Roberto DI COSMO
Inria and University of Paris
Building the Software Pillar of Open Science
Software is everywhere
### Software Source Code

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

**Hello World**

<table>
<thead>
<tr>
<th>Program (excerpt of binary)</th>
<th>Program (source code)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4004e6: 55</td>
<td>/* Hello World program */</td>
</tr>
<tr>
<td>4004e7: 48 89 e5</td>
<td>#include&lt;stdio.h&gt;</td>
</tr>
<tr>
<td>4004ea: bf 84 05 40 00</td>
<td>void main()</td>
</tr>
<tr>
<td>4004ef: b8 00 00 00 00</td>
<td>{</td>
</tr>
<tr>
<td>4004f4: e8 c7 fe ff ff</td>
<td>printf(&quot;Hello World&quot;);</td>
</tr>
<tr>
<td>4004f9: 90</td>
<td>}</td>
</tr>
<tr>
<td>4004fa: 5d 4004fb: c3</td>
<td>}</td>
</tr>
</tbody>
</table>
## 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.”

<table>
<thead>
<tr>
<th>Apollo 11 source code (excerpt)</th>
<th>Quake III source code (excerpt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P63SP0T3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET</td>
<td>float _qrsqrt( float number )</td>
</tr>
<tr>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td>RAND CMAN33</td>
<td></td>
</tr>
<tr>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td>BZF P63SP0T4 # BRANCH IF ANTENNA ALREADY IN POSITION 1</td>
<td></td>
</tr>
<tr>
<td>CAF CODE500 # ASTRONAUT: PLEASE CRANK THE</td>
<td></td>
</tr>
<tr>
<td>TC BANKCALL # SILLY THING AROUND</td>
<td></td>
</tr>
<tr>
<td>CADR GOPERF1</td>
<td></td>
</tr>
<tr>
<td>TCF GOTOPOOH # TERMINATE</td>
<td></td>
</tr>
<tr>
<td>TCF P63SP0T3 # PROCEED SEE IF HE'S LYING</td>
<td></td>
</tr>
<tr>
<td>P63SP0T4 TC BANKCALL</td>
<td></td>
</tr>
<tr>
<td>CADR SETPOS1 # ENTER INITIALIZE LANDING RADAR</td>
<td></td>
</tr>
<tr>
<td>TC POSTJUMP # OFF TO SEE THE WIZARD ...</td>
<td></td>
</tr>
<tr>
<td>CADR BURNBABY</td>
<td></td>
</tr>
</tbody>
</table>

Quake III source code (excerpt):

```c
float _qrsqrt(float number)
{
    long i;
    float x2, y;
    const float threehalves = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = *(long *)5y; // evil floating point bit level hacking
    i = x2)*7F3759df - (1 >> 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
}
```

Len Shustek, Computer History Museum 2006

“Source code provides a view into the mind of the designer.”
**A lightning fast growth**

<table>
<thead>
<tr>
<th>Apollo 11 (~60,000 lines), 1969</th>
<th>Linux Kernel: 20 million lines...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;When I first got into it, nobody knew what it was that we were doing. It was like the Wild West.&quot;</td>
<td><img src="image" alt="Graph showing the growth of source code lines" /></td>
</tr>
<tr>
<td>Margaret Hamilton</td>
<td></td>
</tr>
</tbody>
</table>

Open source software is eating the software world

- tens of millions of developers collaborate on open source software worldwide today

Reuse is the new rule

- 80% to 90% of a new application is... just reuse!  
  (Sonatype survey, 2017)
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**

fruit of human ingenuity: design, algorithm, code, test, documentation, community, funding, and so many more facets…

**copyright law applies!**
### A long overlooked pillar of 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

#### Missing pillar: software (source code)

- The links in the picture are essential

---

#### Software may be a **tool**, an **outcome** and a **research object**

- Open source (open access to the source code) is necessary
  - avoid reinventing the wheel, accelerate scientific discoveries
  - **preserving source code** and its history is necessary for *reproducibility*
A plurality of needs

Researchers

- archive and reference software used in articles
- find useful software
- get credit for software contributions
- verify, reproduce, improve results

Laboratories/teams

- track software contributions
- produce reports
- maintain webpage

Research Organization

know its software assets

- technology transfer
- impact metrics
- funding strategy
- career evaluation
What is at stake: ARDC in increasing order of difficulty

