

Comment évaluer la conformité d'un logiciel avec les principes FAIR ?

 openscience.pasteur.fr/2024/02/14/comment-evaluer-la-conformite-dun-logiciel-avec-les-principes-fair/

CeRIS - Institut Pasteur

14 février 2024

Nous vous présentions précédemment les principes FAIR pour les logiciels de recherche (*FAIR for Research Software – FAIR4RS – principes*). Mais pour **évaluer le degré d'implémentation des principes FAIR**, il est plus simple de disposer d'une checklist.

C'est ce que propose le projet FAIR-IMPACT dans son document *Metrics for automated FAIR software assessment in a disciplinary context*. Ce document définit **17 métriques** qui peuvent être utilisées pour mesurer le caractère FAIR d'un logiciel de recherche. Un des objectifs du projet est de traduire ces métriques en tests pratiques qui pourront être mis en œuvre de manière automatisée et intégrés dans les outils d'évaluation existants, comme par exemple l'outil F-UJI.

Ces 17 métriques peuvent déjà être utilisées comme une **checklist** pour évaluer la conformité de son logiciel avec les principes FAIR. Vous retrouverez les 17 questions à vous poser à partir de la page 11 du document. En voici une traduction en français, agrémentée de quelques commentaires en couleur :

Identifiants et métadonnées :

- Le logiciel a-t-il un identifiant unique et pérenne ?
- Les différents composants du logiciel ont-ils leurs propres identifiants ?
A noter que l'archive Software Heritage attribue des identifiants uniques et pérennes (*SoftWare Heritage persistent IDentifiers, SWHIDs*) aux différents « objets » logiciels.

- Le logiciel inclut-il des métadonnées descriptives qui aident à définir son objectif ?
- Le logiciel inclut-il des métadonnées de développement qui aident à définir son statut ?
- Le logiciel inclut-il des métadonnées sur les contributeurs et leurs rôles ?
- Les métadonnées du logiciel incluent-elles l'identifiant du logiciel ?

Accessibilité des métadonnées et du logiciel :

- Le logiciel dispose-t-il d'une notice de métadonnées publique, librement accessible et pérenne ?
Même si un logiciel n'est plus disponible (en raison du coût de maintenance, parce que le logiciel n'est plus utilisable en toute sécurité, parce que les dépendances ne sont plus disponibles...), les métadonnées doivent tout de même rester accessibles. Le dépôt du code dans HAL permet d'obtenir cette « notice de métadonnées ».
- Le logiciel est-il développé dans un dépôt de code / une forge qui utilise des protocoles de communication normalisés ?
L'obtention du logiciel ne doit pas dépendre d'outils ou de méthodes de communication spécialisés ou propriétaires. Le plus souvent, des protocoles techniques de communication normalisés sont utilisés pour accéder aux logiciels, tels que HTTPS.

Interopérabilité :

- Les formats des données consommées ou produites par le logiciel sont-ils ouverts et la documentation fait-elle référence à ces formats ?
- Le logiciel utilise-t-il des API ouvertes documentées de manière à ce que leurs fonctionnalités puissent être comprises par les machines ? (*machine-readable*)

Réutilisabilité :

- Le logiciel fournit-il des références à d'autres objets qui soutiennent son utilisation ?
Exemples : fichiers de paramètres nécessaires à l'exécution de certaines applications.
- Le logiciel décrit-il ce qui est nécessaire pour l'utiliser ?
Exemples : exigences, importations, bibliothèques... nécessaires pour compiler et exécuter le logiciel.
- Le logiciel est-il accompagné de scénarios de test démontrant son fonctionnement ?
- Le code source du logiciel inclut-il des informations sur la licence du logiciel et de tout logiciel externe intégré ?
- La notice de métadonnées du logiciel contient-elle des informations sur la licence ?

A noter que d'autres checklists existent pour **évaluer la qualité d'un logiciel**, notamment :

- Le programme OpenSSF (Open Source Security Foundation) Best Practices Badge : votre logiciel peut obtenir un badge s'il répond à un certain nombre de critères de bonnes pratiques de développement.
- La checklist proposée par le Software Sustainability Institute.
- La checklist proposée par le réseau EURISE (European Research Infrastructure Software Engineers).

Cet article comprend une traduction d'un extrait de : Chue Hong, N., et al. (2023). D5.2 – Metrics for automated FAIR software assessment in a disciplinary context (1.0 – DRAFT not yet approved by the European Commission). Zenodo. <https://doi.org/10.5281/zenodo.10047401>. Sous licence CC-BY.

