



HAL
open science

Archiver ses codes-sources et logiciels : SoftWare Heritage

Julien Caugant

► **To cite this version:**

Julien Caugant. Archiver ses codes-sources et logiciels : SoftWare Heritage. Semaine Data-SHS, Clément De Belsunce, Dec 2023, Aix-en-Provence, France. hal-04458345

HAL Id: hal-04458345

<https://amu.hal.science/hal-04458345>

Submitted on 14 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

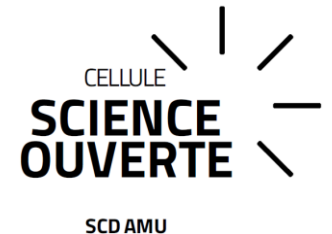
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Archiver ses codes-sources et logiciels : SoftWare Heritage

13/12/2023

Julien Caugant
SCD Aix Marseille Université



Présentation de la Cellule Science Ouverte (1)



La cellule Science Ouverte

Expertise sur les enjeux de la Science Ouverte et de l'édition scientifique

Ses actions :

- **Accompagner** et **sensibiliser** aux enjeux de la **Science Ouverte**
- Administrer **HAL-AMU** et former au **dépôt**
- Former à la **rédaction des plans de gestion de données** (PGD/DMP)
- Former, accompagner, **sensibilisation à la gestion et au dépôt de logiciels**



J. Caugant



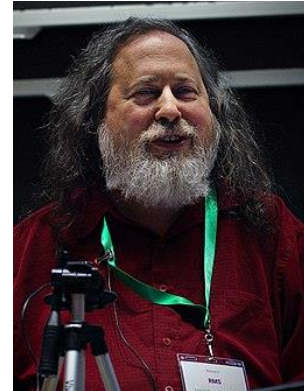
Sommaire

- Contexte
- Pourquoi archiver ses codes et logiciels?
- Comment archiver ses codes et logiciels?
- Points de vigilance

Contexte des codes et logiciels en Science Ouverte

Bref historique sur le logiciel libre

- Octobre 1976 : adoption de la loi du Copyright Act pour les logiciels
- 1984 : création du projet GNU par Richard Stallman, naissance du mouvement du logiciel libre et de la notion de copyleft en opposition au copyright



- 1985 : naissance des licences libres GNU
- 1998 : naissance du mouvement de l'open-source

Déclarations et appels en faveur de la Science Ouverte

Les déclarations et prises de position Initiative de Budapest pour l'Accès Ouvert [14-02-2002]

Déclaration de Berlin sur le Libre Accès à la Connaissance [22-10-2003]

Appel d'Amsterdam : plan d'action pour l'Open Science

Appel de Jussieu

Contexte européen Horizon Europe

European Open Science Cloud (EOSC)

Open Research Europe (ORE)

Coalition S et Plan S

Science Ouverte, données et codes

- 2011 : mission EtaLab, pour l'amélioration de l'action publique grâce aux données et à leur ouverture
« Le code source d'un logiciel détenu par une administration est un document administratif communicable. »
- 2018 : premier PNSO (Plan National pour la Science Ouverte), dont un des axes est de « Structurer et ouvrir les données de la recherche »
- 2021 : second PNSO dont le 3^{ème} axe est d'ouvrir et promouvoir les codes sources produits par la recherche

PNSO 2 : axe sur les codes et logiciels

TROISIÈME AXE : OUVRIR ET PROMOUVOIR LES CODES SOURCES PRODUITS PAR LA RECHERCHE

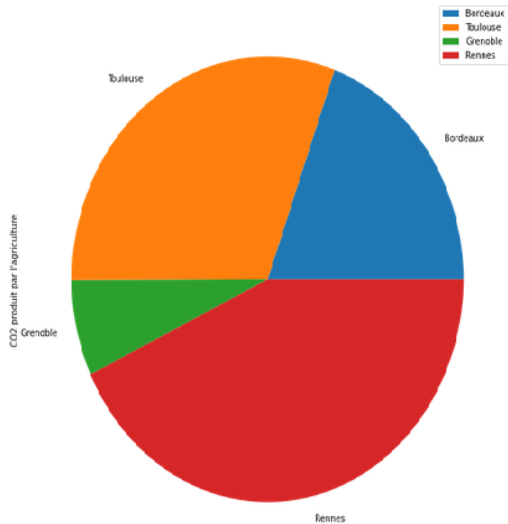
7. **Valoriser et soutenir la diffusion sous licence libre des codes sources** issus de recherches financées sur fonds publics
8. **Mettre en valeur la production des codes sources** de l'enseignement supérieur, de la recherche et de l'innovation
9. Définir et promouvoir une **politique en matière de logiciels libres**

Pourquoi archiver ses codes et logiciels?

