Software Heritage: key infrastructure for Open Science and Open Source

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

25 November 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

Software and Source code: a reminder

“The source code for a work means the preferred form of the work for making modifications to it."

GPL Licence

Software and Source code: a reminder

"The source code for a work means the preferred form of the work for making modifications to it." *GPL Licence*

```
Hello World
```

Software and Source code: a reminder

"The source code for a work means the preferred form of the work for making modifications to it." — *GPL Licence*

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Software and Source code: a reminder

"The source code for a work means the preferred form of the work for making modifications to it." — *GPL Licence*

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Program (source code)

```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

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."

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
```

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;
}
```

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."

~ 50 years, a lightning fast growth

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

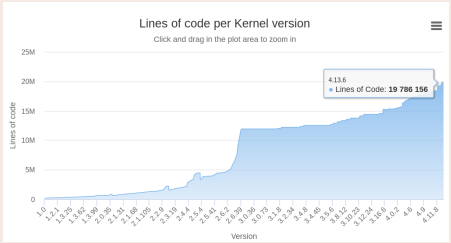
~ 50 years, a lightning fast growth

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel



… now in your pockets!

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,...

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,...

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

First 15 years: 1984-... The early revolution

focus *freedom* (users, developers)

keyword free software (individual)

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,…

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

First 15 years: 1984-… The early revolution

focus *freedom* (users, developers)

keyword free software (individual)

1999-… Progressive industry adoption

focus software quality, reduced cost

keyword open source (entities)

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,…

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

First 15 years: 1984-… The early revolution

focus *freedom* (users, developers)

keyword free software (individual)

1999-… Progressive industry adoption

focus software quality, reduced cost

keyword open source (entities)

2010-… Ecosystems, strategic alignment

focus organisation, foundations

keyword governance and funding

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,...

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

First 15 years: 1984-... The early revolution

focus *freedom* (users, developers)

keyword free software (individual)

1999-... Progressive industry adoption

focus software quality, reduced cost

keyword open source (entities)

2010-... Ecosystems, strategic alignment

