Vers un pilier logiciel de la Science Ouverte

défis et opportunités pour la reproductibilité et pour la science ouverte

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

18 Janvier 2023

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- *30+ years* of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- *20+ years* of Free and Open Source Software
- *10+ years* building and directing structures for the common good

1999 *DemoLinux* – first live GNU/Linux distro

2007 *Free Software Thematic Group*
150 members 40 projects 200Me

2008 *Mancoosi project* www.mancoosi.org

2010 *IRILL* www.irill.org

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

2021 *EOSC Task Force on Infrastructures for Software*, European Union

Outline

Why Open Science?

Open Science (Second National Plan for Open Science, France, 2021)

Unhindered dissemination of results, methods and products from scientific research.
It draws on *the opportunity provided by recent digital progress* to develop *open access* to *publications* and – as much as possible – *data*, *source code* and *research methods*.

Jean-Eric Paquet (EU DGRI, on the objective of Open Science)

"Increase *scientific quality*, the *pace of discovery and technological development*, as well as *societal trust in science*."

Mariya Gabriel (EU Commissionneer for Research)

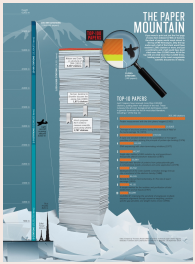
The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results. No nation, no country can tackle any of these global challenges alone.*

Yuval Noah Harari (on COVID 19)

"The real antidote [to epidemic] is scientific knowledge *and* global cooperation."

Software is a pillar of Open Science

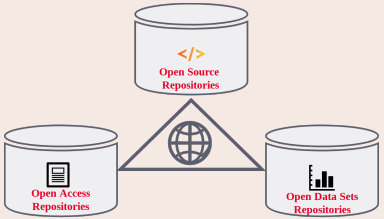
Software powers modern research



[…] software […] essential in their fields.
Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data
Christine Borgman, Paris, 2018

A key pillar: software (source code)