**Archive**
Research software artifacts must be properly **archived**
make sure we can **retrieve** them (*reproducibility*)

**Reference**
Research software artifacts must be properly **referenced**
make sure we can **identify** them (*reproducibility*)

**Describe**
Research software artifacts must be properly **described**
make it easy to **discover** and reuse them (*visibility*)

**Cite/Credit**
Research software artifacts must be properly **cited** *not the same as referenced!* to give credit to authors (*evaluation*)
What is at stake: beyond ARDC

Policy framework for dissemination, reuse, evaluation and recognition
- open source policy for publicly funded research software
- incentives and recognition for researchers and engineers

Sustainability
- Organisational schemas, legal tools, economic models, processes and policies to ensure research software can be maintained and sustained over time

Technology transfer and industry collaboration
- Approaches, support, methods, processes to establish connections with industry in order to foster uptake and transfer of research software

Advanced technologies and tools
- software quality, reproducibility, and traceability
Focus on infrastructures for ARDC
The big picture (EOSC SIRS 2020 report)

Research Software Infrastructures: Overall Architecture

Key observations

- One common layer for archive and reference shared with all the software world
- Added value in the scholarly ecosystem: curation, citation and credit
The HAL – Software Heritage success story

https://hal.archives-ouvertes.fr/hal-02130801

swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65
Strategic remarks

Significant risk factors that we need to take into account

- balkanisation
  - proliferation of infrastructure silos
  - duplicated contents with different identifiers
  - costly efforts to federate after-the-fact
- closed (code, gouvernance) and/or for profit
- operations funded with project money
Focus on broader policy issues
## What is needed

### Policy for dissemination and reuse

- **Set the default to open source** for research software
- **Open source creates value**: adapt technology transfer and industry collaboration to it

### Framework for evaluation and recognition

- Make software development count in a career (not the case in many countries)
- Avoid purely quantitative indicators

### Sustainability

**Technical**
- Improve quality of key research software

**Organisational**
- Professional practices for governance and maintenance

**Financial**
- Make open source research software as easy to fund as buying a license
Good news: awareness is raising

Paris Call on Software Source code (2019)

"[We call to] promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, sharing good practices and recognising in the careers of academics their contributions to high quality software development, in all their forms”


EOSC SIRS report (2020) and EOSC TF on infrastructures for research software

"all research software should be made available under an Open Source license by default, and all deviations from this default practice should be properly motivated »

See https://doi.org/10.2777/28598

UNESCO Open Science Recommendations (2021)

"Open science infrastructures should be organized and financed upon an essentially not-for-profit and long-term vision, which enhance open science practices and guarantee permanent and unrestricted access to all."
Focus on the French National plan for Open Science, 2021-2024

2nd National Plan for Open Science (6/7/2021)

Open and promote research software source code

- actions (selection)
  - charter for research software policy
  - recognize software development (see the 2021 prize)
  - coordinate communities of practice
  - build a connected ecosystem of research outputs

- recommendations (selection)
  - Archive source code in Software Heritage
  - Standardise and use SWHID
  - Build a national catalog of research software

See official announcement
Conclusions
The road ahead

Infrastructures for research software

Recognise software as a key enabler of research, establish an international network of infrastructures for research software, adapt funding instruments to digital infrastructures where human cost is predominant.

See Ulrike Luke’s talk today

Linking software, publications and data

Key role of the publishers and institutional archives: recognize software as a noble research output on its own, not just a piece of data. Software source code is special: it needs specific infrastructures and identifiers.

See Melissa Harrison’s talk today
Institutional representation and support

Establish an office in charge of (open source) research software, to help with funding, open licenses, governance, translation, etc.

See Sayeed Choudhury’s talk today

Incentives, recognition and evaluation

Value quality research software on a par with publications for careers in academia and give them recognition and visibility. **Beware of quantitative indicators** in evaluation, they are much more damaging for software than for publications.

See the Open Science Free Software Awards ceremony today
Let’s build together the software pillar of open science
it’s a long road, but together we can make it

Questions

References

- UNESCO, Draft recommendations on Open Science, 2021, (online)
- French Ministry of Research, Second National Plan for Open Science, 2021, (online)
- J.F. Abramatic, R. Di Cosmo, S. Zacchioli, Building the Universal Archive of Source Code, CACM, October 2018 (10.1145/3183558)