

SoftWare Heritage (SWH) : archive universelle et pérenne du logiciel

Journées Mathrice IMT

Sandrine Layrisse

23 mars 2022



Découvrir le projet

En pratique

Missions

Archivage d'un logiciel en trois étapes

Parcours et Partage des logiciels

En ligne de mire

Conclusion



Le projet

2014 : projet ambitieux de Roberto Di Cosmo et de Stefano Zacchiroli créer une archive qui contiendrait et conserverait les codes sources de tous les logiciels publiquement disponibles

Le défi :

- ▶ quelle infrastructure
- ▶ périmètre mondial
- ▶ dispersion des codes sources



Dates clés du projet

- ▶ **2015** : Inria devient l'établissement porteur du projet
- ▶ **2016** : ouverture du service
- ▶ **2017** : signature d'un accord-cadre entre l'Inria et l'Unesco
- ▶ **depuis 2018**, Software Heritage est **interfacé avec HAL**.
- ▶ **2021** : célébration des 5 ans de Software Heritage au siège français de l'UNESCO et renouvellement de l'accord-cadre INRIA-UNESCO

Forces

- ▶ Conserve tout ce qui concerne les développements logiciels : les lignes de code, et tout l'historique des développements pour le logiciel (versions Git)
- ▶ Collecte régulièrement les environnements de développement logiciel ouverts (GitHub, GitLab, GUIX, npm, SOURCEFORGE, ...) basé sur un mécanisme implémenté par SWH.
- ▶ Complément aux plateformes comme Zenodo ou figshare, dédiées au partage des résultats de la recherche (publications et données de recherche,) mais qui ne sont pas dédiées au code.
- ▶ Autre objectif : se prémunir des disparitions des plateformes qui peuvent présenter des difficultés à être maintenues.



Collecter, Préserver, Partager des logiciels

- ▶ **Collecter** : archivage des logiciels avec leurs différentes versions (code source + documentation)
- ▶ **Préserver** : système de sauvegardes avec plusieurs copies de tous les logiciels collectés.
- ▶ **Partager** : accessibilité de tout logiciel via un identifiant unique attribué à chaque logiciel

<https://www.softwareheritage.org/mission/?lang=fr>

Archivage d'un logiciel

- ▶ Etape 1 - Préparer le dépôt public d'origine
- ▶ Etape 2 - Déposer le logiciel
- ▶ Etape 3 - Référencer le logiciel

<https://www.softwareheritage.org/save-and-reference-research-software/>

Archivage d'un logiciel

Etape 1 - Préparer le dépôt public d'origine

S'assurer que le code source est déjà hébergé sur un serveur accessible au public et qui utilise un des systèmes de contrôle de version pris en charge par Software Heritage (actuellement Subversion, Mercurial et Git).

<https://www.softwareheritage.org/save-and-reference-research-software/>

Archivage d'un logiciel

Etape 1 - Préparer le dépôt public d'origine

Inclure au niveau supérieur de l'arborescence du code source, les quatre fichiers suivants :

- ▶ un fichier "README" qui décrit le logiciel (nom, objectif, liens vers le site Web et la documentation, plate-forme de développement, informations de contact et de support, ...)
- ▶ un fichier "AUTHORS" (AUTEURS) qui liste toutes les personnes créditées pour le logiciel
- ▶ un fichier "LICENSE" ou un répertoire "LICENSES" (LICENSE/LICENCES) qui décrit les termes de la licence du projet
- ▶ Optionnel mais recommandé : un fichier "codemeta.json" qui permet d'indexer le code source dans l'archive Software Heritage



<https://www.softwareheritage.org/save-and-reference-research-software/>

Archivage d'un logiciel

Etape 2 - Déposer le logiciel

Une requête "Enregistrer le code maintenant" propose de choisir le type de système de contrôle de version utilisé pour le logiciel. Actuellement, les types pris en charge sont :

- ▶ **git**, pour les origines utilisant Git
- ▶ **svn**, pour les origines utilisant Subversion
- ▶ **hg**, pour les origines utilisant Mercurial

<https://archive.softwareheritage.org/save/>

Archivage d'un logiciel

Etape 2 - Déposer le logiciel

Une requête "Enregistrer le code maintenant" comprend l'URL d'origine du logiciel :

- ▶ Afin d'éviter d'enregistrer des erreurs dans Software Heritage, fournir l'**URL de clonage/extraction** telle qu'elle est donnée par le fournisseur hébergeant l'origine du logiciel.
- ▶ Par exemple, si vous souhaitez enregistrer une origine git dans l'archive de Software Heritage, vous devez **vérifier** que la commande **git clone <origin url>** ne renvoie pas d'erreur avant de soumettre la demande.

<https://archive.softwareheritage.org/save/>



Archivage d'un logiciel

Etape 2 - Déposer le logiciel

Une fois soumise, votre demande d'archivage peut être :

- ▶ acceptée : le chargement de son contenu dans l'archive sera alors programmée par Software Heritage dans les plus brefs délais
- ▶ rejetée : l'url d'origine fournie est alors blacklistée et aucune visite ne sera planifiée
- ▶ mise en attente : une revue manuelle sera alors effectuée afin de déterminer si l'origine peut être chargée en toute sécurité ou non dans l'archive

Vous pouvez suivre le statut des demandes dans la liste des demandes d'archivage soumises

<https://archive.softwareheritage.org/save/>



Archivage d'un logiciel

Etape 3 - Référencer son travail

Une fois votre code source archivé, plusieurs façons de le **référencer dans un article** avec l'association d'un identifiant unique persistant, le **SWHID** :

- ▶ ajouter un lien vers le **référentiel complet** archivé dans Software Heritage
- ▶ ajouter un lien vers une **version précise du projet** logiciel
- ▶ ajouter un lien vers une vous pouvez suivre son statut en cours dans la liste des demandes d'archivage soumises de code source, jusqu'au niveau de la **ligne de code**.

<https://www.softwareheritage.org/save-and-reference-research-software/>
<https://www.softwareheritage.org/howto-archive-and-reference-your-code/?lang=fr>



SoftWare Heritage - trouver et partager des logiciels

- ▶ Système intégré d'indexation des logiciels afin de les rendre rapidement accessibles.
- ▶ SWHID (HAL, citations diverses)
- ▶ Recherche sur le site web de SWH (pour l'instant en mentionnant un bout d'URL)

<https://archive.softwareheritage.org/>

En ligne de mire

- ▶ voeu de créer une fondation
- ▶ amélioration, optimisation des processus (projets de recherche)
- ▶ consolider, sécuriser
- ▶ rassembler une communauté internationale autour du projet

Conclusion

Coder

- ▶ avec des systèmes de gestion des versions
- ▶ avec des environnements de développements logiciels ouverts : collaboration, ouverture du code, traçage, visibilité, contributions
- ▶ prendre de bonnes habitudes : README/AUTHORS/LICENSE, conseils divers (bonnes pratiques en dev logiciel, présentations de Teresa pour les codes de recherche, ...)

Archiver, conserver et partager : SWH le fait pour vous !

- ▶ en collectant automatiquement tout ce qui est déjà public et valide au sens SWH
- ▶ ou en moissonnant vos dépôts quand vous le décidez



Merci de votre attention :-)

Remerciements à

- ▶ Roberto Di Cosmo (SWH-INRIA),
- ▶ Morane Gruenpeter (SWH-INRIA),
- ▶ Pierre Poulain (Institut Jacques Monod - Université de Paris CNRS , SWH Ambassador),
- ▶ Khodor Khadra (Institut de Math de Bordeaux)