

Preserving our Software Heritage

Roberto Di Cosmo
SWHAP Days

Director, Software Heritage
Inria and Université de Paris Cité

October 19th 2022



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 Software as Heritage
 - 2 How to preserve our software heritage
 - 3 Meet Software Heritage
 - 4 A piece of the puzzle

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)

```
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
```

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;
}
```

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.”

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

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

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 ...

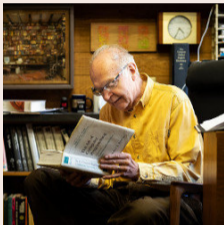


The call is published on Feb 2019

“[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>

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

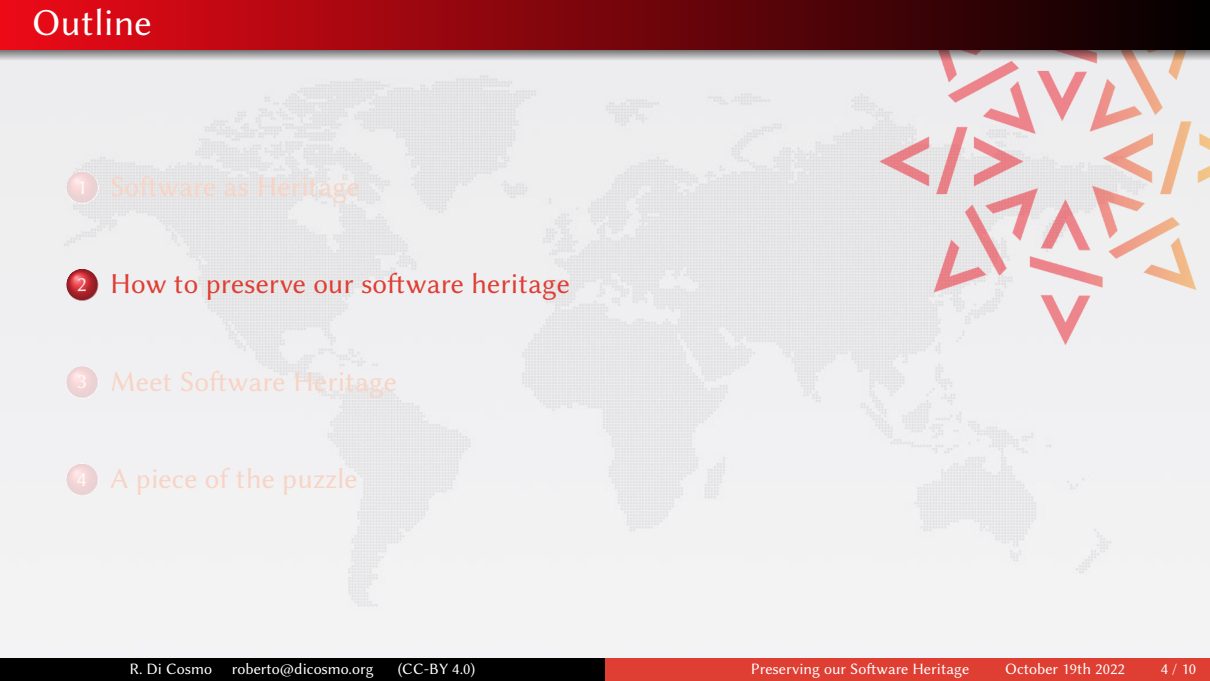
Donald E. Knuth, Len Shustek

<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...

- 
- 1 Software as Heritage
 - 2 How to preserve our software heritage
 - 3 Meet Software Heritage
 - 4 A piece of the puzzle

Some popular approaches, and why they do not fit the bill

A - Since the 1970's 1990's

.zip or .tar file on:

- ftp server
- web page
- document archive (+ DOI)

Some popular approaches, and why they do not fit the bill

A - Since the 1970's 1990's

.zip or .tar file on:

- ~~ftp~~ server
- web page
- document archive (+ DOI)

B - Since the 2000's

Rely on *software forges*

- institutional or project ones
- free commercial ones: BitBucket, GitHub, GitLab, ...

Some popular approaches, and why they do not fit the bill

A - Since the 1970's 1990's

.zip or .tar file on:

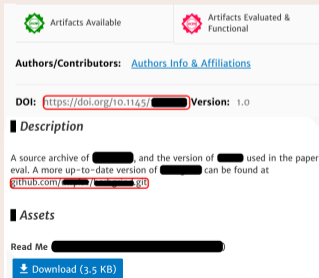
- ~~ftp server~~
- **web page**
- **document archive** (+ DOI)

B - Since the 2000's

Rely on *software forges*

- institutional or project ones
- free commercial ones: BitBucket, GitHub, GitLab, ...

C: a mix of the two



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

- Two status indicators: "Artifacts Available" (green icon) and "Artifacts Evaluated & Functional" (red icon).
