Why we must preserve the world's software history, and how we can do it.

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris

16/03/2022
Convegno sul Software, Bologna 2022

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

Outline

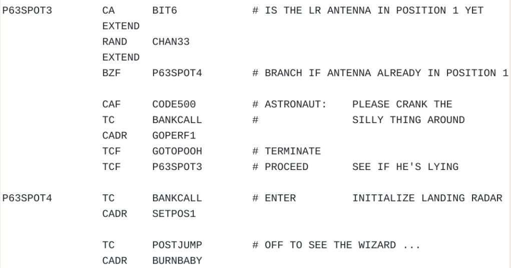
1. Introduction
2. Why we must preserve the history of Software Source Code
3. An Endangered Knowledge
4. How we can preserve our Software Heritage
5. Demo time!
6. Preserving our software commons: the past
7. Preserving our software commons: the present and the future
8. The road ahead

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)



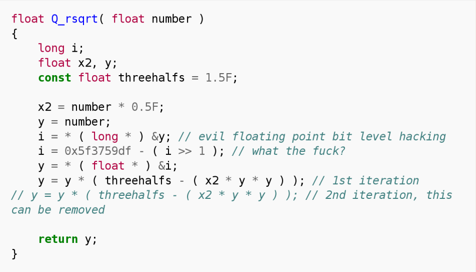
```
P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                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

United Nations Educational, Scientific and Cultural Organization

Memory of the World

Inria inventors for the digital world

PARIS CALL
SOFTWARE SOURCE CODE
AS HERITAGE FOR SUSTAINABLE DEVELOPMENT

Software Heritage

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

Calling for preservation: Donald Knuth and Len Shustek

Communications of the ACM, February 2021

"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."

Let's Not Dumb Down the History of Computer Science
Donald E. Knuth, Len Shustek
https://doi.org/10.1145/3442377

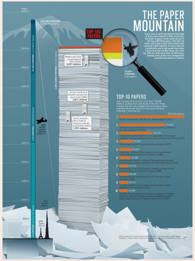
A unique opportunity

most of the creators are still here: we can talk to them!

but the clock is ticking...

Source code history for Open Science

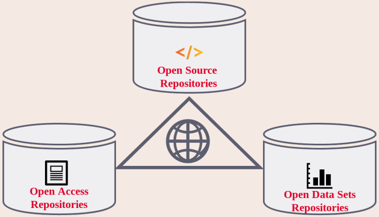
Software powers modern research



[...] 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

Missing pillar: software (source code)



The links in the picture are important

Nota Bene

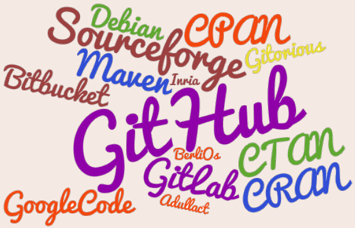
software may be a *tool*, a *research outcome* and a *research object*

access to the *source code* is essential!

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

Source code history for Security and Transparency

Where does reused software come from?



Do *you* know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
  - has that bug
  - has that vulnerability

KYSW: Know Your SoftWare

Like KYC in banking, KYSW is now essential all over IT…

Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

May 2021 POTUS Executive Order

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

damage
disaster
malicious
obsolete
deletion
reference
storage
media
aging
tear
attack
dependencies
dangling
wear
corruption
encryption
format

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 in a nutshell

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

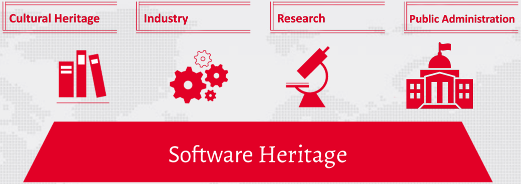
preserve all software source code

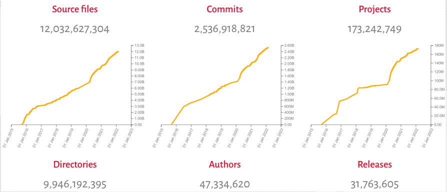
Research infrastructure

enable analysis of all software source code

Largest software archive, principled