The links in the picture are important

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*

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     CODE500         # ASTRONAUT:    PLEASE CRANK THE
                TC      BANKCALL        #               SILLY THING AROUND
                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 ...
                CADR    BURNBABY
```

Quake III source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 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 * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( 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."

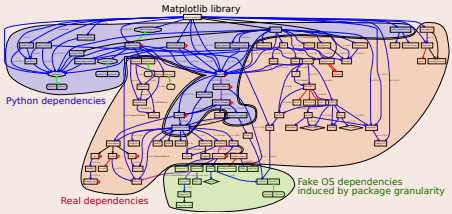
Software source code is *not* data

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*



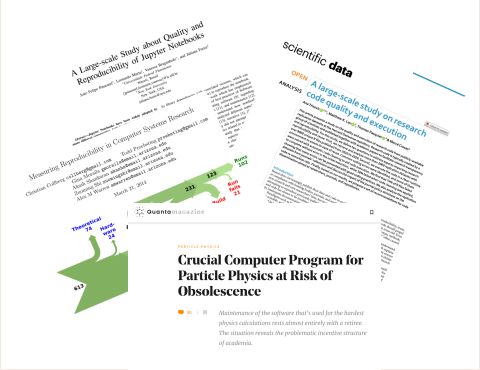
The human side

design, algorithm, code, test, documentation, community, funding

and so many more facets …

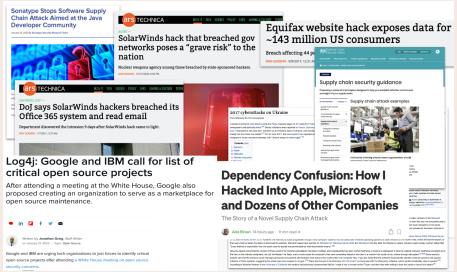
How are we managing our software ?

Reproducibility, maintenance in Academia



(articles: here, here, here and here)

Security, integrity, traceability in Industry



Can they track the software that they

- ship, use, acquire
- has that bug or vulnerability

awareness is raising at the level of public policy

Outline

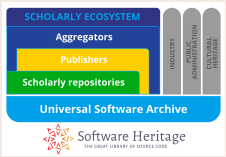
International highlights

Paris Call on Software Source code (2019, UNESCO)

40 international experts call to *"promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, […] recognising in the careers of academics their contributions to high quality software development, in all their forms"*

☞ Open Source in UNESCO recommendations for Open Science, 2021

Software in the EOSC



2020 EOSC SIRS connect scholarly ecosystem via Software Heritage
2021 EOSC Task Force on Infrastructures for Research Software
2022 FAIRCORE4EOSC project WP6 implements SIRS report
2023 INFRAEOSC call on quality of scientific software

And much more

Software track in OSEC 2022, Software working group launched in Science Europe, DFG adds software to model CV (9/22), NASA unveils Open Science policy (12/22), …

French National plan for Open Science, 2021-2024

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION

Liberté
Égalité
Fraternité

SECOND FRENCH PLAN FOR OPEN SCIENCE

Generalising open science in France 2021-2024

1

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
DE LA RECHERCHE
ET DE L'INNOVATION

Second French Plan for Open Science

Launch on 6 July 2021 by Frédérique Vidal, Minister for Higher Education, Research and Innovation

- Multiplying the **levers for change** in order to **generalise open science practices**
- Structuring the **policy for opening up or sharing research data**
- New commitments to the **opening of source code** produced by research
- **European and international inclusion** in the context of the French Presidency of the European Union
- **Disciplinary and thematic variations**: open science policies must be adapted to disciplinary specificities

2

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
DE LA RECHERCHE
ET DE L'INNOVATION

Path Three :

Opening up and promoting source code produced by research

7 **Recognize and support the dissemination under an open source license of software** produced by publicly funded research programmes

8 **Highlight the production of source code** from higher education, research and innovation

9 Define and promote an **open source software policy**

« The opening of software source code is a major challenge for the **reproducibility** of scientific results. »

« Distribution of software products under **open source licence** will be preferred. »

3

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
DE LA RECHERCHE
ET DE L'INNOVATION

Define and promote an open source software policy

- Produce a **National Charter for Open Source Software** coming from higher education, research and innovation
- Develop the **link between data and software** through a network of **Chief Data Officers** in the various universities and research performing organisations.
- Develop the **economic models of open source software** and make them known within commercialization services
