get the proper SWHID for your software (see detailed HOWTO)
- add it to research articles for reproducibility (see detailed HOWTO)

Describing and Citing/Crediting

For software you want to put forward (mention in your CV, reports, etc., get citations and credit for it), do the following extra steps:

- add codemeta.json with description (see the codemeta generator)
- reference in the HAL portal (french partners, see online HAL documentation)
- cite software using the biblatex-software package (in CTAN and TeXLive)
Demo time: a walkthrough

- **Browse the archive**
- **Trigger archival** of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- Example with Parmap: development on Github, archive in SWH, curated deposit in HAL
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Extracting all the software products for Inria, for CNRS, for LIRMM or for Rémi Gribonval using HalTools
- Example in a journal: an article from IPOL
Outline

1. Software and Open Science

2. Building the software pillar of Open Science: assessing the needs

3. Focus on ARDC and infrastructures

4. Call to action
Let's foster adoption

Train students and colleagues to archive and reference relevant source code
- full details in the ICMS 2020 article
- short operational HOWTO online
- french connection: deposit in HAL
- see the following presentation

Engage conferences, journals, learned societies to use Software Heritage and SWHIDs
APIs for save code now and deposit are available to integrate with
- Research Articles
- Artifact Evaluation Committees
- Badging initiatives

Help grow and structure the community
- Promote the ambassador program
- Encourage our institutions to
  - include Software Heritage in their Open Science policy
  - become member/sponsor
  - build a Software Heritage mirror (see ENEA)
Some pointers

it’s a long road, but together we can make it

References

UNESCO, Draft recommendations on Open Science
2021, (online)

French Ministry of Research, Second National Plan for Open Science
2021, (online)

EOSC SIRS Task Force, Scholarly Infrastructures for Research Software
2020, Publications office of the European Commission, (10.2777/28598)

R. Di Cosmo, Archiving and Referencing Source Code with Software Heritage
International Conference on Mathematical Software 2020 (10.1007/978-3-030-52200-1_36)

J.F. Abramatic, R. Di Cosmo, S. Zacchirol, Building the Universal Archive of Source Code
CACM, October 2018 (10.1145/3183558)