Outline

★ Introduction
★ Preserving source code
★ Software Heritage the universal source code archive
★ Research Software: a first class research output
★ Good practices for software curation
★ Conclusion
Software engineer and metadata specialist

Timeline:

- **2008-2011** B.A in *Musique* (Harpist)
- **2012-2015** Licence (B.SC) in *Computer Science* @CNAM
- **2015-2017** Master (M.SC) in *Software Engineering* (R&D) @UPMC
- **2017** Internship *Software Heritage* (SWH)
- **2018-2019** European project EU2020 *CROSSMINER* (on SWH team)
- **2020-2022** European project *FAIRsFAIR* (on SWH team, @Inria)
- **2022-2025** European project *FAIRCORE4EOSC* (on SWH team, @Inria)

Working groups for Open Science and digital preservation

- the Research Data Alliance’s *Software Source Code* Interest Group (SSC IG),
- the FORCE11’s *Software Citation* Implementation Working Group (SCI WG),
- the joint RDA, ReSA & FORCE11 *FAIR for Research Software* Working Group (FAIR4RS WG)
- WikiData for *Digital Preservation* initiative (WikiDigi).
What is software?

Software as a concept
- project or entity
- the community around the project
- the software idea / algorithms / solutions

Software artifacts
- Executables
- Source code
What about software source code?
Hello World program

```c
#include<stdio.h>

void main()
{
    printf("Hello World");
}
```
The Knowledge is in the Source Code

“Programs must be written for people to read, and only incidentally for machines to execute.”

Harold Abelson, 1985
Structure and Interpretation of Computer Programs (1st ed.),

“Source code provides a view into the mind of the designer.”

Len Shustek, 2006
Computer History Museum

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

GPL Licence
Outline

- Introduction
- Preserving source code
- Software Heritage: the universal source code archive
- Research Software: a first class research output
- Good practices for software curation
- Conclusion
Why source code is fragile?
Source code can be destroyed

Google Kills Off Google Code

Natasha Lomas @riptari / 10:58 AM GMT+1 • March 13, 2015
Source: TechCrunch

When we started the Google Code project hosting service in 2006, the world of project hosting was limited. We were worried about reliability and stagnation, so we took action by giving the open source community another option to choose from. Since then, we've seen a wide variety of better project hosting services such as GitHub and Bitbucket bloom. Many projects moved away from Google Code to those other systems. To meet developers where they are, we ourselves migrated nearly a thousand of our open source projects from Google Code to GitHub.

As developers migrated away from Google Code, a growing share of the remaining projects were spam or abuse. Lately, the administrative load has consisted almost exclusively of abuse management. After profiling non-abusive activity on Google Code, it has become clear to us that the service is by now needed no more.

Beginning today, we have disabled new project creation on Google Code. We will be shutting down the service about 10 months from now on January 25th, 2016. Below, we provide links to migration tools designed to help you move your projects off of Google Code. We will also make ourselves available over the next three months to those projects that need help migrating from Google Code to other hosts.

- March 12, 2015 - New project creation disabled.
- August 24, 2015 - The site goes read-only. You can still checkout/view project source, issues, and wikis.
- January 25, 2016 - The project hosting service is closed. You will be able to download a tarball of project source, issues, and wikis. These tarballs will be available throughout the rest of 2016.

Google will continue to provide Git and Gerrit hosting for certain projects like Android and Chrome. We will also continue maintaining our mirrors of projects like Eclipse, kernel.org and others.
In science, reproducibility requires long-term access to source code.

Source: BitBucket blog

Sunsetting Mercurial support in Bitbucket

April 21, 2020 | 3 min read

Denise Chan

Just realized @Bitbucket disabled all mercurial repositories when the @asclnet informed me that a link associated with an old paper of mine was down. Thought all was lost, but someone archived all the repos! very classy move by @octobus_net and @SWHeritage.

[Update Aug 26, 2020] All hg repos have now been disabled and cannot be accessed.

[Update July 1, 2020] Today, mercurial repositories, snippets, and wikis will turn to read-only mode. After July 8th, 2020 they will no longer be accessible.