Le code : outil ou objet de recherche?

```
Entrée [12]: 1 # A partir du dataset initial, récupération des valeurs pour 4 villes
2 data_bordeaux = df.loc[(df["Commune"]=="BORDEAUX").values[0, 2:12]
3 data_toulouse = df.loc[(df["Commune"]=="TOULOUSE").values[0, 2:12]
4 data_grenoble = df.loc[(df["Commune"]=="GRENOBLE").values[0, 2:12]
5 data_rennes = df.loc[(df["Commune"]=="RENNES").values[0, 2:12]
```

```
Entrée [13]: 1 df_bis = pd.DataFrame({'CO2 produit par l\'agriculture': [data_bordeaux[0], data_toulouse[0], data_grenoble[0], data_rennes[0]]
2
3     index=["Bordeaux", "Toulouse", "Grenoble", "Rennes"]
4 plot = df_bis.plot.pie(y="CO2 produit par l\'agriculture", figsize=(11, 11))
```



HAL
Nouvelle interface
fr Julien Caugant

Accueil
Dépôt
Consultation
Recherche
Documentation
Mon espace
Privilèges

hal-03275052, version 1

Software_PLoM_with_partition_2021_06_24

Christian Soize¹ [Details](#)

1 MSME - Laboratoire Modélisation et Simulation Multi-Echelle

Abstract : The software "Probabilistic Learning on Manifolds (PLoM) with Partition" is a novel version of the PLoM for which the first version of the algorithm was published in Ref. [1] and for which the mathematics foundations can be found in Ref. [2]. The present version of this PLoM software with partition is based on Ref.[3] and includes four novel capabilities: - probabilistic learning on manifolds with partition that consists (i) in computing, before the learning, an optimal partition in terms of independent random vectors (groups) using the algorithm presented Ref.[4] and (ii) in performing the probabilistic learning for each group of the identified partition. - parallel computing. - automatic identification of the smoothing parameter of the DMAP kernel as explained in Ref.[3]. - possibility to introduce constraints for preserving the normalization of the PCA coordinates during probabilistic learning process as explained in Ref.[3], based on Ref.[5]. Publications: [1] C. Soize, R. Ghanem, Data-driven probability concentration and sampling on manifold, Journal of Computational Physics, doi:10.1016/j.jcp.2016.05.044 [SMASH](#), 321, 242-258 (2016). [2] C. Soize, R. Ghanem, Probabilistic learning on manifolds, Foundations of Data Science, American Institute of Mathematical Sciences (AIMS), doi: 10.3934/fods.2020013 [SMASH](#), 2(3), 279-307 (2020). Also in arXiv:2002.12653 [math.ST], 28 Feb 2020, <https://arxiv.org/abs/2002.12653>. [3] C. Soize, R. Ghanem, Probabilistic learning on manifolds with partition, in arXiv:2010.14324 [stat.ML], 21 Feb 2021, <https://arxiv.org/abs/2102.10894>. Also submitted in International Journal for Numerical Methods in Engineering, 2021. [4] C. Soize, Optimal partition in terms of independent random vectors of any non-Gaussian vector defined by a set of realizations, SIAM-ASA Journal on Uncertainty Quantification, doi: 10.1137/16M1062223 [SMASH](#), 5(1), 176-211 (2017). [5] C. Soize, R. Ghanem, Physics-constrained non-Gaussian probabilistic learning on manifolds, International Journal for Methods in Engineering, doi: 10.1002/nme.6202 [SMASH](#), 121 (1), 110-145 (2020). This version allows for reproducing Application 1 of the paper: [3] C. Soize, R. Ghanem, Probabilistic learning on manifolds with partition, in arXiv:2010.14324 [stat.ML], 21 Feb 2021, <https://arxiv.org/abs/2102.10894>. Also submitted in International Journal for Numerical Methods in Engineering, 2021". The input data parameters entered for each STEP correspond to those for Application AP1 for which the results are in the directory "Results_API"

Type de document : [Logiciel](#)

Domaine : [Statistiques \[stat\]](#) / [Machine Learning \[stat.ML\]](#)

MÉTADONNÉES

Keywords : [Machine Learning](#) | [Probabilistic Learning](#)
[Probabilistic Learning on Manifolds](#) | [PLoM](#)

