Preserving our Software Heritage and its Stories

Roberto Di Cosmo
Capturing Hidden ACM Heritage Seminar

Director, Software Heritage
Inria and Université de Paris Cité

September 30th 2022
1. Software as Heritage
2. How to preserve our software heritage
3. Meet Software Heritage
4. Preserving the past
5. Preserving the present and the future
6. Conclusions
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)

<table>
<thead>
<tr>
<th>P63SPOT3</th>
<th>CA</th>
<th>BIT6</th>
<th># IS THE LR ANTENNA IN POSITION 1 YET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAND</td>
<td>CHAN33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BZF</td>
<td>P63SPOT4</td>
<td># BRANCH IF ANTENNA ALREADY IN POSITION 1</td>
</tr>
<tr>
<td>CAF</td>
<td>CODE508</td>
<td># ASTRONAUT: PLEASE CRANK THE</td>
<td></td>
</tr>
<tr>
<td>TC</td>
<td>BANKCALL</td>
<td># SILLY THING AROUND</td>
<td></td>
</tr>
<tr>
<td>CADR</td>
<td>GOPERF1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCF</td>
<td>GOT0POOH</td>
<td># TERMINATE</td>
<td></td>
</tr>
<tr>
<td>TCF</td>
<td>P63SPOT3</td>
<td># PROCEED SEE IF HE’S LYING</td>
<td></td>
</tr>
<tr>
<td>P63SPOT4</td>
<td>TC</td>
<td>BANKCALL</td>
<td># ENTER INITIALIZE LANDING RADAR</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>SETPOS1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>POSTJUMP</td>
<td># OFF TO SEE THE WIZARD ...</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>BURNBABY</td>
<td></td>
</tr>
</tbody>
</table>

Quake III source code (excerpt)

```c
float Q_rsqrt(float number )
{
  long i;
  float x2, y;
  const float threehalves = 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 * ( threehalves - ( x2 * y * y ) ); // 1st iteration
  // y = y * ( threehalves - ( 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 …

“[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”

"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. Preserving the past
5. Preserving the present and the future
6. Conclusions
Some popular approaches

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

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)

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

1. Software as Heritage
2. How to preserve our software heritage
3. Meet Software Heritage
4. Preserving the past
5. Preserving the present and the future
6. Conclusions
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
Universal software archive, principled


One infrastructure
open and shared

Software Heritage

Largest archive

Technology
- transparency and FOSS
- replicas all the way down

Content (billions!)
- intrinsic identifiers
- facts and provenance

Organization
- non-profit
- multi-stakeholder
An international, non profit initiative built for the long term

Sharing the vision

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

Bronze sponsors
Global development history permanently archived in a uniform data model

- over 12 billion unique source files from over 180 million software projects
- ~1PB (uncompressed) blobs, ~25 B nodes, ~350 B edges
Intrinsic Identifiers for software artefacts

Software Heritage Identifiers (SWHID)

20+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs

Emerging standard: Linux Foundation SPDX 2.2; IANA registered; WikiData P6138

Full fledged source code references for reproducibility

Examples: Apollo 11 AGC excerpt, Quake III rsqrt; Guidelines available, see ICMS 2020
A walkthrough

- **Browse** (e.g. Apollo 11, and your work may be already there!)
- **Trigger archival**, use the `updateswh` browser extension (GitHub action available too)
- Get and use SWHIDs (full specification available online)
- Cite software using the `biblatex-software` package from CTAN
- Example in a journal: an article from IPOL
- Example with Parmap: `devel` on Github, archive in SWH, curated deposit in HAL
- Extracting all the software products for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Example use in research articles:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment
Outline

1. Software as Heritage
2. How to preserve our software heritage
3. Meet Software Heritage
4. Preserving the past
5. Preserving the present and the future
6. Conclusions
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 and the SWHAP Days hybrid event, 19 and 20 October 2022
Electronic music in Pisa: group led by the late M° P. Grossi

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

See this live

- the archived SWHAP repository
- and its Software Story
Meet the team

Laura Bussi
PhD candidate in Computer Science
SWHAP
University of Pisa

Carlo Montangero
Computer Scientist, SWHAP
University of Pisa

Kenneth Seals-Nutt
Computer Scientist & Software Engineer
Co-founder of ScienceStories.io

Roberto Di Cosmo
Computer Scientist
Founder of Software Heritage

Elisabetta Mori
Historian of Computing
SWH Visiting Scientist

Katherine Thornton
Information Scientist
Co-founder of ScienceStories.io

Morane Gruenpeter
Software Engineer
Project Manager

Guido Scatena
Computer Scientist, SWHAP
ISPRA
A proposal for a working agenda

Search and find software source code associated to landmark ACM articles

Reconstruct development history, archive in SWH

Link publications to the source code using the SWHID identifier

Collect oral and documentary history around it, and build a Software Story

Connect with all the relevant history collections
Outline

1. Software as Heritage
2. How to preserve our software heritage
3. Meet Software Heritage
4. Preserving the past
5. Preserving the present and the future
6. Conclusions
Source code history for Open Science

Software powers modern research

[...] software [...] essential in their fields.
Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data
Christine Borgman, Paris, 2018

A key 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
Let’s do it right from the start

Archiving and Referencing

For all source code used in research (yes, even small scripts!)
- ensure it is archived in Software Heritage (see save code now)
- get the proper SWHID for your software (see detailed HOWTO)
- add it to research articles for reproducibility (see detailed HOWTO)

Describing and Citing/Crediting

For software projects, go the extra mile:
- add proper metadata (e.g. codemeta.json, see the codemeta generator)
- cite software (e.g. using biblatex-software, in CTAN, TeXLive and acmart)
- index on par with publications (see the french portal HAL)

ACM action item

connect ACM DL and Badging program with Software Heritage
Focus on Academia: growing adoption (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts
International Journal of Digital Curation, 2020

Reference archive for swmath.org

See code links, e.g. SemiPar package

Policy: France
National Plan for Open Science and Research Infrastructures

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: Europe

EOSC SIRS report
- SWHIDs
- archive

Guidelines
- summary
- ICMS 2020

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)

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)
A long way to go: ACM has an important role to play!

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


- Laura Bussi, Roberto Di Cosmo, Carlo Montangero, Guido Scatena. *The software heritage acquisition process*. UNESCO, Università di Pisa, Inria, 2019