Outline

1. Software Source Code is knowledge
2. Software Heritage
3. Demo time!
4. The way forward
Software source code: human readable and executable knowledge

Harold Abelson, Structure and Interpretation of Computer Programs (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 GOTOPO0M  # 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 threehalves = 1.5F;
    x2 = number * 0.5F;
    y = number;
    i = *( long * ) &y; // evil floating point bit level hacking
    i = 0xf5379dfL - ( 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.”

@rdicosmo Roberto Di Cosmo @swheritage

Archiving and Referencing source code CC-BY 4.0 July 2020 2 / 11
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
- sophisticated *developer communities*

A vast part is not research software

- industry and communities drive standards, build the necessary support layers
Source code is *special*, cont’d

**Versioning, granularity**

**Project**  “Inria created OCaml and Scikit-learn”

**Release**  “2D Voronoi Diagrams were introduced in CGAL 3.1.0”

**Precise state of a project**  “This result was produced using commit 0064fbd…”

**Code fragment**  “The core algorithm is in lines 101 to 143 of the file parmap.ml contained in the precise state of the project corresponding to commit 0064fbd…”

**Authors can have multiple roles:**

- Architecture, Management, Development, Documentation, Testing, …
Software Source code: pillar of Open Science

Three pillars of Open Science

A plurality of needs

**Researcher**
- archive and reference software used in articles
- find useful software
- get credit for developed software
- verify/reproduce/improve results

**Laboratory/team**
- track software contributions
- produce reports / web page

**Research Organization**
- know its software assets
- technology transfer
- impact metrics
<table>
<thead>
<tr>
<th>What is at stake</th>
<th>in increasing order of difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Archival</strong></td>
<td>Research software artifacts must be properly archived</td>
</tr>
<tr>
<td></td>
<td>make sure we can retrieve them (reproducibility)</td>
</tr>
<tr>
<td><strong>Identification</strong></td>
<td>Research software artifacts must be properly referenced</td>
</tr>
<tr>
<td></td>
<td>make sure we can identify them (reproducibility)</td>
</tr>
<tr>
<td><strong>Metadata</strong></td>
<td>Research software artifacts must be properly described</td>
</tr>
<tr>
<td></td>
<td>make it easy to discover them (visibility)</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Research software artifacts must be properly cited (not the same as referenced!)</td>
</tr>
<tr>
<td></td>
<td>to give credit to authors (evaluation!)</td>
</tr>
</tbody>
</table>

We need infrastructures designed for software source code: now we have one!
1. Software Source Code is knowledge
2. Software Heritage
3. Demo time!
4. The way forward
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
Addressing the four needs

**Archive (8B+ files, 130M+ projects!)**

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

**Reference (20 billion SWHIDs)**

Intrinsic, decentralised, cryptographically strong identifiers, SWHIDs

Now supported in SPDX 2.2, Wikidata etc.

**Describe**

- Intrinsic metadata from source code
- Contributed the Codemeta generator

**Cite**

- Contributed software citation style
  biblatex-software, v 1.2-2 now on CTAN

Archiving and Referencing source code  CC-BY 4.0  July 2020  8 / 11
1. Software Source Code is knowledge

2. Software Heritage

3. Demo time!

4. The way forward
A walkthrough

- Browse the archive
- Get and use SWHIDs (full specification available online)
- cite software with the biblatex-software style from CTAN
- Example use in a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Example use in a research article: extensive use of SWHIDs in a replication experiment
- Trigger archival of your preferred software in a breeze
- curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- rescue landmark legacy software, see the SWHAP process with UNESCO
Software Source Code is knowledge

Software Heritage

Demo time!

The way forward
An international, non-profit initiative built for the long term

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Platinum sponsors

Silver sponsors

Bronze sponsors

Gold sponsors

@rdicosmo Roberto Di Cosmo @swheritage

Archiving and Referencing source code CC-BY 4.0 July 2020
Come in, we’re open!

### Software Heritage
- **universal** source code archive
- **intrinsic** identifiers (SWHIDS)
- **open, non profit**, long term
- *infrastructure* for Open Science

### You can help improve science!
- **use** SWH and **save** relevant source code
- **build on** SWH (see swmath.org and ipol.im)
- **contribute** to SWH: *it is open source*
- spread the word

---

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli  
*Building the Universal Archive of Source Code*, CACM, October 2018 ([10.1145/3183558](https://doi.org/10.1145/3183558))

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli  
*Referencing Source Code Artifacts: a Separate Concern in Software Citation*,  
CiSE 2020 ([10.1109/MCSE.2019.2963148](https://doi.org/10.1109/MCSE.2019.2963148)) ([hal-02446202](https://hal.archives-ouvertes.fr/hal-02446202))

Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, Arnaud Legrand and Nicolas Rougier  
*Attributing and referencing (research) software: Best practices and outlook from Inria*,  
CiSE 2020 ([10.1109/MCSE.2019.2949413](https://doi.org/10.1109/MCSE.2019.2949413)) ([hal-02135891](https://hal.archives-ouvertes.fr/hal-02135891))