focus organisation, foundations

keyword governance and funding

2015-... Industry consolidation

focus mergers and acquisitions

keyword control

Free Software: 40 years, 4 layers, in a nutshell

Free Software, AKA: *Open Source*, FOSS, FLOSS,…

Software that offers to *its users* the freedom to:

- use the software
- study and adapt the software
- distribute software copies
- distribute modified copies

First 15 years: 1984-… The early revolution

focus *freedom* (users, developers)

keyword free software (individual)

1999-… Progressive industry adoption

focus software quality, reduced cost

keyword open source (entities)

2010-… Ecosystems, strategic alignment

focus organisation, foundations

keyword governance and funding

2015-… Industry consolidation

focus mergers and acquisitions

keyword control

we faced many common issues *way before Open Science was on the radar*

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*.

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*."

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.*

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."

The scientific method…

The experimental method

- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- elaborate a *theory*

And then we reproduce and verify.

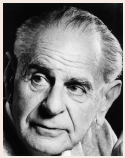
The scientific method…

The experimental method

- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- elaborate a *theory*

And then we reproduce and verify.

Reproducibility is the key

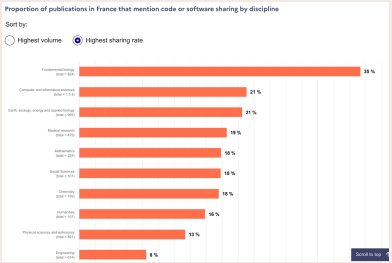


non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

Software is a pillar of Open Science

Software powers modern research

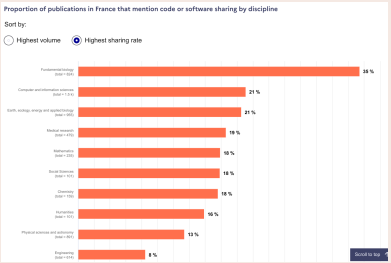


Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

Software is a pillar of Open Science

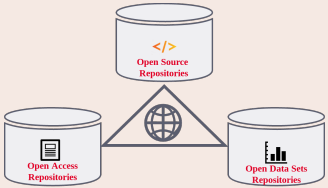
Software powers modern research



Over 20% of articles across all disciplines share software

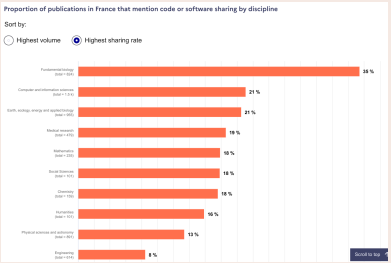
2023 French Open Science Monitor

Key pillar: software



Software is a pillar of Open Science

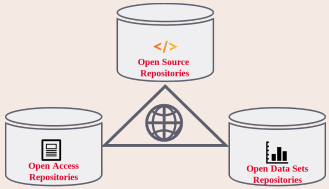
Software powers modern research



Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

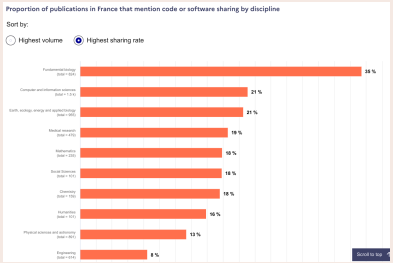
Key pillar: software



Links are important

Software is a pillar of Open Science

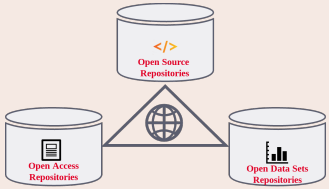
Software powers modern research



Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

Key pillar: software



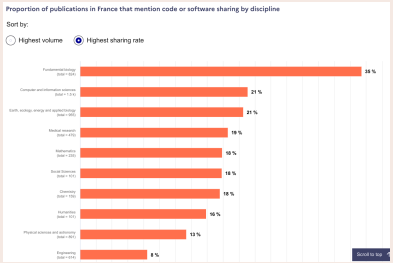
Links are important

Nota Bene

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

Software is a pillar of Open Science

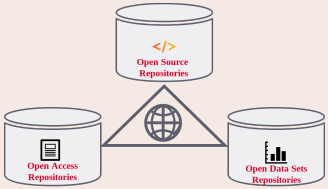
Software powers modern research



Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

Key pillar: software



Links are important

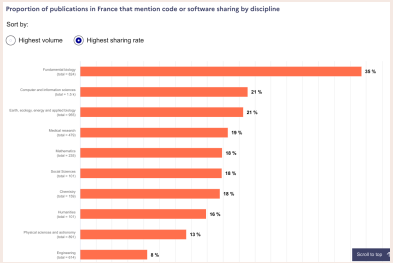
Nota Bene

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