http://bit.ly/swhpaper




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

Technology

- transparency and FOSS
- replicas all the way down

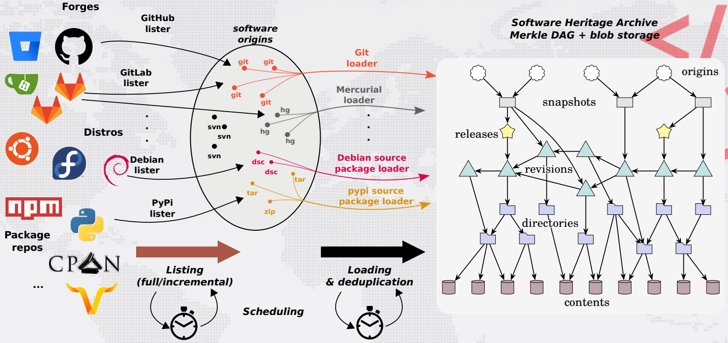
Content (billions!)

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- multi-stakeholder

A peek under the hood: a universal archive

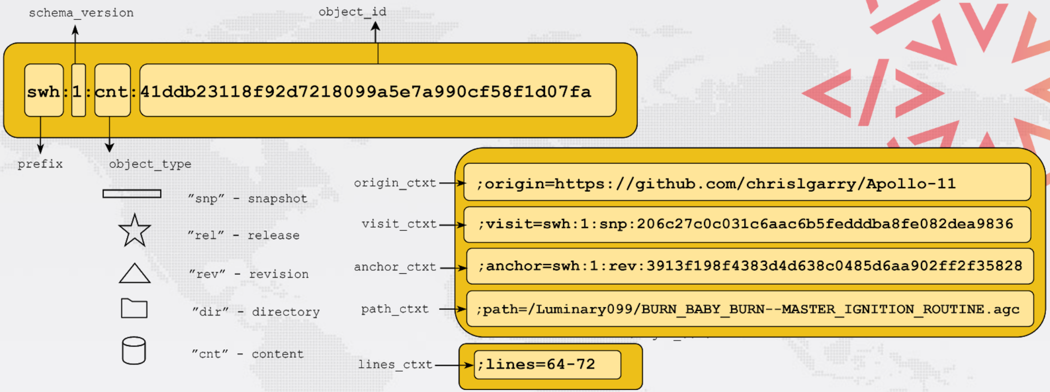


Global development history permanently archived in a *unique* git-like Merkle DAG

- ~400 TB (uncompressed) blobs, ~20 B nodes, ~300 B edges

Software Heritage *intrinsic* Identifiers (SWHID)

link to full docs



schema_version

object_id

swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa

prefix

object_type

"snp" - snapshot

"rel" - release

"rev" - revision

"dir" - directory

"cnt" - content

origin_ctxt → ;origin=https://github.com/chrislgarry/Apollo-11

visit_ctxt → ;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836

anchor_ctxt → ;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828

path_ctxt → ;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc

lines_ctxt → ;lines=64-72

An emerging standard

- in Linux Foundation's SPDX 2.2
- IANA registered, WikiData property P6138

Examples:

- Apollo 11 AGC excerpt,
- Quake III rsqrt

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

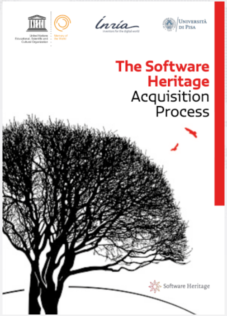
Outline

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

The *SWHAP* process, with UNESCO and University of Pisa

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”



The Software Heritage Acquisition Process

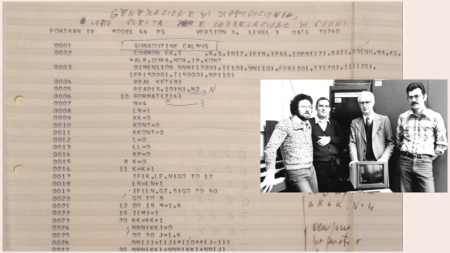
- Rescue Legacy Software from different media
- Curate the code
  - reconstruct the development history
    - *Software Heritage - GitHub* work on fixing git
  - collect the metadata
- Archive in Software Heritage

UNESCO, UniPi and Software Heritage collaboration

now we can build Software Stories!

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

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



GENERAZIONE DI DISPOSIZIONI,
LORO SCELTA ~~PER~~ E CORRELAZIONE DI SUONI

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 3   DATE  70240

          SUBROUTINE CALMUS
          COMMON FR,T      ,R,S,INIZ,IFIN,IPAS,IDENT(2),BATS,RDCND,MA,KS,
         *ALR,DURA,NOA,IP,KONT
          DIMENSION NNN(1700),I(10),NN(10),FFR(20),TT(20),III(20),
         1FR(5000),T(5000),NP(10)
          REAL KFT(8)
          READ(5,10)N1,N2 -> N
       10 FORMAT(2I4)
          N=4                 3
          LN=1
          KK=0
          KONT=0
          KKONT=0
          L=0
          LL=0
          KP=0
        8 K=0
       11 K=K+1
          IF(K.LE.N)GO TO 12
          LN=LN+1
          IF(LN.GT.5)GO TO 50
          GO TO 8
       12 DO 15 M=1,K
       15 I(M)=1
       20 KK=KK+1
          NNN(KK)=0
          DO 30 J=1,K
          NN(J)=I(J)*(10**(J-1))
       30 NNN(KK)=NNN(KK)+NN(J)
```