Source: BitBucket blog

250,000 repos
GitHub now belongs to Microsoft

Microsoft to acquire GitHub for $7.5 billion

June 4, 2018 | Microsoft News Center

Acquisition will empower developers, accelerate GitHub’s growth and advance Microsoft services with new audiences

Source: Microsoft Blog

GitLab gains developers after Microsoft buys rival GitHub

By Vibhuti Sharma, Supantha Mukherjee

Source: Reuters
Hosting your open-source project

- on a free, publicly available platform is fine.
- But you have to prepare for the platform shutdown (you need a plan B).
Version control system (VCS) history

- records changes made to a (set of) source code file(s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development
The development history is key

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
Outline

- Introduction
- Preserving source code
- **Software Heritage the universal source code archive**
- Research Software: a first class research output
- Good practices for software curation
- Conclusion
Collect, preserve and share all software source code

Preserving our heritage, enabling better software and better science for all

- Non-profit organization
- Launched in 2016 by INRIA (Roberto Di Cosmo & Stefano Zacchiroli)
- Archives publicly available code permanently and for free.
Software Heritage in a nutshell

**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
In numbers [https://archive.softwareheritage.org/](https://archive.softwareheritage.org/)

**Source files**
- 10545239307

**Commits**
- 2215028000

**Projects**
- 161168065

**Directories**
- 8776445689

**Authors**
- 43822776

**Releases**
- 17972096

Source: Software Heritage (June 2021)
Rescuing software

Google Code content now safely collected

Rescuing 25000+ endangered Mercurial repositories
An international, non profit initiative
How to archive the world's source code?
Archiving software
Crawling

The SWH archive **harvests actively** source code from different sources and converts all the source code into a single and universal data structure which is an enormous Merkle DAG [Merkle, 1987].

Crawling is separated into three phases:

1. **listing software sources**,  
2. **scheduling updates** and  
3. **loading the software artifacts into the archive**.
Data model

The data model adopted by Software Heritage to represent the information that it collects is centered around the notion of *software artifact*, using the following canonical names, from bottom to top:

- contents,
- directories,
- revisions and
- releases.

Using also origins, visits and snapshots to store provenance information. Read more in Software Heritage: Why and How to Preserve Software Source Code.

Merkle tree

Combination of:
- tree
- hash function

Classical *cryptographic construction*

- fast, parallel signature of large data structures
- widely used (e.g., Git, blockchains, IPFS, . . . )
- built-in deduplication
Intrinsic identifier

SWH provides a Persistent IDentifier (PID) that can identify each and every source code artifact with integrity, called a SWHID.

SWHIDs are intrinsic identifiers which are intimately bound to the designated object, they do not need a register, only agreement on a standard.

Intrinsic vs. extrinsic blog post

Go to API endpoint
The complete workflow

Full development history permanently archived in a uniform data model
Contents

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of
this license document, but changing it is not allowed.

Preamble
The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want, that you can change the software or use pieces of it in new free
programs, and that you know you can do these things.

To protect your rights, we need to make restrictions that prevent
exploitation of the free nature of the General Public License.

This license, the General Public License, applies to some special
problems that do not fit in well with the classical copyleft licenses.

Thisannounce
sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147
Directories

id: 515f00d44e92c65322aaa9bf3fa097c00d9b9c7d
Revisions

Details | Changes | Files
---|---|---
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep  1 14:26:13 2016)
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep  1 14:26:13 2016)
Subject: provenance.tasks: add the revision -> origin cache task
Parent: fc3a8b59ca1df424d860f2c29ab07fe4dc35d10 : test_storage: properly pipeline origin and cont...
provenance.tasks: add the revision -> origin cache task

swf/storage/provenance/tasks.py

```
tree 515f00d44e92c65322aaa9bf3fa097c00dddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fe4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task
```