access to the *source code* is essential!

Software is a pillar of Open Science

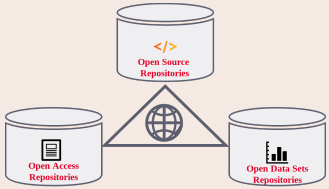
Software powers modern research



Over 20% of articles across all disciplines share software

2023 French Open Science Monitor

Key pillar: software



Links 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*

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

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

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.)

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

How (not) to preserve and share research software

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

.zip or .tar file on:

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

How (not) to preserve and share research software

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

.zip or .tar file on:

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

B - Since the 2000's

Rely on *software forges*

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

How (not) to preserve and share research software

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

.zip or .tar file on:

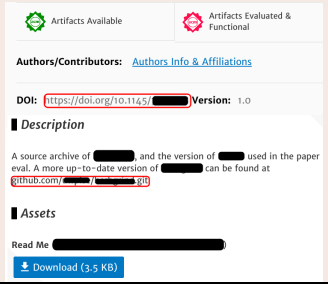
- ~~ftp server~~ (e.g. gnu)
- web page (example)
- document archive (+ DOI sample)

B - Since the 2000's

Rely on *software forges*

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

C: a mix of the two



How (not) to preserve and share research software

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

.zip or .tar file on:

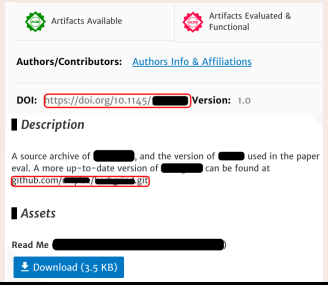
- ~~ftp server~~ (e.g. gnu)
- web page (example)
- document archive (+ DOI sample)

B - Since the 2000's

Rely on *software forges*

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

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)

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

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

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:

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

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

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

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

Sharing the vision

UNESCO
United Nations Educational, Scientific and Cultural Organization

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

Donors, members, sponsors

Inria

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

Bronze sponsors

The largest software archive, a shared infrastructure

One infrastructure
open and shared

The largest software archive, a shared infrastructure

One infrastructure
open and shared

The largest archive ever built

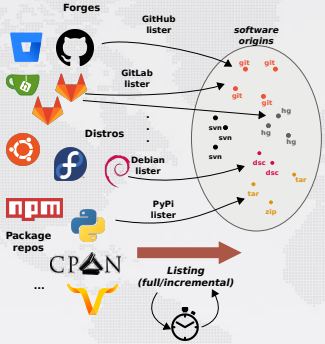
The largest software archive, a shared infrastructure

One infrastructure
open and shared

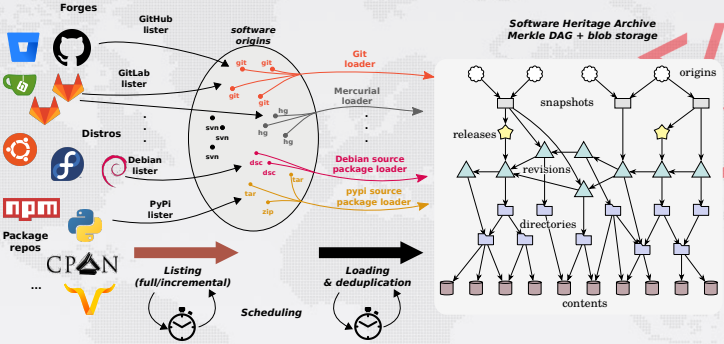
The largest archive ever built

Bitbucket	git	
2,012,133 origins	19,494 origins	21,486 origins