- **Support Software Heritage** and recommend it for the archiving and referencing of source code

Recognise source code as a contribution to research

- Create an **open source research software prize**
- **Provide greater recognition** for software production in the career of researchers, research support staff

Build an ecosystem that connects code, data and publications

- Develop **proper coordination** between software forges, open publication archives, data repositories and the scientific publishing sector.

4

Software College in the CoSO

Five action lines (see details online)

- Identifying and highlighting research software production
- Technical and social tools and best practices
- Valorization and sustainability
- Liaison and animation at national, European, and international levels
- Recognition and careers

Leveraging experience and connections

- Open Source thematic group in Systematic (since 2007, more on demand)
- Collaboration with DINUM, Eclipse Foundation, OW2, …

Composition

Chairs: Roberto Di Cosmo and François Pellegrini
20+ active members from a broad panel of institutions and fields

Outline

What is at stake

ARDC

- Archive for retrieval (*reproducibility*)
- Reference for identification (*reproducibility*)
- Describe for discovery and reuse
- Cite/Credit for credit and evaluation

Before ARDC

- Development practices and tools (VCS, build system, test suites, CI, code quality, …)
- Opening up towards a community (documentation, organization, communication)

Need training, tooling, infrastructures, best practices

Beyond ARDC

- Policies (dissemination, reuse, careers, …)
- Sustainability (legal, financial, etc.)
- Technology transfer
- Advanced technologies and tools (quality, traceability, etc.)

Here we will focus on ARDC

Some popular approaches, and why they do not fit the bill

A - Since the ~~1970's~~ 1990's

.zip or .tar file on:

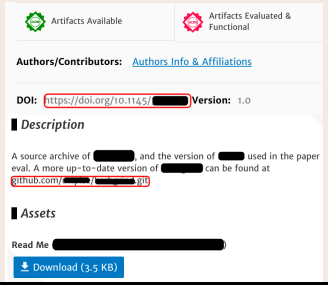
- ~~ftp server~~ (e.g. gnu)
- web page (e.g. myself)
- document archive (+ DOI sample)

B - Since the 2000's

Rely on *software forges*

- institutional/project (e.g. myself)
- free commercial ones: BitBucket, GitHub, GitLab, … (e.g. myself)

C: a mix of the two



Can get no satisfaction…

A *Poor user experience*

B *No preservation guarantee*

C Can do *so much* better

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)

Big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases *250.000+* repositories (including research software)
- summer 2022: GitLab.com considers erasing all projects that are inactive for a year

In Academia too!

- 2021: Inria's old gforge is unplugged… breaks the Opam build chain for OCaml

We need a universal archive of software source code: now we have one!

Outline

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

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 and share all software source code

Research infrastructure

enable analysis of all software source code

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

Bronze sponsors

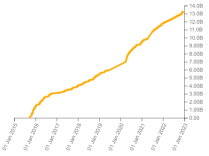
The largest software archive, a shared infrastructure

Cultural Heritage | Industry | Research | Public Administration

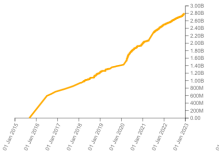



Software Heritage

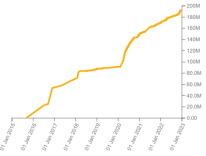
Source files
13,338,879,609



Commits
2,801,474,518



Projects
193,279,669



Directories
10,905,976,959

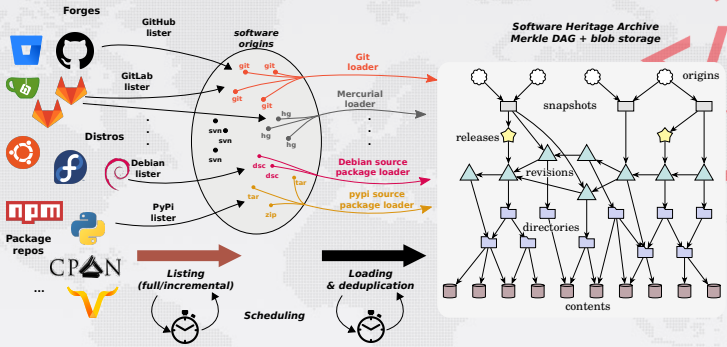
Authors
51,691,686

Releases
35,938,300

Bitbucket	git	R
1,925,997 origins	21,603 origins	21,113 origins
debian		GitHub
128,719 origins	5,947 origins	137,564,899 origins
GitLab	Guix	GNU
3,982,586 origins	12,032 origins	354 origins
heptapod	launchpad	Maven
1,068 origins	329,908 origins	93,738 origins
NixOS	npm	
12,032 origins	1,799,296 origins	4,083 origins
Phabricator	python	SOURCEFORGE
192 origins	410,582 origins	308,990 origins

Outline

Address common Open Science and Open Source needs: archival



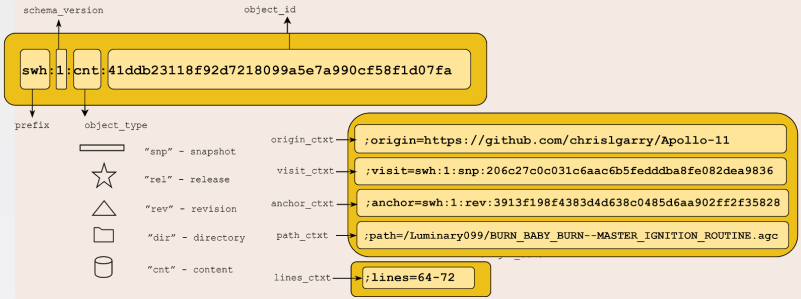
Global development history permanently archived in a uniform data model

- over 13 billion unique source files from over 200 million software projects