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6
Releases

object c0c9f16b1e134f593e7567570a1761b156e6eb1d

type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

Release swh.storage v0.0.51

- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

-----BEGIN PGP SIGNATURE-----

iQzbABAABDJvXZTNFhxsuAWNvbGFrzQGRhbmrYraW1vbnQzXUACgkO7AWLmo2++
neqorw/aq6SOb5DpZ+Ea+iKN3rXgYs+1K1vEYh1wNKAwx8ek7aXz2EILDIt7uf
ahp26hpzq8nq5saC1+yXbfcih3L2ytrdDeshWXWq8xwWNMaEoEyDB8qaphwWbADSti2
ICBlitSjlXyuCrD9r3eK6Pvz8zsg+bBoSmjW53Dr6ju7Z7chK4m/Pgjly/HPPY5yo
IGEndWhn7VYh1Vm8t1n5zB7f5mXRaqA+becqddubTZxjx+jpIUTk8qycq3n3hm/FL
qs2muK8yvz3I8G/H1/pv/1450wBhInS53Th0fzuojEvP7kzH5F79quUh9ZKq+a
kij6aKvYiU00Mh+yKXj/JeLdR3+yWBF3JQ9oA1V8o87t6E1aLCnNpEhKc0Kmt/
d/gMkax11l/g0EfnsW67G6sDwKPKPHhgsVLO3v3QaQTNt1Rpm2006H9/tawzC
Gg/k1PhbH74hOli46wYPZyjeOzu2VXGFU6vVU89FQ42R/Ojnp/+ozXzdC15rjJSUOMn
RPTTFU8b5XUdXHGF6kgxhSTytnvpI9gdpc76U5bKu0aGe84AZm1ik0mGrwXCVFpqy9
nhhibB5NHIMoqyF6y7SoPUDbYK070l/YRUGKwDeK0KowSxXWkWUZG1kty6yYqlo29
quwgqzOQf8WQCB00ontAL2+HvF0cYvckMeuh62c9p/EHvukw=

-----END PGP SIGNATURE-----
Snapshots

commit 0ffebe2b5770109525eb3ce21691466c53a1d9150 refs/heads/atime
commit ba54a42a4e3f9fe32a48c3292c2e47cbe61c67eb refs/heads/directory-listing-arrays
commit d69e0d8bf892383f16589b27fbee1c85d27238db9c refs/heads/foo
commit c77f9e9eae0eb22f89409805a80119f67e46e080 refs/heads/master
commit teca197fc6626e2407e54b1ed9eb8a4461a0f7c2 refs/heads/tmp/generic-releases
git show-refs
tag 28043b1379cfe688d966597790fd4907c7571755 refs/tags/v0.0.1
tag 72a1991a3884e539969db06876f080bee72e6e2cd refs/tags/v0.0.10
tag 3590ee8a60b070e05ba376975fa230bbfa4fa5cc refs/tags/v0.0.11
tag 33378423f9a4b36a96777b0d85b7f6674d6556 refs/tags/v0.0.12
tag 06f7462755b327cf590311c2bfa063cf33b435d refs/tags/v0.0.13
tag 5a6235f8e86ab85581d7442667d92a1e32f3b refs/tags/v0.0.14
tag 586fb4ade588bdf5f6b0f939367b643cbb4b9c7f refs/tags/v0.0.15
tag 8cdd68b5f49ffbf363177742d2b9f6f668e06e51c refs/tags/v0.0.16
tag a5424446e3f0fbed3584e363c2ee93cc89a0bc7d6 refs/tags/v0.0.17
tag 228a2f1659d8d2222e556559462ce1e90fc4993d9 refs/tags/v0.0.18
tag 606979a4ca05d497fc0d24aa0d0dce82363ef47c refs/tags/v0.0.19
tag 3b2f5a59fc2a3d23bb06d5f156a16ad5382ec275a67 refs/tags/v0.0.2
tag 3147c3d31c4c6f6492f881e980b12376bdf72c7 refs/tags/v0.0.20
tag 215ea59dbab11e0826e0b7e76eb4b66673808f8 refs/tags/v0.0.21
tag 3fb168c2072a5d6525124257a1e5dfc0f5ffaa1df refs/tags/v0.0.22
tag 8cdbee88a4d73fc5d262789e460a16ac372aba4 refs/tags/v0.0.23

id: b464cad1b66ff266a37b46ea6e7a04b545e904b
The complete workflow

Full development history permanently archived in a **uniform data model**
Outline

★ Introduction
★ Preserving source code
★ Software Heritage: the universal source code archive
★ Research Software: a first class research output
★ Good practices for software curation
★ Conclusion
Software in Research: A pillar of Open Science

