Why we must preserve the world's software history

and how Software Heritage tackles this challenge

Roberto Di Cosmo
Inria and Université de Paris

5 October 2021
CodeFest

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

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

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

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."

Calling for preservation: UNESCO

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 meet in Paris …

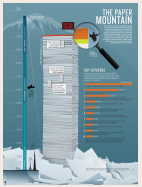
The call is published on Feb 2019

"[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive"

https://en.unesco.org/foss/paris-call-software-source-code

Source code history for Open Science

Software is everywhere in modern research



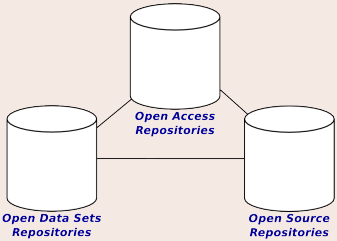
[…] software […] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data

Christine Borgman, Paris, 2018

Open Science: three pillars



Nota bene

The links in the picture are essential

Preserving the history of source code is important for *reproducibility*

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

Fragile

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- business-driven code loss (e.g., Gitorious, Google Code, Bitbucket)
- data rot: physical media with legacy software decay

If a website disappears you go to the Internet Archive…

where do you go if (a repository on) GitHub or GitLab goes away?
and what about all the landmark legacy source code that is rotting away?

We are at a turning point

Looking at the past

- a lot of old software misplaced, lost, or behind barriers, but…
- most early creators are still here, and willing to share
- urgent to collect their knowledge

Only a few years left.

Looking at the future

- software development and use skyrockets: more programmers, and more code!
- essential to provide a universal platform for all the future software source code

Every year that goes by makes the problem worse.

it is urgent to take action!

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

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

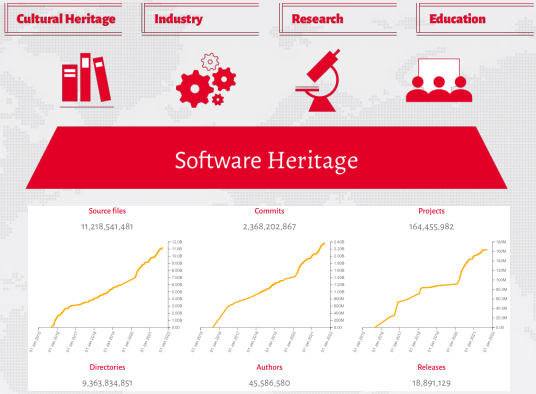
find and reference all software source code

Universal archive

preserve all software source code

Research infrastructure

enable analysis of all software source code



Technology

- transparency and FOSS
- replicas all the way down

Content (billions!)

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- multi-stakeholder

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

A walkthrough

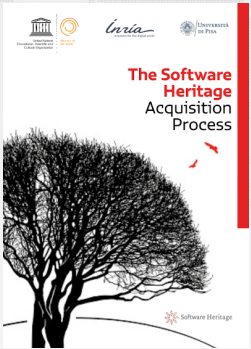
- Browse the archive
- Trigger archival of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- The Apollo 11 AGC source code example
- Cite software with the biblatex-software style from CTAN
- Example use in a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Example in a journal: an article from IPOL
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Rescue landmark legacy software, see the SWHAP process with UNESCO

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

Paris Call on Software Source Code

"[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive"

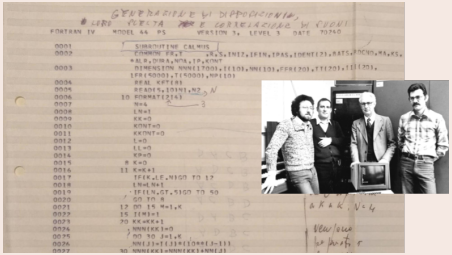


- Rescue Legacy Software from different media
- Curate the code
  - reconstruct the development history
    - *Software Heritage - GitHub* work on fixing git
  - collect the metadata
- Archive in Software Heritage

UNESCO, UniPi and Software Heritage collaboration worldwide scope

An example: TAUmus, from Pisa (70's)

Electronic music in Pisa: group led by the late M° P. Grossi



- Control code of the music synthesizer TAU2
- FORTRAN II, TAUmus command language
- Istituto di Elaborazione dell'Informazione CNR
- e.g. Le Sacre du Printemps (ABSTRACT)

Stay tuned

great news for software stories in a few months...

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

Focus on Academia: growing adoption (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 Open Science

Policy: Europe

EOSC SIRS report

- SWHIDs
- archive

Guidelines

- summary
- ICMS 2020

Outline

1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria — INVENTEURS DU MONDE NUMÉRIQUE

Platinum sponsors

cnrs · intel · Microsoft · HUAWEI · SOCIETE GENERALE

Gold sponsors

openinventionnetwork · Université de Paris · SORBONNE UNIVERSITÉ

Silver sponsors

CAST · GitHub · Google · Università di Pisa · vmware

Bronze sponsors

DANS · FOSSID

You may help!

Foster adoption and best practices

- archive and reference relevant source code (save code now, and deposit)
- use Software Heritage in research articles, journals, and books
- rescue and preserve landmark legacy source code with SWHAP

Engage with Software Heritage as an individual

- make a donation
- join the ambassador program, help raise awareness
- contribute to technical and scientific development

Engage with Software Heritage as an organization

- become a member/sponsor
- build a Software Heritage mirror (like ENEA is doing)

Questions?

Resources

newsletter https://www.softwareheritage.org/newsletter/

blog https://www.softwareheritage.org/blog/

archive https://archive.softwareheritage.org/

media, press, etc. https://annex.softwareheritage.org/

References (see https://www.softwareheritage.org/publications)

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

R. Di Cosmo, *Archiving and Referencing Source Code with Software Heritage*
ICMS 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)