Licences
[Free License](#)

COLLECTIONS

CNRS | MSME | MSME_MECA | UPEC | UNIV-EIFFEL

CITATION

Christian Soize, Software_PLoM_with_partition_2021_06_24, 2021. (hal-03275052)

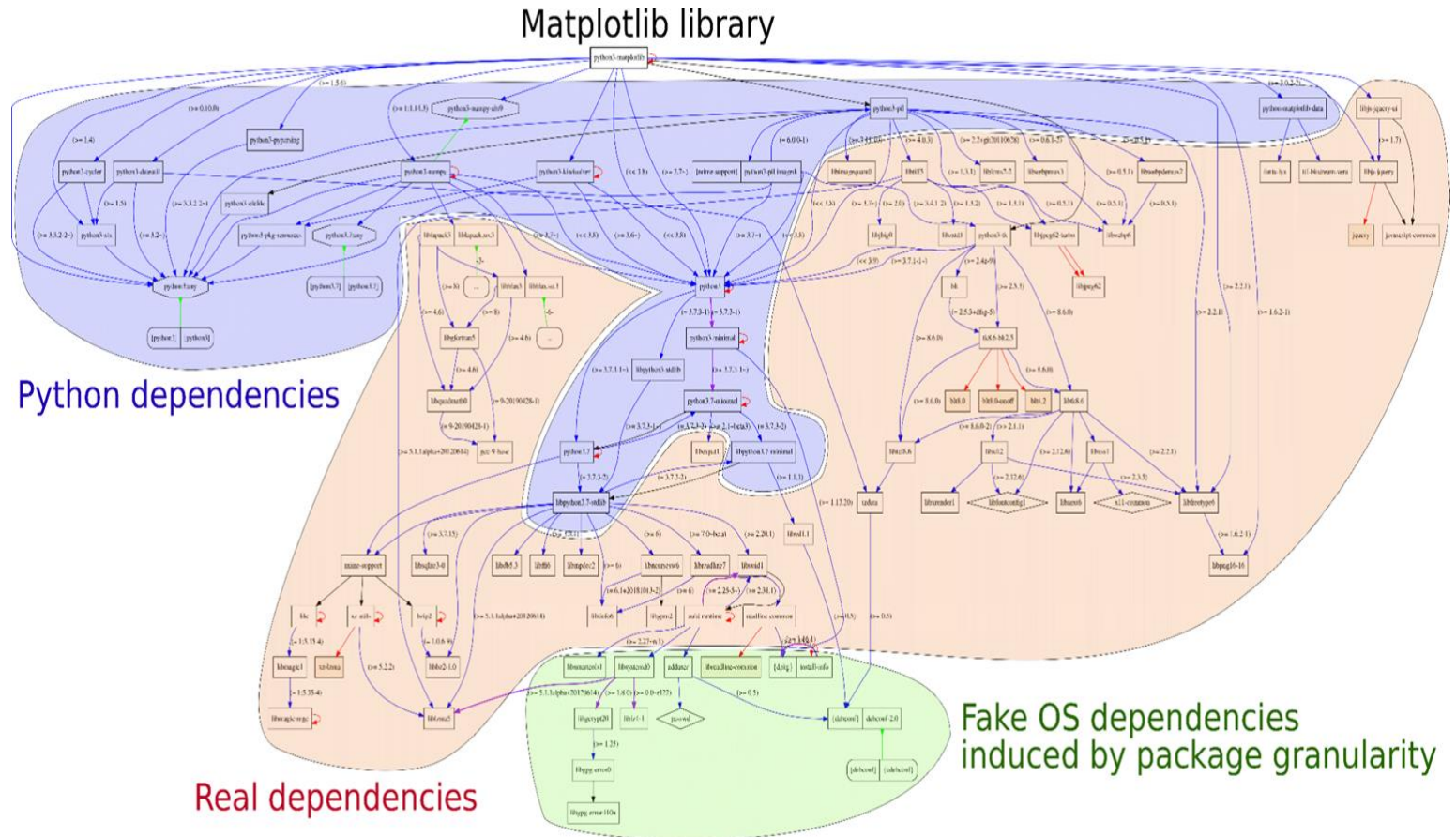
EXPORTER

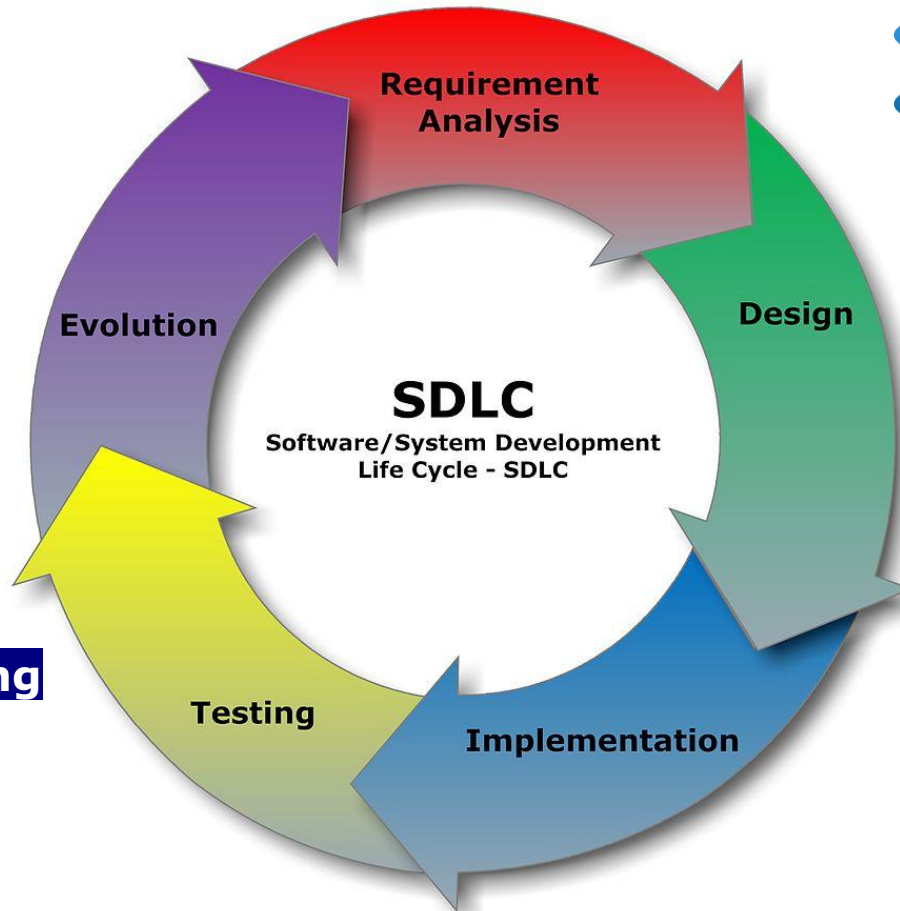
[CodeMeta](#) | [BibTeX](#) | [TEI](#) | [DC](#) | [DCterms](#)
[EndNote](#) | [Datacite](#)

PARTAGER

[f](#) [t](#) [w](#) [m](#) [+](#)

Un code pour les gouverner tous?





GitLab

Stockage / Versioning

Le code, une donnée menacée?



2015
Gitorious



25 Janvier 2016
Google Code



2020
Mercurial

Posted: Thursday, March 12, 2015

+1 377

Tweet 1,210

Like 404

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 own 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 simply isn't needed anymore.

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.