debian		GitHub
129,217 origins	6,424 origins	152,282,093 origins
GitLab	Guix	GNU
3,989,638 origins	12,451 origins	354 origins
heptapod	launchpad	Maven
1,096 origins	356,873 origins	93,710 origins
NixOS	npm	
12,451 origins	1,799,296 origins	4,083 origins
Phabricator	python	SOURCEFORGE
185 origins	427,135 origins	308,970 origins

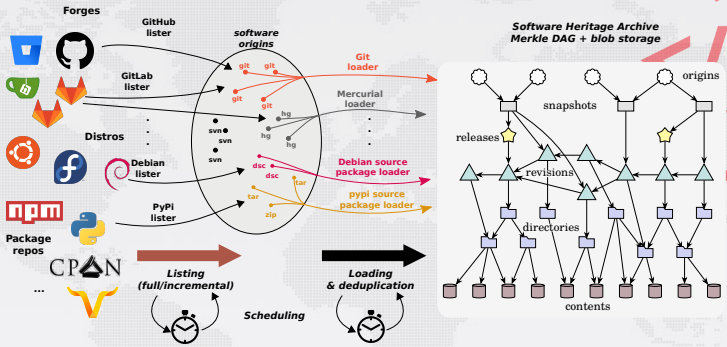
Address common Open Science and Open Source needs: archival



Address common Open Science and Open Source needs: archival



Address common Open Science and Open Source needs: archival



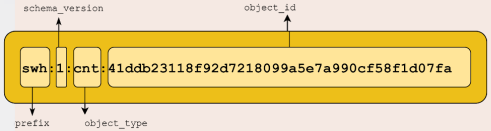
Global development history permanently archived in a uniform data model

- over 16 billion unique source files from over 260 million software projects
- ~1.5PB (compressed) blobs, ~35 B nodes, ~500 B edges

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs

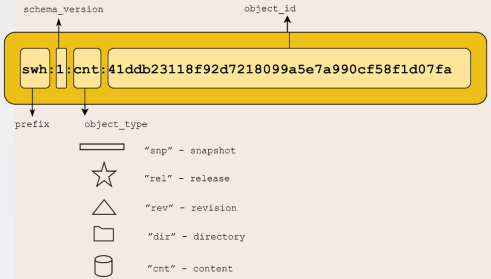


35+B
intrinsic,
decentralised,
cryptographic

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs



35+B
intrinsic,
decentralised,
cryptographic

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs

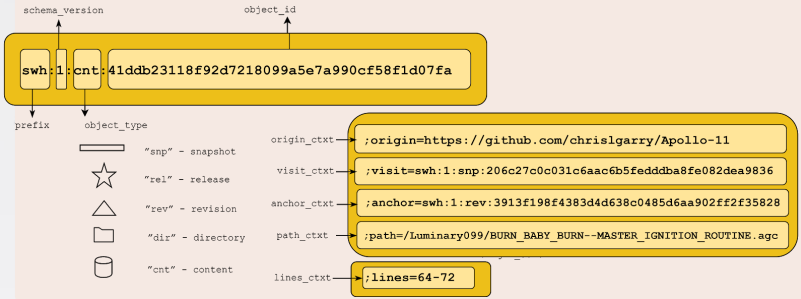
35+B

intrinsic, decentralised, cryptographic

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs



35+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

Outline

A walkthrough

- Browse and Reference (e.g. Apollo 11 [excerpt], your work may be already there !)
- Trigger archival, use the updateswh browser extension, configure the webhooks
- Cite with biblatex-software (CTAN, Overleaf ACMART template)
- Describe with Codemeta (use codemeta generator)
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, ...
- Extracting all the software products for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- Example with Parmap: devel on Github, archive in SWH, curated deposit in HAL
- Example research articles:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment

Outline

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 …

Calling for preservation: UNESCO

UNESCO, Inria, Software Heritage invite

40 international experts meet in Paris …

The call is published on Feb 2019

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 …

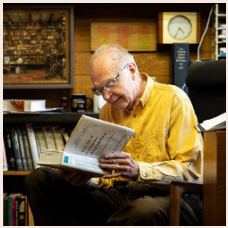
The call is published on Feb 2019

"[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

Calling for preservation: Donald Knuth and Len Shustek

Communications of the ACM, February 2021

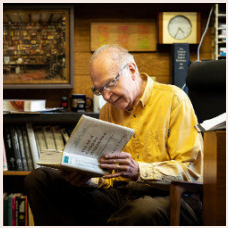


"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."

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

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!

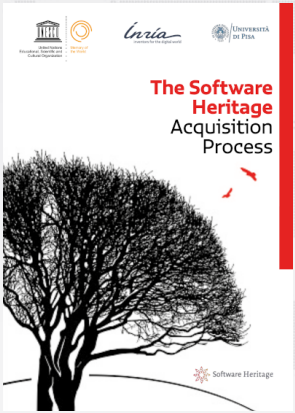
but the clock is ticking…

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"

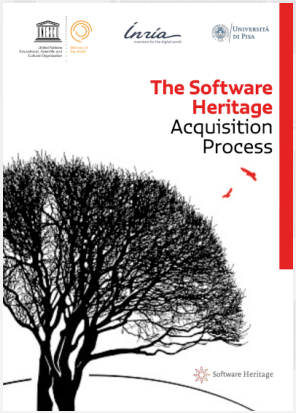
Paris Call on Software Source Code

"[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"



