The Software Pillar of Open Science
policy, needs, and how to address them

Roberto Di Cosmo
Director, Software Heritage
Inria and Université de Paris Cité

May 2022
1. Software and Open Science
2. Policy framework and growing needs
3. Can you address these needs?
4. Yes you can!
5. 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

A key pillar: software (source code)

The links in the picture are important
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)

```
P63SPOT3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET
EXTEND
RAND CHAN33
EXTEND
BZF P63SPOT4 # BRANCH IF ANTENNA ALREADY IN POSITION 1
CAF CODE508 # ASTRONAUT: PLEASE CRANK THE SILLY THING AROUND
TC BANKCALL #
CADR GOPERF1
TCF GOTOPOOH # TERMINATE
TCF P63SPOT3 # PROCEED SEE IF HE’S LYING
P63SPOT4 TC BANKCALL # ENTER INITIALIZE LANDING RADAR
CADR SETPOS1
TC POSTJUMP # OFF TO SEE THE WIZARD ...
```

Quake III source code (excerpt)

```
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.”

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)
1. Software and Open Science

2. Policy framework and growing needs

3. Can you address these needs?

4. Yes you can!

5. Call to action
Experts call for greater recognition of software source code as heritage for sustainable development.

6 November 2018

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
Remise des prix science ouverte du logiciel libre de la recherche

Le ministère de l’Enseignement supérieur, de la Recherche et de l’Innovation remet pour la première année les Prix science ouverte du logiciel libre de la recherche. Dix logiciels mis au point par des équipes françaises sont récompensés pour leur contribution à l’avancée de la connaissance scientifique.
A plurality of needs that we must address

<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>produce reports</td>
<td>technology transfer</td>
</tr>
<tr>
<td>get credit for developed software</td>
<td>maintain web page</td>
<td>impact metrics</td>
</tr>
<tr>
<td>verify, reproduce, improve results</td>
<td></td>
<td>funding strategy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>career evaluation</td>
</tr>
</tbody>
</table>
| **Archive** | Research software artifacts must be properly **archived**  
make sure we can **retrieve** them (**reproducibility**) |
| **Reference** | Research software artifacts must be properly **referenced**  
make sure we can **identify** them (**reproducibility**) |
| **Describe** | Research software artifacts must be properly **described**  
amake it easy to **discover** and **reuse** them (**visibility**) |
| **Cite/Credit** | Research software artifacts must be properly **cited** (*not the same as referenced!*  
to give **credit** to authors (**evaluation!**)) |
A word of warning: 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!
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
An international, non profit initiative built for the long term

Sharing the vision

Donors, members, sponsors

And many more ...

www.softwareheritage.org/support/testimonials

www.softwareheritage.org/support/testimonials

Towards a Software Pillar of Open Science

May 2022
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

Towards a Software Pillar of Open Science May 2022 12 / 16
HAL and Software Heritage: building a curated software catalog

https://hal.archives-ouvertes.fr/hal-02130801

swh:1:dir:393b611a1424f032e83569bf6762502371cfc65
Outline

1. Software and Open Science
2. Policy framework and growing needs
3. Can you address these needs?
4. Yes you can!
5. Call to action
An example is worth a thousand words

- Browse the archive (your work may be already there!)
- Trigger archival of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- Cite software using the biblatex-software package from CTAN
- Example in a journal: an article from IPOL
- Example with Parmap: devel on Github, archive in SWH, curated deposit in HAL
- Extracting all the software products for Inria, for CNRS, for LIRMM or for Rémi Gribonval using HalTools
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Example use in a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Example use in a research article: extensive use of SWHIDs in a replication experiment
Outline

1. Software and Open Science
2. Policy framework and growing needs
3. Can you address these needs?
4. Yes you can!
5. Call to action
### 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)*
Bottomline

HAI+SWH let you address all the needs at once…

- **researcher, engineer**: archival, reference, credit, CV etc. *with a little effort from them*
- **labs, organizations**: track and report software production in a simple way
- **technology transfer offices**: view the software production
- **national level**: a *curated* catalog of the software production

… *with a little effort from your side*

- Update the Open Science policy to include software
- Train on the use of SWH and HAL for software
- Join the network of HAL moderators for software

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

**Questions?**