Sunsetting Mercurial support in Bitbucket

April 21, 2020 | 3 min read



Denise Chan

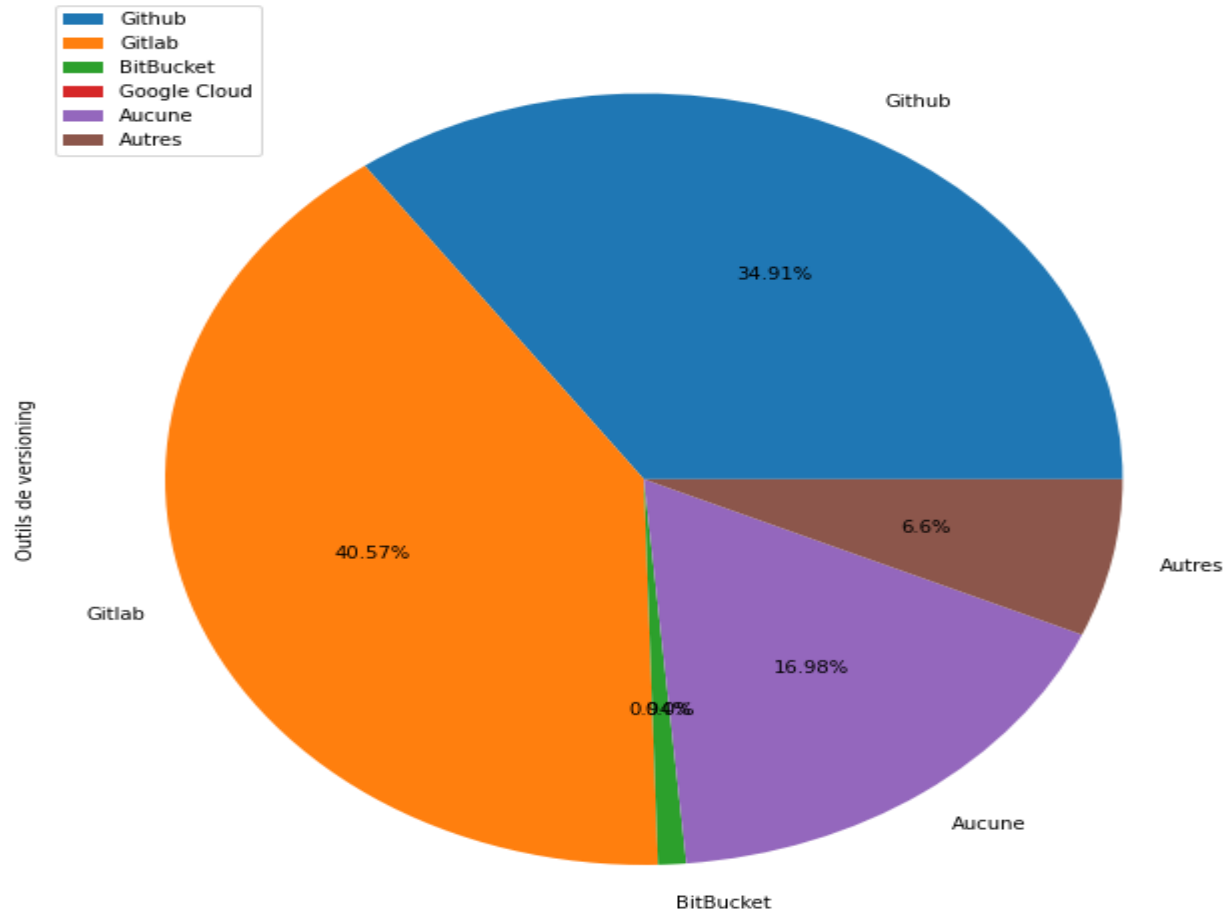
[Update Aug 26, 2020] All hg repos have now been disabled and cannot be accessed.

[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.

Quels intérêts pour le chercheur ou le développeur?