Paris Call on Software Source Code

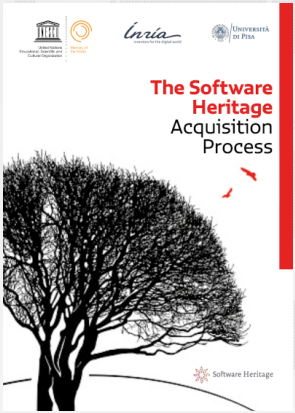
"[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive"



- Rescue Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported

Paris Call on Software Source Code

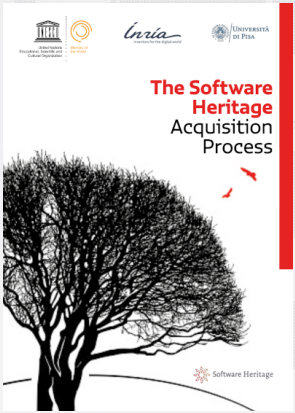
"[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive"



- Rescue Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported
- Curate the code
  - reconstructing the development history
  - collecting metadata

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
  - physical
  - digital
    - legacy / unsupported
    - recent / supported
- Curate the code
  - reconstructing the development history
  - collecting metadata
- And illustrate with dedicated presentations

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"

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"

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"

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history

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"

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- Preserve and Present all this material

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"

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- Preserve and Present all this material
- Share process and tools (all open source!)
  - with museums, archives and all interested parties

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"

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- Preserve and Present all this material
- Share process and tools (all open source!)
  - with museums, archives and all interested parties

see this live on the Software Stories website, and get *the guide*

Outline

Hold your breath!

- large scale *compression* of the archive
- large scale, efficient, *search* in the archive
- … and much more

Outline

Software Heritage: a shared open infrastructure

Software Heritage offers

- archival of all public source code
- reference of all public source code
- sharing cost with other partners
- standards based approach

Software Heritage: a shared open infrastructure

Software Heritage offers

- archival of all public source code
- reference of all public source code
- sharing cost with other partners
- standards based approach

Software Heritage is

- vendor neutral
- open source
- worldwide, long term
- born and based in the EU

Software Heritage: a shared open infrastructure

Software Heritage offers

- archival of all public source code
- reference of all public source code
- sharing cost with other partners
- standards based approach

Software Heritage is

- vendor neutral
- open source
- worldwide, long term
- born and based in the EU

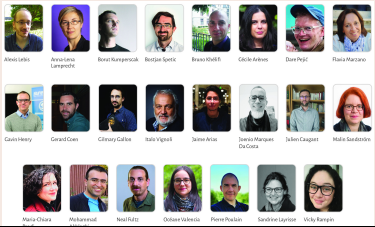
A growing and active community

Team

A growing and active community

Team

Ambassadors



A growing and active community

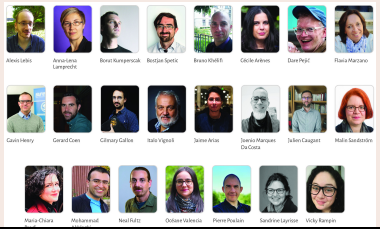
Team

Contributors to the platform

