



HAL
open science

Les logiciels de la recherche et leurs licences : trois visions sur un objet

Teresa Gomez-Diaz

► To cite this version:

Teresa Gomez-Diaz. Les logiciels de la recherche et leurs licences : trois visions sur un objet. Master. Toulouse, France. 2019. hal-02434287v1

HAL Id: hal-02434287

<https://hal.science/hal-02434287v1>

Submitted on 9 Jan 2020 (v1), last revised 7 Apr 2023 (v2)

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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Les logiciels de la recherche et leurs licences : trois visions sur un objet

Teresa Gomez-Diaz

CNRS - Laboratoire d'informatique Gaspard-Monge

Contient des collaborations avec T. Recio (U. Cantabria) et G. Romier (CC-IN2P3)

Cette œuvre est mise à disposition selon les termes de la licence
Creative Commons Attribution - Pas d'utilisation commerciale - Pas de modification 4.0

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

CERFACS, Toulouse, 10 décembre 2019



Motivation (1/2)

Répondre à des questions :

- l'aspect légal et les caractéristiques des différentes licences
- choix de licence : critères à prendre en compte
- compatibilité des licences
- avantages/inconvénients d'une politique propriétaire ou open source
- détailler la notion de fork
- comment collaborer, comment se séparer

Nous mène à considérer trois visions de la **production de RS** :

- Philosophie : production d'une communauté avec valeurs
- Aspects juridiques : œuvre protégée (droit d'auteur, licences)
- Politique scientifique : production scientifique

Note : RS veut dire *research software* o logiciel de la recherche

Motivation : qui ?, quoi ?, comment ? (2/2)

Pour qui ? : *auteur, collaborateur*

Ce cours s'adresse aux **producteurs** de RS, personnels des laboratoires.

Ce qui est fait ? :

**utiliser, contribuer,
produire, diffuser,
modifier, inclure et re-diffuser**

Comment ? (faites-vous) : *en collaboration*

- dans un contexte de recherche, souvent international
- avec de personnes : qui peuvent avoir toutes sortes de statut étudiant, stagiaire, doctorant, post-doctorant, personnel salarié (ou pas), du même établissement, d'un autre établissement, du même laboratoire, d'un autre laboratoire, du même pays, d'un autre pays, personnel retraité...

Plan

1 Philosophie, production d'une communauté

- Définitions : free et/ou open source software
- Production d'une communauté
- Forks (fourches)

2 Aspects juridiques

- Définition de logiciel
- Le droit d'auteur : les œuvres et les logiciels
- Types de licences
- Compatibilité et héritage des licences
- Libre et propriétaire : c'est possible
- Contribuer à un logiciel
- Article vs. Logiciel : aspects légaux
- Procédure de diffusion

3 Aspects de politique scientifique

- Science ouverte et logiciels en France
- Article vs. Logiciel : aspects de politique scientifique
- PRESOFT : Software management Plan
- Évolution des pratiques d'évaluation de la recherche
 - Logiciel de la recherche (RS)
 - Auteur d'un logiciel de la recherche
 - Publication d'un logiciel de la recherche (RS papers)
 - Référence et citation
 - Protocole **CDUR** : évaluation de la recherche et RS

Définition de logiciel libre - free software

Selon la Free Software Foundation (FSF, 1985), fondée par R. M. Stallman, un logiciel est libre si ces quatre libertés sont garanties :

<http://www.gnu.org/philosophy/free-sw.fr.html>

- liberté d'exécuter le logiciel (tout usage),
- (*) liberté d'**étudier** et de modifier le fonctionnement,
- liberté de redistribuer des copies,
- (*) liberté de distribuer des copies de vos versions modifiées
 \implies (*) **condition nécessaire : accès au code source**

Exemples : T_EX by D. Knuth (1978), the Berkeley Software Distribution (BSD) by U. of California (1977-1995).

Garantie des quatre libertés : il faut une licence.

Un logiciel libre n'est pas *libre de droits* (protégé par CPI).

Un logiciel qui n'est pas libre se dit propriétaire (*privatif*).

Cette définition est née dans des milieux universitaires.

Définition de logiciel code ouvert - open source software

Selon l'Open Source Initiative (OSI), 1998, un logiciel est open source si sa licence respecte ces conditions :

<http://www.opensource.org/docs/osd>

1. Free Redistribution
The license shall not restrict any party from **selling** or giving away...
2. Source Code (⇒ **disponibilité du code**)
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

Cette définition est née dans des milieux proches des entreprises.
Besoin de licence pour garantir ces 10 conditions.