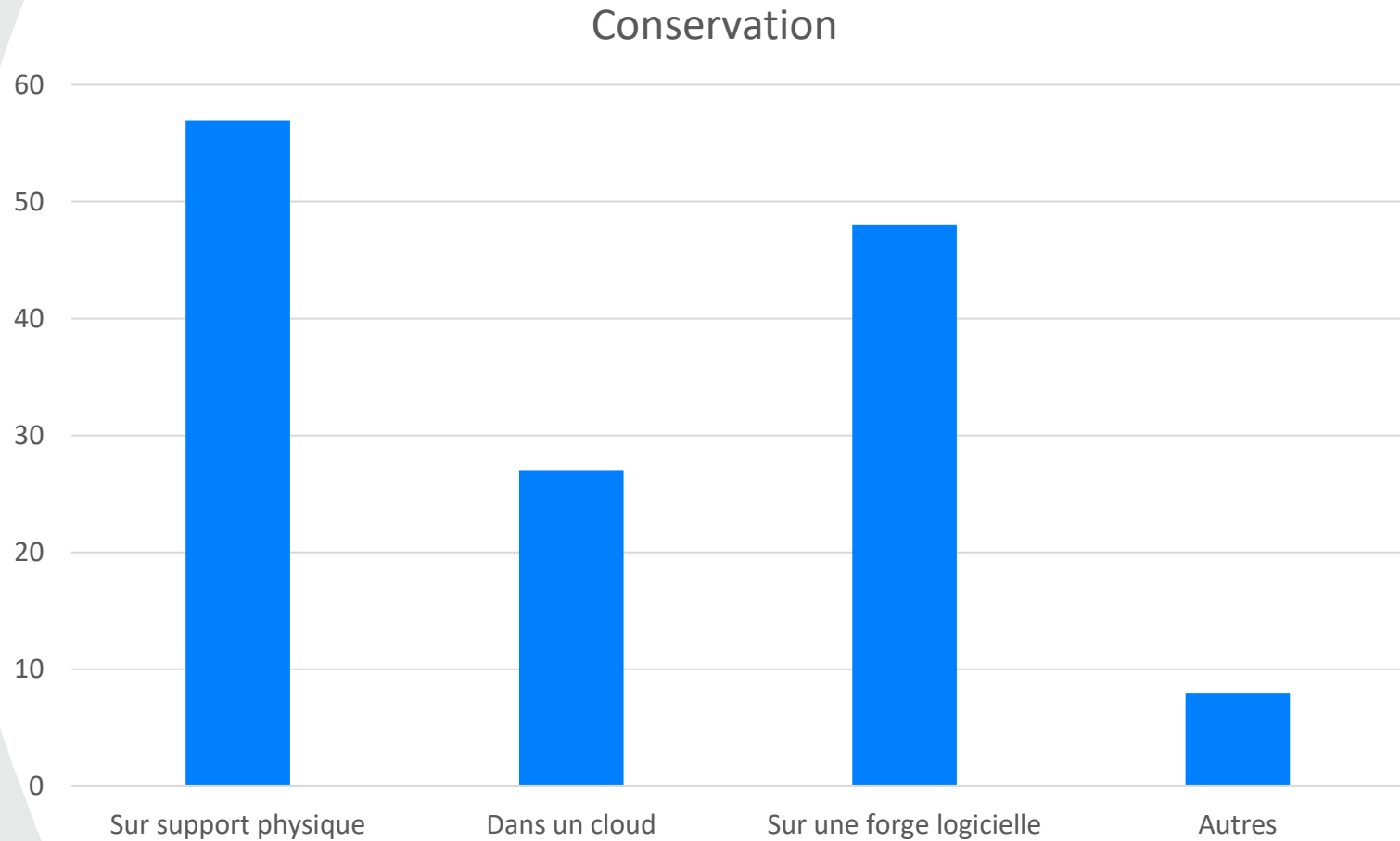
- Conservation pérenne
- Partage du code
- Reproductibilité et transparence
- Citation

Enquête AMU (2023) : Outils de versioning



- **Autres** : SourceSup de RENATER
- SourceForge.net
- Dataverse

Enquête AMU : Conservation du code à long terme



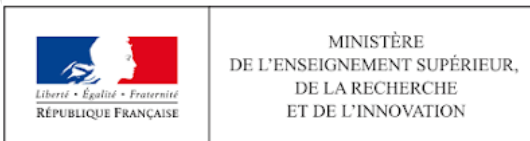
* Question à choix multiples, plusieurs réponses possibles



Comment archiver ses codes et logiciels?

SoftWare Heritage



















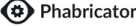



- Initiative à but non lucratif lancée en 2016, avec le soutien de l'INRIA, par Roberto Di Cosmo et Stefano Zacchiroli, en partenariat avec l'UNESCO ainsi qu'avec des acteurs du monde scientifique et de l'industrie (CNRS, Intel, Microsoft...)
- Cette archive contient aujourd'hui plus de 17 milliards de fichiers sources, et plus de 269 millions de projets logiciels



Les missions de SWH

SoftWare Heritage (SWH) a vocation à répondre à 4 défis majeurs : *récolter, organiser, préserver et rendre accessible* :

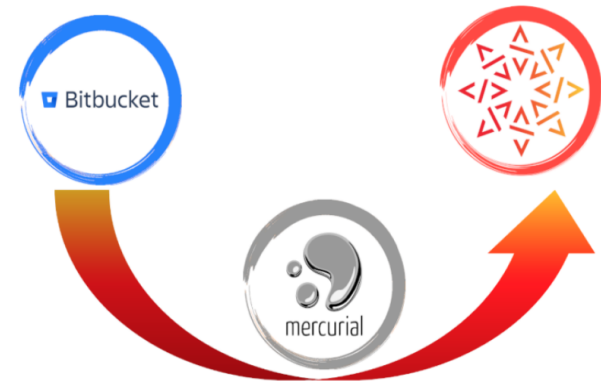
1. *Récolter* : SWH va moissonner les forges logicielles pour collecter et conserver les codes-sources. Il est aussi possible de déposer son code directement dans l'archive SWH ou bien dans HAL
2. *Organiser* : l'archive SWH va conserver le code de manière pérenne et permettre de signaler ce dernier sur HAL. Inversement, on peut déposer son code et le décrire sur HAL, et celui-ci sera transféré vers SWH. Pour permettre ce dépôt, certains fichiers sont nécessaires en supplément de ceux du code
3. *Préserver* : pour garantir un archivage pérenne, SWH repose sur une infrastructure dite distribuée et multipartite
4. *Rendre accessible* : les codes déposés (le code-source et non le code compilé) sur l'archive sont librement consultables et téléchargeables, s'inscrivant pleinement dans le cadre de la politique nationale de Science Ouverte. Une fois déposés, un SWHID leur est attribué afin qu'ils puissent être clairement identifiés et facilement retrouvables. Le dépôt dans HAL permet également de faciliter la recherche et la citation. On peut même citer une partie de code uniquement!