Multiple facets, it can be seen as:
- a tool
- a research outcome or result
- the object of research

Three pillars of Open Science
Gruenpeter, Software Heritage CC-By 4.0 2019
Much more complex than it seems

- **Structure**
  - monolithic/
  - composite;
  - self-contained/
  - external dependencies

- **Lifetime**
  - One-shot / long term

- **Community**
  - one person / one team/distributed community

- **Authorship**
  - complex set of roles

- **Authority**
  - institutions/organizations/communities/single person

Various granularities

- **Exact status** of the source code for *reproducibility*, e.g.
  - “you can find at swh:1:cnt:cdff19c4487c43c76f3612557d4dc61f9131790a4;lines=146-187 the core algorithm used in this article”

- **(Major) release** “This functionality is available in OCaml **version 4**”

- **Project** “Inria has created OCaml and Scikit-Learn”
A plurality of needs

Researchers
• **archive and reference** software used and created in articles
• **find** useful software
• **get credit** for developed software
• **verify/reproduce/improve** results

Research Organization
know its **software assets** for:
• technology transfer,
• impact metrics,
• strategy

Laboratories/teams
• **track** software contributions
• **produce** reports
• **maintain** web page
Four pillars: Archive, Reference, Describe, Credit

EOSC Scholarly Infrastructures for Research Software

- Chairs
  - Roberto Di Cosmo, Software Heritage, Inria
  - José Benito Gonzalez Lopez, Zenodo, CERN

Important policy tool in Open Science

- 9 infrastructures
  - 3 archives
  - 3 open access publishers
  - 3 aggregators

- recommendations
  - archive in Software Heritage,
  - use SWHID
  - open non profit
  - default to open source 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](https://doi.org/10.2777/28598)
L’ouverture des codes sources des logiciels est un enjeu majeur de reproductibilité des résultats scientifiques.

Deuxième Plan national pour la science ouverte

Définir et promouvoir une politique en matière de logiciels libres

- Établir une Charte nationale des logiciels libres issus de l’enseignement supérieur, de la recherche et de l’innovation.
- Développer le lien entre données et logiciels grâce au réseau des administrateurs des données, des algorithmes et des codes sources dans les établissements.
- Émettre des recommandations auprès des organismes financeurs pour accompagner au mieux le développement logiciel.
- Faire monter en compétence les structures de valorisation sur les modèles économiques associés à la production de logiciels libres.
- Soutenir Software Heritage et recommander son adoption pour l’archivage et le référencement des codes sources.

Reconnaître les codes sources comme une contribution à la recherche

- Créer un prix du logiciel libre pour la recherche qui récompense les équipes et projets exemplaires dans le domaine.
- Mieux valoriser les productions logicielles dans la carrière des chercheurs, des personnels d’accompagnement à la recherche et dans l’évaluation des structures de recherche.
- Suivre dans le temps la production de codes et logiciels de la recherche française pour en identifier les dynamiques, l’ouverture et les impacts grâce au baromètre de la science ouverte.
- Construire un catalogue des logiciels issus de la recherche en utilisant un schéma de métadonnées normalisé et partagé entre tous les acteurs de l’enseignement supérieur, de la recherche et de l’innovation.
Outline

- Introduction
- Preserving source code
- Software Heritage: the universal source code archive
- Research Software: a first class research output
- Good practices for software curation
- Conclusion
What is at stake? In order of difficulty

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

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

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

**Cite (for credit)**
- Research software artifacts must be properly cited (not the same as referenced!)
- to give credit to authors (evaluation!)
Archive

Submit code to HAL

- A scholarly repository
- An archive
- Software is more findable
- Transferring code to SWH

Save code now to SWH

- Easy (only submit repo’s url)
- All dev history archived
- different vcs are accepted
- PID to reference specific pieces of code (even algorithms)

Classic deposit

.zip, .tar.gz

SWHID deposit

COMING SOON

Save code now

Git, svn, hg

Morane Gruenpeter | UGA | Gricad, MaiMoSiNE et SARI | CC-BY - SA | 21/04/2022 | Page #47/66
HAL and SWH collaboration

