# Software and Source Code for Open Science

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

# Outline



B_BLOCK

# 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 Software *Source Code*

## 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."*

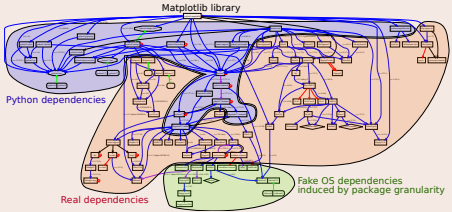
# Source code is *special* (software is *not* data)

## 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*



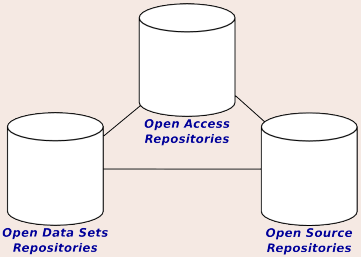

## Precious, endangered *executable* and *human readable* knowledge

key people passing away, platforms (GoogleCode, Gitorious, etc.) closing down …

no organised effort to catalog and archive it

# Software Source code: a pillar of Open Science

## Three pillars of Open Science




## A plurality of needs

**Researcher**

- archive and reference software used in articles
- find useful software
- get credit for developed software
- verify/reproduce/improve results

**Laboratory/team** track software contributions

- produce reports / web page

**Research Organization** know its software assets

- technology transfer
- impact metrics

# What is at stake: ARDC in increasing order of difficulty

## 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

make 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*!)

We need an infrastructure *designed for* software source code: *now we have one!*

# What is at stake: before ARDC

## Development practices

- version control system
- key metadata information (README, AUTHORS, LICENCE, etc.)
- build system
- test suites
- ...

## Opening up

- documentation
- community building
- ...

See Liam's talk!

# What is at stake: beyond ARDC

## Sustainability, technology transfer

Organisational schemas, legal tools, economic models, processes and policies to ensure research software can be maintained and sustained over time, maybe in connection with industry

## Evaluation (funding, careers, etc.)

- beware of *naive software citation counting*, prefer qualitative evaluation (see the French National Prize)
- identify *roles* in software projects, see:

  P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M.-S. Hacid, A. Legrand and N. Rougier *Attributing and referencing (research) software: Best practices and outlook from Inria*, CiSE 2020 (10.1109/MCSE.2019.2949413)

## Regulations are coming

software management plans, licensing, metadata and identification standards

# 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 all software source code

### Research infrastructure

enable analysis of all software source code

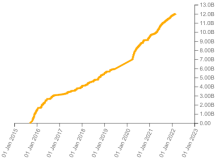
# The largest software archive, a shared infrastructure

**Cultural Heritage** | **Industry** | **Research** | **Public Administration**

## Software Heritage

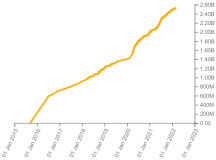
**Source files**

12,032,627,304



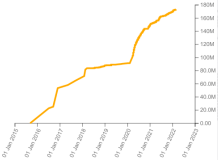

**Commits**

2,536,918,821




**Projects**

173,242,749




**Directories**

9,946,192,395

**Authors**

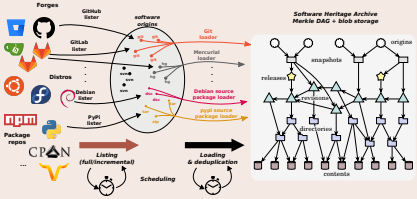
47,334,620

**Releases**

31,763,605

# Addressing the four needs (see ICMS 2020 for details)

## Archive (10B+ files, 150M+ 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

# Outline

# Focus on Archive and Reference

- Browse the archive
- Trigger archival of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- Example use in a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Cite software with the biblatex-software style from CTAN
- Example use in a research article: extensive use of SWHIDs in a replication experiment
- Example in a real journal: an article from IPOL
- Supporting reproducible builds: Guix and Nix

# Recent news, and a lesson to be learned

## Saving 250.000 endangered repositories…

- summer 2019: BitBucket announce Mercurial VCS phase out
- fall 2019: Software Heritage teams up with Octobus (funded by NLNet, thanks!)
- july 2020: BitBucket erases *250.000* repositories
- august 2020: bitbucket-archive.softwareheritage.org is live

## … preserving the web of knowledge (Tweet is here )


**Bottomline**

*explicit deposit* is important, …

… and we must promote it…

… but will never be enough.

*(think also of all software dependencies!)*

# Outline

# An international, non profit initiative built for the long term

## 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

# Growing adoption of SWH in Academia (selection)

## HAL software curated deposit workflow

*Curated Archiving of Research Software Artifacts*
International Journal of Digical Curation, 2020

## Reference archive for swmath.org

swMATH
an information service for mathematical software

See *code* links, e.g.
SemiPar package

## IPOL (image processing)

- archive (deposit)
- reference
- BibLaTeX

## eLife (life sciences)

- archive (save code now)
- reference

## JTCAM (mechanics)

- instructions for authors
- biblatex-software in journal LaTeX class

## Policy: France

NATIONAL PLAN FOR OPEN SCIENCE

*National Plan for Open Science*

## Policy: Europe


*EOSC SIRS report*

- SWHIDs
- archive

## Guidelines


- summary
- ICMS 2020

# Getting onboard

## You can foster adoption and best practices

- archive source code in Software Heritage, reference it with SWHID,
- cite using biblatex-software

# Questions?

## References


*Open Science European Conference, (OSEC 2022)*, online recordings at osec2022.eu

EOSC SIRS Task Force, *Scholarly Infrastructures for Research Software*, 2020, (10.2777/28598)

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

P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M.-S. Hacid, A. Legrand and N. Rougier *Attributing and referencing (research) software: Best practices and outlook from Inria*, CiSE 2020 (10.1109/MCSE.2019.2949413) (hal-02135891)

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