 2,444,377 origins	 56,982 origins	 23,400 origins
 26,217 origins	 135,119 origins	 38,408 origins
GitHub 187,808,011 origins	gitiles 9,988 origins	 4,129,099 origins
 1,707 origins	 122 origins	 874,791 origins
 354 origins	 1,180 origins	 497,363 origins
Maven 312,483 origins	 13,731 origins	 3,469,890 origins
 4,959 origins	 297,475 origins	 67,588 origins
 201 origins	 46,611 origins	 489,946 origins
 381,154 origins	stagit 208 origins	

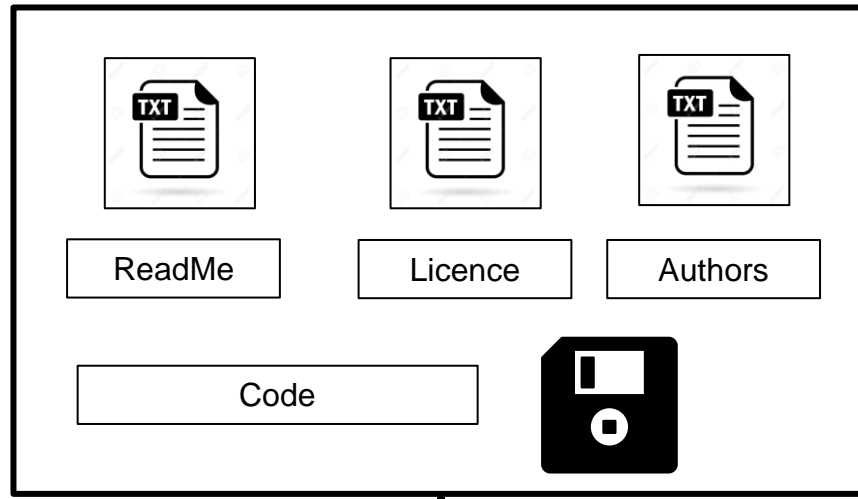


April 23, 2020

Rescuing 250000+ endangered Mercurial repositories



* Si dépôt via SWH, une fois l'archivage effectué, il faut signaler le dépôt via son SWHID sur HAL



Je crée mon fichier codemeta.json, et je le dépose sur ma forge avec mon code et les autres fichiers

Oui

Mon code est-il dans une forge ou un site de gestionnaire de versions de logiciel?

Non

Je place mes fichiers dans une archive (zip, tar...)