Droit, exemples et terminologie

FSF : freedom to **use, study, copy, modify**, and **redistribute** computer software (<http://www.gnu.org/>).

Du point de vue juridique (FR), les licences open source et les licences libres font intervenir les mêmes droits : **utiliser, modifier, redistribuer**.


Exemples de open source software mais pas free software :

- licence NASA v1.3
 - ▶ OSI : ok, voir <http://www.opensource.org/licenses/nasa1.3>
 - ▶ FSF : <http://www.gnu.org/licenses/license-list.fr.html#NASA>

L'accord open source de la NASA, version 1.3, n'est pas une licence de logiciel libre car elle comporte une clause exigeant que les modifications soient votre "création originale" ...

Nous vous demandons instamment de ne pas utiliser cette licence ...

- DRM : empêcher la modification d'un logiciel libre à utiliser dans un support numérique physique (changement du code exécutable).

 J'utilise souvent *libre*, mais aussi **FOSS, FLOSS**.

Communauté (milieu scientifique)

- ensemble d'une ou plusieurs personnes
- développeurs, experts scientifiques, parfois utilisateurs
- imprime sa personnalité, ses valeurs, sa philosophie
- création d'un RS, dans un contexte, avec un but (scientifique)
- s'exprime avec un vocabulaire
- historique : création, organisation, évolutions
- décisions, accords, conflits, gestion du "pouvoir"
- historique RS : création, étapes de développement, évolutions, forks

Vidéo : Pierre Mounier, Journée PLUME-CLÉO (16/06/2010)

https://webcast.in2p3.fr/video/l_outil_et_ses_usages

RS : production d'une communauté

Licence : exprime les valeurs de la communauté, elle doit être acceptée par tous, y compris par les nouveaux membres.

Important : cadre de collaboration scientifique

Forks ou fourches ou embranchements

- **Définition** : programme développé à partir des sources d'un autre
- plusieurs types, en fonction des raisons de la rupture
 - ▶ continuation d'un logiciel abandonné
 - ▶ évolution dans une direction différente
 - ▶ libération d'un programme sous licence propriétaire (droits)
 - ▶ propriétérisation d'un logiciel libre (droits)
 - ▶ divergence dans la gouvernance
- fork : problème ou solution ? technique ou de communauté ?
- le code original peut continuer à évoluer
- la communauté peut s'affaiblir (scindage) ou attirer des nouveaux
- futur, impact : difficiles à déterminer
- pertinence, sérieux, adhésion au nouveau projet...

Les fourches/forks, AFUL 2012, <https://aful.org/ressources/fourches-forks>

<https://www.projet-plume.org/fr/ressource/fourches-forks-dun-logiciel-libre-definition-propriete-conseils>

Voir aussi <https://www.projet-plume.org/en/news/build-a-community-practice-foss-projects-two-references>

- Build and sustain a community of practice : method applied to F/OSS projects, S. Ribas & M. Cezon, Novática N. 199, 2009.
- How to kill an Open Source community, S. Ribas, FOSSa 2009

Plan

1 Philosophie, production d'une communauté

2 Aspects juridiques

- Définition de logiciel
- Le droit d'auteur : les œuvres et les logiciels
- Types de licences
- Compatibilité et héritage des licences
- Libre et propriétaire : c'est possible
- Contribuer à un logiciel
- Article vs. Logiciel : aspects légaux
- Procédure de diffusion

3 Aspects de politique scientifique

Motivation

La diffusion des logiciels libres (2005)

Dominique Dalmas, Directrice juridique (CNRS)

Lyasid Hammoud, Juriste (CNRS)

*Il est regrettable de constater que les **aspects juridiques** sont encore trop souvent **méconnus et négligés** par les créateurs lors de la diffusion de leurs logiciels.*

Les surprises peuvent être douloureuses notamment en cas de litiges car les auteurs de bonne foi risquent de voir leur licence invalidée ou leur responsabilité mise en cause.

Il est pour eux essentiel de veiller à accompagner leur logiciel d'une licence bien construite, seule garantie que celui-ci vivra selon les principes qu'ils auront choisis.

Définition : qu'entend-on par **logiciel** ?