- Section "Authors/Contributors:" with a link to "Authors Info & Affiliations".
- Section "DOI:" with a red box around the URL "https://doi.org/10.1145/[redacted]".
- Section "Version:" with the value "1.0".
- Section "Description" with a paragraph: "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/\[redacted\]/\[redacted\].git](https://github.com/[redacted]/[redacted].git)".
- Section "Assets" with a "Read Me" link and a "Download (3.5 KB)" button.

Some popular approaches, and why they do not fit the bill

A - Since the 1970's 1990's

.zip or .tar file on:

- ~~ftp server~~
- **web page**
- **document archive** (+ DOI)

B - Since the 2000's

Rely on *software forges*

- institutional or project ones
- free commercial ones: BitBucket, GitHub, GitLab, ...

C: a mix of the two

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

- Two status indicators: "Artifacts Available" (green icon) and "Artifacts Evaluated & Functional" (red icon).
- Section "Authors/Contributors:" with a link "Authors Info & Affiliations".
- Section "DOI:" with a red box around the URL "https://doi.org/10.1145/..." and "Version: 1.0".
- Section "Description" with a red box around the text "A source archive of ... and the version of ... used in the paper eval. A more up-to-date version of ... can be found at github.com/.../...-git".
- Section "Assets" with a "Read Me" link and a "Download (3.5 KB)" button.

Can get no satisfaction...

- A *Poor user experience*
- B *Preservation?*
- C *Can do 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!

- 
- 1 Software as Heritage
 - 2 How to preserve our software heritage
 - 3 Meet Software Heritage
 - 4 A piece of the puzzle



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 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 all software source code

Research infrastructure



enable analysis of all software source code

One infrastructure
open and shared



One infrastructure
open and shared



Largest archive

One infrastructure
open and shared



Largest archive

Technology

- transparency and FOSS