Key dates

★ 2017 - Collaboration launch
★ Mars 2018 - Beta-test on HAL-Inria
★ Septembre 2018 - Launch on all HAL instances
★ April 2020 - BibLaTeX @software export
★ April 2022 - Deposit with SWHID (beta test on HAL-Inria)
  ○ Release on YouTube of the Open Science tutorial series: software source code deposit

The actors

HAL open science
CCSD
Inria
Software Heritage
Advantages:

★ **Metadata moderation** by the digital archivist team

★ **Export formats** available on the software record - to cite software

**Deposit guide:**
Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault, Alain Monteil. Create software deposit in HAL: User guide and best practices. [Technical Report] Inria; CCSD; Software Heritage. 2022. ([hal-01872189v2](https://hal.archives-ouvertes.fr/hal-01872189v2))

---

**Curated Archiving of Research Software Artifacts: Lessons Learned from the French Open Archive (HAL) on IJDC**

https://doi.org/10.2218/ijdc.v15i1.698
Save your any code now!

https://save.softwareheritage.org/

You can contribute to extend the content of the Software Heritage archive by submitting an origin save request. To do so, fill the required info in the form below:

- **Origin type**
  - git

- **Origin url**

Submit

A "Save code now" request takes the following parameters:
Advantages:

- **Metadata moderation** by the digital archivist team
- **Export formats** available on the software record - to cite software
- **Retrieve metadata with codemeta.json** to complete form
Reference - what do we need to identify?

Software concept / project / collection
Description in registry, a homepage or any other form of metadata record
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

Software artifact
- Executable (download link)
- Software source code
  - Dynamic artifact - current development code
  - Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit - a specific point in development history
  - Directory
  - File
  - Algorithm

Research Data Alliance/FORCE11 Software Source Code Identification WG et al. (2020). Use cases and identifier schemes for persistent software source code identification (V1.1). Research Data Alliance: https://doi.org/10.15497/RDA00053
Reference source code in SWH

Choose the granularity level for the reference:

code fragment
  ●  swh:1:cnt:c60366bc03936eede6509b23307321faf1035e23;lines=473-537
  ●  James McCaffrey’s algorithm in sageMath

specific version - release
  ●  swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f
  ●  release 2.3.0 of Darktable, dated 24 December 2016

full repository - snapshot
  ●  swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453
  ●  a snapshot of the entire Darktable repository (4 May 2017, GitHub)
Citation export in HAL

- Citation accessible on the HAL record
- Export BibTeX using the format BibLaTeX @software or @softwareversion (if a version property was submitted)
- Export used in activity reports for scientific outputs at Inria since 2020.

**HAL’s citation format**

Matteo Frigo, Mauro Zucchelli, Rachid Deriche, Samuel Deslauriers-Gauthier. TALON: Tractograms As Linear Operators in Neuroimaging. 2021, ⟨swh:1:dir:f25157ad1b13cb20ac3457d4f6f756b49ac63d079;origin=https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:snp:465d8956196578717f4cb5155e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70cccac:pa th=/⟩.

@software{frigo:hal-03116143v1,  TITLE = {{TALON: Tractograms As Linear Operators in Neuroimaging}},  AUTHOR = {Frigo, Matteo and Zucchelli, Mauro and Deriche, Rachid and Deslauriers-Gauthier, Samuel},  URL = {https://hal.archives-ouvertes.fr/hal-03116143},  NOTE = {},  YEAR = {2021},  MONTH = Jan,  SWHID = {swh:1:dir:f25157ad1b13cb20ac3457d4f6f756b49ac63d079;origin=https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:sn p:465d8956196578717f4cb5155e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70cccac:pa th=/},  VERSION = {0.3.0},  REPOSITORY = {https://gitlab.inria.fr/cobcom/talon},  LICENSE = {MIT License},  KEYWORDS = {diffusion MRI ; dMRI ; tractography ; python ; optimization},  FILE = {https://hal.archives-ouvertes.fr/hal-03116143/file/talon-source.zip},  HAL_ID = {hal-03116143},  HAL_VERSION = {v1},}
BibLaTex style extension for software


Describe - mandatory files for a HAL deposit

★ Prepare your code with the following files:

