

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

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

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

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.”

Apollo 11 source code (excerpt)

```
P63SP0T3      CA      BIT6      # IS THE LR ANTENNA IN POSITION 1 YET
              EXTEND
              RAND    CHAN33
              EXTEND
              BZF     P63SP0T4      # BRANCH IF ANTENNA ALREADY IN POSITION 1

              CAF     CODE500      # ASTRONAUT:  PLEASE CRANK THE
              TC      BANKCALL     #              SILLY THING AROUND
              CADR    GOPERF1
              TCF     GOTOP00H     # TERMINATE
              TCF     P63SP0T3     # PROCEED   SEE IF HE'S LYING

P63SP0T4      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

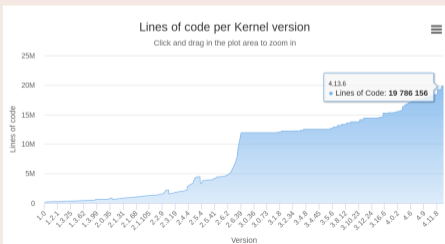
Apollo 11 Guidance Computer (~60.000 lines), 1969



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

Margaret Hamilton

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

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*

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

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 Commissioner](#) 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.”

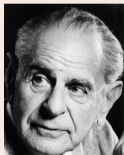
The experimental method



- 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

... updated for the digital age

For an experiment involving software, we need

- open access** to the scientific article describing it
- open data sets** used in the experiment
- source code** of all the components
- environment** of execution
- stable references** between all this

Remark

The first two items are already widely discussed!

... what about *software*?

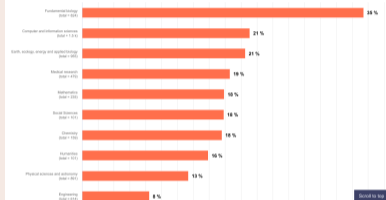
Software is a pillar of Open Science

Software powers modern research

Proportion of publications in France that mention code or software sharing by discipline

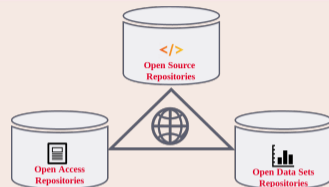
Sort by:

Highest volume Highest sharing rate



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*

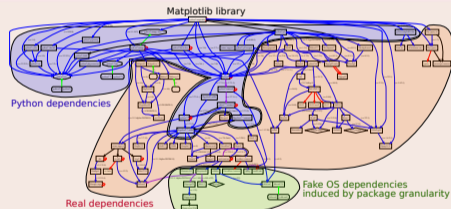
Source code is *special* (software is *not* data)

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*



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

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

International highlights

Paris Call on Software Source code (2019, UNESCO)

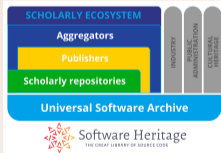


40 international experts call to “promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”



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



SECOND FRENCH PLAN FOR OPEN SCIENCE

Generalising open science in France 2021-2024



1

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

Path Three : Opening up and promoting source code produced by research

7

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

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

8

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

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

9

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

3

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

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, ...

20+ active members

Chairs: Roberto Di Cosmo and François Pellegrini

- Florent CHUFFART (Univ Grenoble Alpe)
- Mélanie CLÉMENT-FONTAINE (Univ Paris-Saclay - Versailles Saint-Quentin)
- Laurent COSTA (UMR 7041 ArScAn)
- Ludovic COURTÈS (Inria)
- Sébastien GÉRARD (Univ Paris-Saclay, CEA, List)
- Mathieu GIRAUD (CNRS, Univ Lille)
- Timothée GIRAUD (CNRS)
- Jean-Yves JEANNAS (Univ Lille, AFUL)
- Nicolas JULLIEN (IMT Atlantique)
- Daniel LE BERRE (Univ Artois, CNRS)
- Violaine LOUVET (CNRS / GRICAD - Univ Grenoble Alpes)
- Camille MAUMET (Inria, Univ Rennes, CNRS, Inserm)
- Clémentine MAURICE (CNRS)
- Grégory MIURA (Univ Bordeaux Montaigne)
- Raphaël MONAT (LIP6, Sorbonne Université)
- Sophie RENAUDIN (AP-HP)
- Jeanne ROBINEAU (IRD)
- Nicolas ROUGIER (Inria, Univ Bordeaux, CNRS)
- François SABOT (IRD)
- Sylvie TONDA-GOLDSTEIN (Inria)
- Samuel THIBAUT (Univ Bordeaux) (Univ Paris-Saclay)

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs**
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