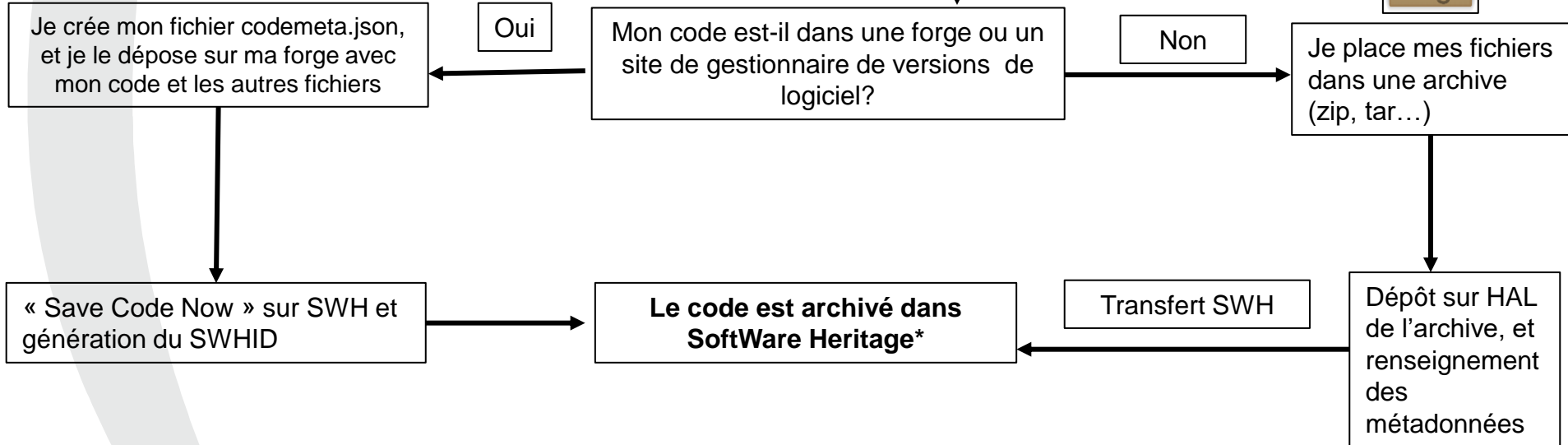


Dépôt sur HAL de l'archive, et renseignement des métadonnées

Transfert SWH

**Le code est archivé dans
SoftWare Heritage***

« Save Code Now » sur SWH et génération du SWHID



Plugin SWH pour faciliter le dépôt

Browser extensions

UpdateSWH: check and update archival of a repository

This tab shows the archival status of the repository

- Green: up to date
- Yellow: not up to date
- Grey: not archived yet
- Red: not archivable (private)

This handy browser extension allows you to seamlessly check if a repository that you are browsing is archived and up to date in Software Heritage.

Getting the extension

- + Identifiant pérenne pour le code
- + Possibilité de signaler un morceau de code

Points de vigilance

1. Veiller à la clarté et à la précision de son code

Notebook naming

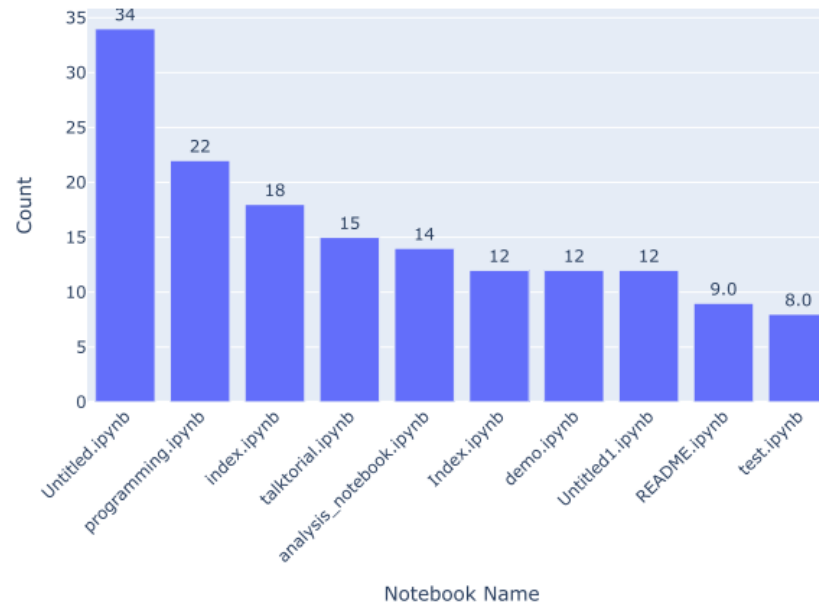



Figure 12. Most frequent notebook titles.

2. Déposer son code-source et non le code compilé

```
Code source écrit  
en langage C
```


```
main()  
{  
int var = 1;  
return 0;  
}
```



COMPILATION

```
Programme écrit  
en langage machine
```

```
0001000100  
0101011110  
1011000010
```



3. Une forge logicielle n'est pas une archive

Hébergement \neq archivage

Et après?

- Reconnaissance du développement de codes comme activité de recherche (avec dépôt dans une archive)
- Accompagnement des chercheurs lors des procédures de dépôts, et modération locale
- Mise en place d'un réseau de référents?
- Organisation de journées thématiques? D'ateliers virtuels?
- Autres besoins (forge logicielle AMU?)?

Quelques ressources

- Guide de dépôts et bonnes pratiques sur HAL :
<https://hal.inria.fr/hal-01872189v2>
- Obtenir des informations sur les licences logicielles :
<https://spdx.org/licenses/>
- Générer un fichier metadata.json :
<https://codemeta.github.io/codemeta-generator/>
- Bilan de l'enquête sur les pratiques de développement à AMU : <https://oaamu.hypotheses.org/4854>

Nous contacter : scd-science-ouverte@univ-amu.fr