Définition de *logiciel* en tant qu'**objet juridique**

Selon l'article L. 112-2 du Code de la propriété intellectuelle (CPI) :
un logiciel est une œuvre de l'esprit protégée par le droit d'auteur.

[*] *Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données.*

D'un point de vue légal, un logiciel est une œuvre de l'esprit, avec un titre, des auteurs et des **droits** associés. C'est un concept large, qui contient le code source, le code compilé et éventuellement la documentation.

La définition qui s'applique n'est pas mathématique ni informatique, elle est **juridique**. Elle s'applique *inévitablement* (et bien malgré nous) dans toute sa dimension **lors de la diffusion** d'un logiciel.

[*] Arrêté du Ministère de l'Industrie du 22 décembre 1981 relatif à l'enrichissement du vocabulaire de l'informatique.

Le droit d'auteur des œuvres (1/2)

Les droits protégés par le Code de la propriété intellectuelle (CPI) sont automatiquement associés à l'auteur lors de la création de l'œuvre, sous condition de son **originalité** (ceci dépend de la date).

L'œuvre doit être **mise en forme** : les idées, les concepts ne sont pas protégeables.

Deux types de droits associés : droits moraux et droits patrimoniaux.

Droits moraux : ce sont des droits imprescriptibles, inaliénables, incessibles, ils sont en général associés à des personnes physiques (auteurs ou leurs héritiers). Il y en quatre :

- Droit à la paternité, relatif à la mention de l'auteur.
- Droit de divulgation, relatif au moment et aux conditions de livraison.
- Droit de repentir, permet de retirer une œuvre.
- Droit au respect de l'œuvre, permet de s'opposer aux modifications.

Le droit d'auteur des œuvres (2/2)


Droits patrimoniaux : concernent l'exploitation de l'œuvre, ce sont des droits monnayables, cessibles, temporaires.

On considère qu'il y a deux types d'exploitation :

- la représentation (par exemple d'une œuvre de théâtre) et
- la reproduction (musique sur CD par exemple).

Ce sont des droits associés souvent à des personnes morales (suite à des cessions effectuées par les auteurs), on parle alors des **détenteurs** des droits patrimoniaux, ou des **propriétaires**.

Terminologie :

- Œuvres orphelines :
il n'y a plus de personne physique associée aux droits moraux.
- Œuvres de domaine public :
fin des droits patrimoniaux, 70 ans après le décès de l'auteur.
 Ce terme est parfois (mal) utilisé dans le cadre de LL.

Le droit d'auteur du logiciel : traitement spécial

Pour les logiciels, il y a des **différences** :

- Droits moraux réduits : paternité.
- L'auteur ne peut (sauf stipulations contraires) s'opposer à la modification de l'œuvre ou exercer son droit de retrait.
- Les droits patrimoniaux (sauf stipulations contraires) sont dévolus à l'employeur. Cela s'applique aussi à leur documentation.
- Originalité : effort personnalisé allant au-delà de la simple mise en œuvre d'une logique automatique et contraignante.

Les détenteurs des droits patrimoniaux (propriétaires) d'un logiciel interviennent dans les décisions sur les licences.

La liste des détenteurs de ces droits est établie en fonction de :

- les auteurs
- leur statut et/ou le mode de collaboration
- les contrats : employeurs, collaboration, commande, conventions...
- ⚠ laboratoires : les accords entre tutelles (quadriennaux...)

Qui peut utiliser un logiciel ?

Art. L. 335-2 du CPI


Toute personne utilisant, copiant, modifiant ou diffusant le logiciel sans autorisation explicite des détenteurs des droits patrimoniaux est coupable de **contrefaçon** et passible de trois ans d'emprisonnement et de 300000 euros d'amende.

Source : T. Aimé, Guide pratique d'usage des logiciels libres dans les administrations, 2007
<https://www.projet-plume.org/ressource/guide-logiciels-libres-administrations>
Voir aussi : Directive 2009/24 EC du Parlement européen et du Conseil du 23 avril 2009

Les licences complètent le cadre juridique établi par la loi : s'il n'y a pas de droit explicitement donné, utiliser un logiciel relève de la contrefaçon.

Les licences sont des **contrats** et protègent les auteurs, les utilisateurs et les éventuels collaborateurs au développement.