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

Where is the source code?

Collaborative development platforms (aka "forges")

- BitBucket, GitLab(.com), GitHub, etc.
- support for version control, issues, etc.
- example:
 - <https://github.com/rdicosmo/parmap>
 - <https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/>

Distribution platforms

- CTAN, CRAN, PyPi, Debian, etc.
- example: <https://ctan.org/pkg/biblatex-software>

Archives

- Software Heritage
- example: [archived version of biblatex-software](#)

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

The screenshot shows a software artifact page with the following details:

- Two status indicators: "Artifacts Available" (green) and "Artifacts Evaluated & Functional" (red).
- Authors/Contributors: [Authors Info & Affiliations](#)
- DOI: <https://doi.org/10.1145/...> Version: 1.0
- Description: A source archive of [redacted], and the version of [redacted] used in the paper eval. A more up-to-date version of [redacted] can be found at github.com/.../.../git
- Assets: Read Me [redacted]
- Download (3.5 KB)

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!

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach**
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd



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

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 archive ever built



Directories

13,486,743,792

Commits

3,555,307,942

Authors

63,835,635

Projects

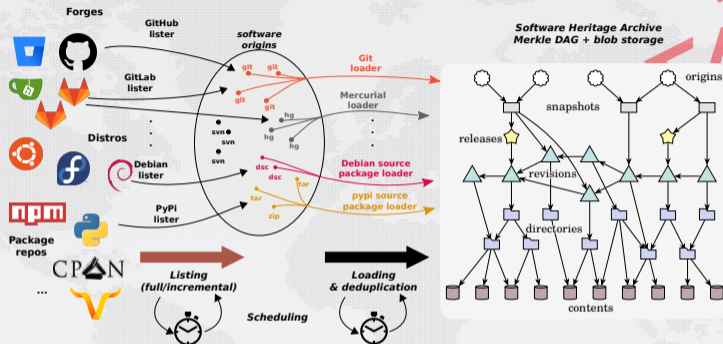
262,077,475

Releases

74,477,682

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

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

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](#)

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!**
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

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](#)

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage**
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

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 ...

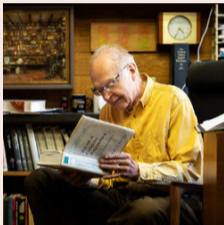
The call is published on Feb 2019

“[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>



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>

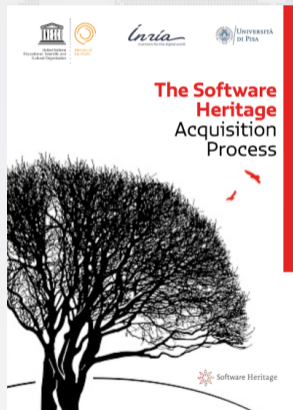
A unique opportunity

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”



- **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”



- **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*

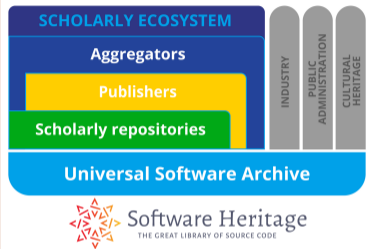
- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge**
- 9 Software Heritage, cont'd

Hold your breath!

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

- 1 Introduction
- 2 Software source code for Open Science
- 3 An emerging policy framework
- 4 Addressing the needs
- 5 Meet Software Heritage: a radically different approach
- 6 Demo time!
- 7 Collaboration with UniPi: software source code as Heritage
- 8 Collaboration with UniPi: software source code as Knowledge
- 9 Software Heritage, cont'd

EOSC SIRS report: Software Source Code and Open Science, 2020



Connect scholarly ecosystem with the whole software ecosystem

See e.g. [the French public administration open source catalog](#)

Ongoing work: FAIRCORE4EOSC

A full workpackage:

- connectors with InvenioRDM, episcience, Dagstuhl, swMath, etc.
- Software Heritage mirror for the EOSC
- standardisation of CodeMeta and SWHID

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

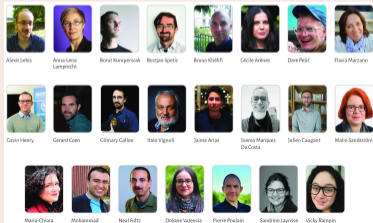
- vendor neutral
- open source
- worldwide, long term
- born and based in the EU

A growing and active community

Team



Ambassadors

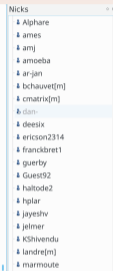


R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)

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, complaining it can't find sw.h.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 sw.h.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 sw-h-scheduler without the missing dependency on pytest-postgresql
  
```



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/air-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>

Annual report



Collecting, preserving
and sharing software
source code since 2013

5 years in 5 minutes

[Link](#)



Evolution of our codebase

[Link](#)