- ~1PB (uncompressed) blobs, ~25 B nodes, ~350 B edges

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs



25+B
intrinsic,
decentralised,
cryptographic

Full fledged *source code references* for traceability, integrity and reproducibility

- Linux Foundation SPDX 2.2
- IANA-registered "swh:"
- WikiData property P6138

Examples: Apollo 11 AGC excerpt, Quake III rsqrt
Guidelines available, see the HOWTO

Breaking news: standardisation, see swhid.org

Archival and reference, hands on

A quick tour as a user

- designed for source code: Browse (e.g. Apollo 11 excerpt, see also Apollo 11 blog post) like on a developer platform, not a document archive!
- reference source code: all granularities, using SWHIDs (full specification available online)
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment
  - guidelines and a full article
  - SWHIDs *guarantee integrity* like in *blockchains*
    demo if time left:
    1. download a version of a project for a given SWHID
    2. compute locally the SWHID with swh-identify
    3. check that the computed id match the given one

Archival and reference, hands on, cont'd

Getting software archived

- automated harvesting: over 200 million software origins, your researchers' work may already be there (actually, here)!
- universal archive: *all* source code from *all* platforms (BitBucket, GitHub, GitLab, your own forge, etc.)
  - trigger archival of *any code* in one click with the updateswh browser extension
  - use webhooks to automatically archive *your code* (a GitHub action is available too)
  - journals, libraries, open access portals may *deposit sourcecode and metadata*
    - Example article from IPOL
    - Example article from eLife

A look at some adoption indicators

From Melissa Harrison's OSEC 2022 talk

What are they "referencing"?

source	n	percentage
Not available	2868	46.22
GitHub	1151	18.55
software heritage	387	6.24
zenodo	142	2.29
r package	70	1.13
cran	56	0.90
r package version	54	0.87
gitlab	35	0.56

- 6205 "software" references identified
- Top 8 listed, then long tail of 1055 other sites – 932 are unique "source"

Upcoming on replicabilitystamp.org (please do not spoil :-))

HAL+SWH in the Open Science software booklet

Funding agencies recommendations ANR 2023 guidelines (p. 17)

Enfin, conformément au 2ème Plan national pour la science ouverte, L'ANR recommande que les logiciels développés durant le projet soient mis à disposition sous une licence libre[30] et que les codes sources soient stockés dans l'archive Software Heritage[31] en indiquant la référence au financement ANR.

Outline

Describe, cite, credit

Software metadata: codemeta.json

- example from Parmap, created using the Codemeta generator

Integration with the HAL national french open access archive

- Curated deposit: metadata quality due to moderation
  - all pieces of the puzzle together: one researcher does all the steps (Parmap)
- export of citation information for biblatex-software
- examples: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, ...
- generation of reports, cv, web pages: for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools

Bottomline

Software Heritage + a *curated* metadata repository allows to address all needs …

- *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

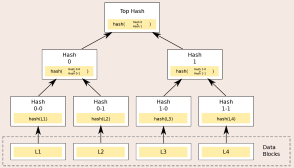
Outline

A few words about what we did not see

The *graph* of Software Development

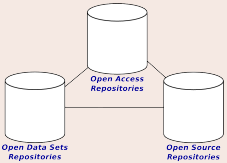
All software development in a single graph …

The *blockchain* of Software Development



… a single Merkle graph!

A *pillar* of Open Science



Reference archive of Research Software

Reference platform for *Big Code*

A single, uniform data structure

Outline

Call to action: best practices for ARDC are available… today!

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)

- train students and colleagues
- engage journals, conferences, learned societies

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

Questions?

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. Zacchiroli, *Building the Universal Archive of Source Code*
CACM, October 2018 (10.1145/3183558)