a k a k ; N=4

- 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)

New!

see this live on the Software Stories website

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

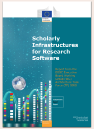
JTCAM (mechanics)

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

Policy: France

National Plan for Open Science

Policy: Europe


Scholarly Infrastructures for Research Software

EOSC SIRS report

- SWHIDs
- archive

Guidelines

Software Heritage
1 Prepare your public repository
README, AUTHORS & LICENSE files
2 Save your code
http://save.softwareheritage.org/
3 Reference your work
(full repository, specific version or code fragment)

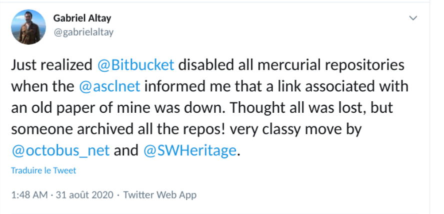
- summary
- ICMS 2020

Recent preservation news

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 (original tweet is here)



Gabriel Altay
@gabrielaltay

Just realized @Bitbucket disabled all mercurial repositories when the @asclnet informed me that a link associated with an old paper of mine was down. Thought all was lost, but someone archived all the repos! very classy move by @octobus_net and @SWHeritage.

Traduire le Tweet

1:48 AM · 31 août 2020 · Twitter Web App

Bottomline

explicit deposit is important, …

… and we must promote it…

… but will never be enough.

(think also of all software dependencies!)

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

An international, non profit initiative built for the long term

Sharing the vision

UNESCO

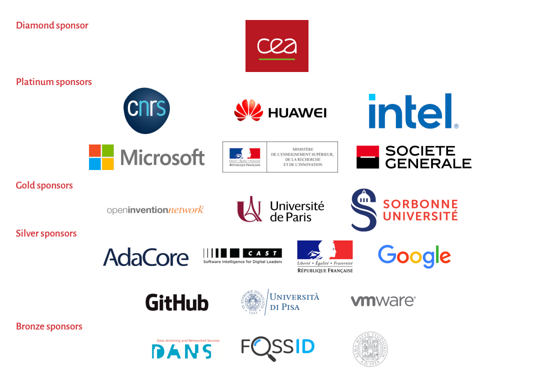
United Nations
Educational, Scientific and
Cultural Organization

IEEE Software
GitLab
ENGINEERING
eclipse
OW2
fsfe
nlnet foundation
SIGSOFT
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
INAPI
THE LINUX FOUNDATION
ADULLACT
INFORMATICS EUROPE
Computer History Museum
software freedom conservancy
Software Freedom Law Center
AdaCore
The GNAT Pro Company
gandi.net
TDF
FREE SOFTWARE FOUNDATION
creative commons
SIF
Société Informatique de France
open source initiative
openinventionnetwork

And many more ...
www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria



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 and Software Stories

Engage with Software Heritage as an individual

- 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)
- contribute to the preservation mission

Thank you!

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)