```
13:21:32    <seirl> from last time i ran it? it very likely is
13:21:47    <seirl> we had a x2 on the edges in a single year
13:23:14  <vlorentz> ah
13:53:44      <zack> seirl: I think i was remembering the LLP time on granet rather than the one (on the previous
                     graph) on the big mem telecom machine
13:54:01      <zack> wasn't it something like 10-14 days (on granet)?
13:55:11      <seirl> zack: it depends on the number of weights you use
13:55:23      <seirl> i had something like that to do the parameter sweep
13:55:31      <seirl> but then i settled on a few good gamma values
13:55:44      <seirl> and afterwards it was only ever ~3-4 days
14:02:57      <zack> ok
15:19:35    <jelmer> vlorentz: when is jenkins meant to kick in ? I didn't think the CI would mean you pasting test
                     results in comments :P
15:19:59    <jelmer> alternatively, I could try to get it working locally - for some reason tox doesn't run here,
                     complainining it can't find swh.scheduler
15:20:48  <vlorentz> jenkins is down until tomorrow evening (paris time)
15:20:59  <vlorentz> bad day for submitting your code :D
15:21:18  <vlorentz> er yeah I just fixed that issue
15:21:31  <vlorentz> but the fixed swh.scheduler is not pushed to pypi because jenkins
15:23:25    <jelmer> ah
15:23:40  <vlorentz> in the meantime, you can change apply this patch: https://gitlab.softwareheritage.org/-/
                     snippets/1546
15:23:44  <vlorentz> as an ugly workaround
15:24:13  <vlorentz> actually, just adding "pytest-postgresql < 4.0.0" should do it
15:25:00  <vlorentz> when jenkins is back online I'll push a new release of swh-scheduler without the missing
                     dependency on pytest-postgresql
```

Nicks: Alphare, ames, amj, amoeba, ar-jan, bchauvet[m], cmatrix[m], dan-, deesix, ericson2314, franckbret1, guerby, Guest92, haltode2, hplar, jayeshv, jelmer, KShivendu, landre[m], marmoute

Ambassadors

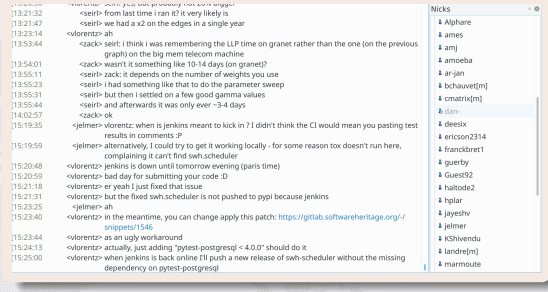


Ambassadors

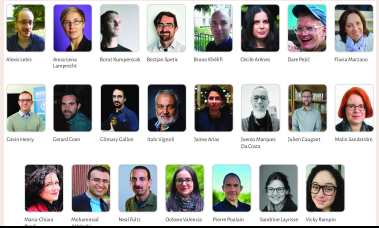
A growing and active community

Team

Contributors to the platform



Ambassadors



Awards

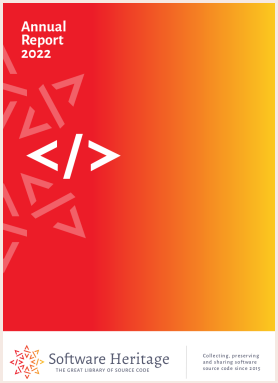
Antoine Pietri, best French PhD in Software Engineering **"Enabling Big Code analysis on a very large source code corpus"**. Awarded by the CNRS research working group GPL
https://theses.hal.science/tel-03515795v1

Stefano Zacchiroli with **Davide Rossi**. Google Award for Inclusion Research 2022, for the research project **"What Causes the Lack of Diversity in Open Source?"**.
https://research.google/outreach/aii-program/recipients/

Antoine Pietri with **Stefano Zacchiroli**. Award Best Dataset Paper: **"A Large-scale Dataset of (Open Source) License Text Variants"**.
https://arxiv.org/abs/2204.00256

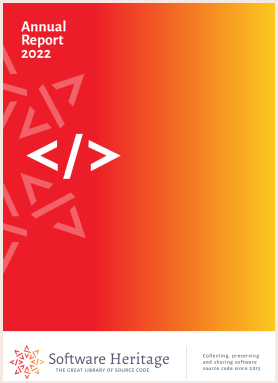
Report and videos

Annual report



Report and videos

Annual report

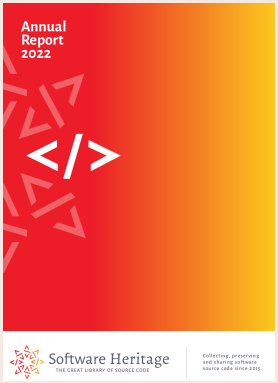


5 years in 5 minutes

Link

Report and videos

Annual report



5 years in 5 minutes

Link

Evolution of our codebase

Link