These files are verified by moderators

- **README** ([https://readme.so/fr/editor](https://readme.so/fr/editor))
- **AUTHORS** (containing list of authors)
- **LICENSE**
  - Open-source [SPDX compliant](https://choosealicense.com/) license
  - [https://choosealicense.com/](https://choosealicense.com/)
  - [https://reuse.software/](https://reuse.software/)
- **codemeta.json** (not mandatory but useful)

See also: [HOWTO archive and reference your code](https://reuse.software/)

Morane Gruenpeter | UGA | Gricad, MaiMoSiNE et SARI | CC-BY - SA | 21/04/2022 | Page #57/66
Describe: What’s a good README

★ MUST include:
  ○ Name and a description of the software.
★ SHOULD include:
  ○ how to run and use the source code
  ○ build environment, installation, requirements
★ CAN include:
  ○ project website or documentation pointer and recent news
  ○ visuals

extracted from Eric Steven Raymond and Make a README
Describe - choose a vocabulary?

Software schemes

General schemes

Software ontologies landscape from Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018).
(Gruenpeter & Thornton, 2018) CC-by-4
CodeMeta

- An initiative
  - An academic community discussing software metadata
- A vocabulary
  - A subset of schema.org
- A crosswalk table - mapping the metadata landscape

An open source tool to create codemeta.json files

Contributed to the community by

Software Heritage

The Great Library of Source Code

CodeMeta generator

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>My Software, the software title</td>
</tr>
<tr>
<td>Description</td>
<td>My Software computes ephemerides and orbit propagation. It has been developed from early '80.</td>
</tr>
<tr>
<td>Creation date</td>
<td>YYYY-MM-DD</td>
</tr>
<tr>
<td>First release date</td>
<td>YYYY-MM-DD</td>
</tr>
</tbody>
</table>

Use it directly on the CodeMeta [hosted version](#)

Contributions are welcome on the [code repository](#)
Describe: Software metadata terms

<table>
<thead>
<tr>
<th>Identify</th>
<th>administrate</th>
<th>classify</th>
<th>execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>- identifier</td>
<td>- maintainer</td>
<td>- description</td>
<td>- buildInstructions</td>
</tr>
<tr>
<td>- name</td>
<td>- citation</td>
<td>- keywords</td>
<td>- issueTracker</td>
</tr>
<tr>
<td>- author(s)</td>
<td>- funder(s)</td>
<td>- supportingData</td>
<td>- operatingSystem</td>
</tr>
<tr>
<td>- contributor(s)</td>
<td>- license</td>
<td>- referencePublication</td>
<td>- softwareRequirements</td>
</tr>
<tr>
<td>- version</td>
<td>- editor</td>
<td>- algorithms*</td>
<td>- runtimePlatform</td>
</tr>
<tr>
<td>- applicationCategory</td>
<td>- publisher</td>
<td>- readme (docs*)</td>
<td>- downloadUrl</td>
</tr>
<tr>
<td>- codeRepository</td>
<td>- dates</td>
<td></td>
<td>- memory, processor, storage</td>
</tr>
<tr>
<td></td>
<td>- created, modified, published</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- developmentStatus</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Advantages

- BibTeX export with type \texttt{@software}
- Citation.cff are also indexed in the SWH archive
Wrap up

★ 📦 Archive your code!
https://save.softwareheritage.org/

★ 👓 Describe your code with metadata
  README, LICENSE, AUTHORS, codemeta.json

★ 🗺️ Reference your code
  SWHID over DOI, context

★ 🎓 Cite your code
  Version, release, file, lines
The SWH ambassadors program

Who can be an ambassador?

★ **Enthusiastic individuals** who wish to volunteer as ambassadors to help grow the [Software Heritage community](#).

★ An ambassador can come from **different areas**: academia, cultural heritage, industry and public administration.

Do I need a technical background to become an ambassador?

★ Absolutely not! Source code archival is a cross-domain, cross-expertise concern.

★ you can have a technical background which might be helpful to read technical documentation and trying some of the more advanced features.
Thank you for your attention!

Keep in touch: morane@softwareheritage.org
@moraneottilia, @SWHeritage
https://www.softwareheritage.org/newsletter/