- replicas all the way down

Content (billions!)

- **intrinsic identifiers**
- facts and provenance

Organization

- non-profit
- multi-stakeholder

Sharing the vision



United Nations
Educational, Scientific and
Cultural Organization



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors



Diamond sponsor



Platinum sponsors



Gold sponsors

openinventionnetwork



Silver sponsors

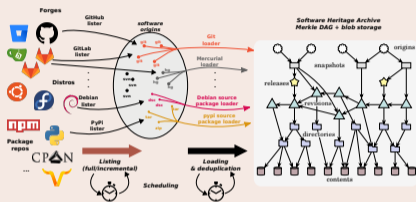


Bronze sponsors



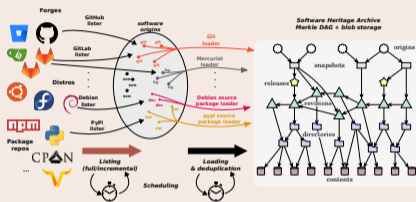
Solid foundation for archive and reference

Archive (12B+ files, 180M+ projects)



Solid foundation for archive and reference

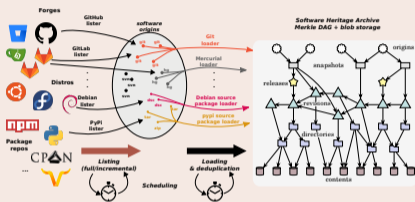
Archive (12B+ files, 180M+ projects)



- save.softwareheritage.org
- deposit.softwareheritage.org

Solid foundation for archive and reference

Archive (12B+ files, 180M+ projects)



- save.softwareheritage.org
- deposit.softwareheritage.org

Reference (25 billion SWHIDs)

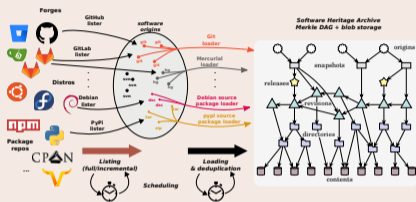
Intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



Now supported in [SPDX 2.2](#), [Wikidata](#) etc.

Solid foundation for archive and reference

Archive (12B+ files, 180M+ projects)



- save.softwareheritage.org
- deposit.softwareheritage.org

Reference (25 billion SWHIDs)

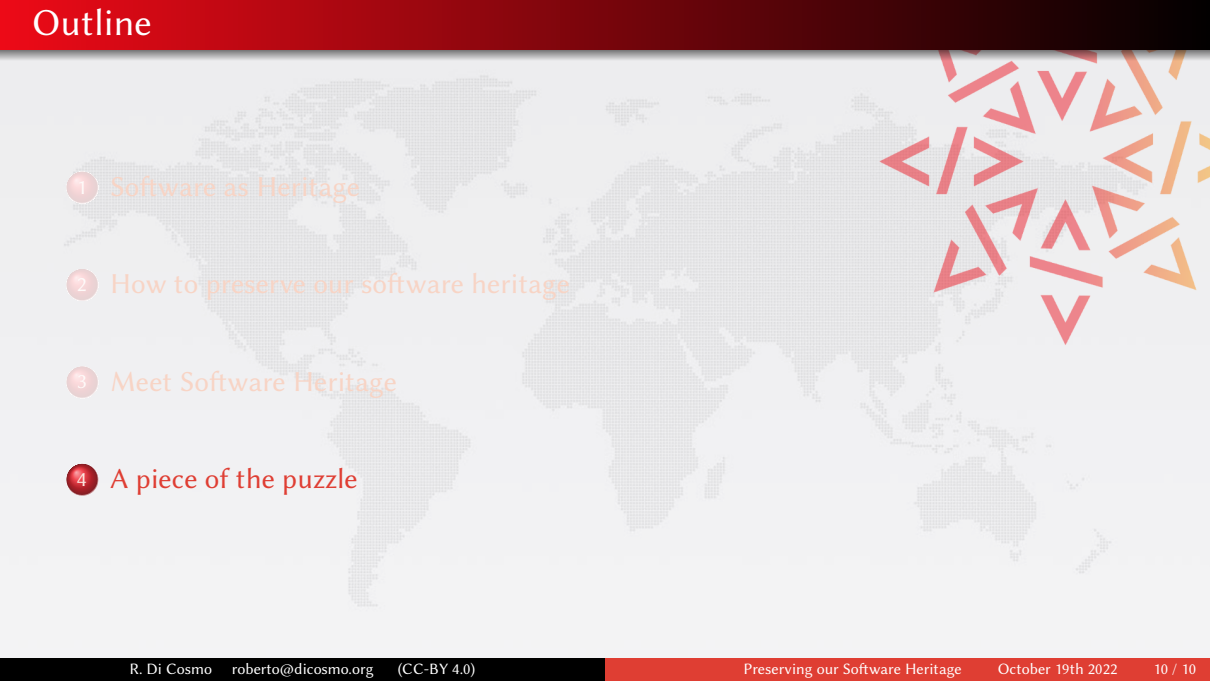
Intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



Now supported in [SPDX 2.2](#), [Wikidata](#) etc.

An example is worth a thousand words

Compare the links to code in this [Quartz 2016 article](#) and this [2019 blog post](#)

- 
- 1 Software as Heritage
 - 2 How to preserve our software heritage
 - 3 Meet Software Heritage
 - 4 A piece of the puzzle

A great challenge

collecting, curating, preserving and telling the history of landmark software

Software Heritage's commitment

- provide archival and reference infrastructure
- humbly contribute some tools to the vast preservation landscape
 - the SWHAP source code acquisition and curation protocol
 - the Software Stories approach to interconnect historical data with source code

Let's work together

alone we go faster, together we go further
ancient african saying