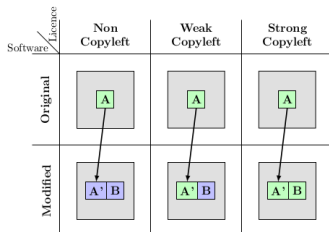
Elles octroient des **droits** (et des libertés) et peuvent contenir des clauses de réciprocité ou imposer des **obligations** qui sont à respecter.

 Pas de licence == Tous droits réservés.

Les types de licences libres/open source

- Copyleft fort (*diffusives*)
 - Licence initiale s'impose sur tout.
 - Obligation de réciprocité, évite de fermer un code libre.
- Copyleft faible (*persistentes*)
 - Licence initiale reste.
 - Ajouts peuvent avoir autre licence.
- Sans Copyleft (*évanescentes*)
 - Licence initiale ne s'impose pas.
 - Les dérivés peuvent avoir n'importe quelle licence.

- Copyleft fort : GNU GPL, CeCILL v2
- Copyleft faible : MPL, GNU LGPL, CeCILL-C
- Sans Copyleft : Apache, BSD, MIT, CeCILL-B



GPLv2 : *"You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License."*

Source, image : T. Aimé, Guide pratique d'usage des logiciels libres dans les administrations, 2007

<https://www.projet-plume.org/ressource/guide-logiciels-libres-administrations>

J.-L. Archimbaud, T. Gomez-Diaz, PLUME, 2009

FAQ : Licence & copyright pour les développements de logiciels libres de laboratoires de recherche

<https://www.projet-plume.org/ressource/faq-licence-copyright>

Compatibilité et héritage des licences (1/2)

Incompatibilité de licences : si deux licences imposent des obligations contradictoires (p. 230, Framabook B. Jean).

Quelques pistes :

- éditer, compiler, étudier, enregistrer votre code avec un outil libre ou propriétaire ne produit pas d'héritage de licence sur votre code ;
- il est possible de diffuser votre code XXXX (avec XXXX logiciel propriétaire) sous licence libre, mais ne distribuez pas XXXX avec ... indiquez à vos utilisateurs qu'il faut avoir XXXX pour l'utiliser, et sinon ils pourront au moins voir le code ;
- inclure beaucoup de briques dans votre logiciel peut créer des problèmes d'incompatibilité avec la licence que vous souhaitez pour le code final, voici deux tableaux de compatibilité de licences :
 - ▶ GNU : <http://www.gnu.org/licenses/gpl-faq.fr.html#AllCompatibility>
 - ▶ Annexe A, Framabook B. Jean (p. 315)

Compatibilité et héritage des licences (2/2)

- il est possible de diffuser un logiciel sous plusieurs licences, ce qui aide à traiter les problèmes d'incompatibilité ;
- il ne faut jamais modifier les informations des droits d'auteur ou de licence d'un logiciel récupéré, si ces informations ne sont pas claires, prenez contact avec les auteurs ou les responsables du projet ;
- si la licence (ou son absence) ne convient pas à l'usage prévu, prenez contact avec les auteurs, demandez une autre licence ou des accords (écrits) d'utilisation, modification...

Références :

- <http://www.gnu.org/licenses/gpl-faq.fr.html>
- Framabook Option Libre. Du bon usage des licences libres, B. Jean (2011)
<http://framabook.org/option-libre-du-bon-usage-des-licences-libres>
- Logiciels : FOSSology, OSLC, BlackDuck, ScanCode, Sourcetrail...
- Formats : SPDX, Open source cartouche

Libre et propriétaire : c'est possible

Il est possible de donner à un logiciel plusieurs licences.

Cela veut dire que l'utilisateur a la liberté de choisir le modèle qui s'adapte le mieux à chaque situation ou usage prévu.

Avoir des logiciels sous licences libres et propriétaires est donc possible.

Les licences propriétaires peuvent être accompagnées des contrats **signés** qui établissent (par exemple) les modalités de support et de collaboration et les retours économiques.

Impératif : à voir avec les services de valorisation.

Impératif : cadre légal cohérent (forks?).

Il est aussi possible de diffuser un code avec des modules sous licences différentes, par exemple un noyau de calcul sous licence libre et une interface graphique avec autre licence (utile pour les **RS**?).

Contribuer à un logiciel

Des logiciels comme GCC peuvent avoir une longue liste d'auteurs, il peut être très compliqué de les contacter tous face à un pb. légal.

La FSF pourra représenter les intérêts des développeurs si des cessions de droits sont faites (*disclaimer of rights, copyright holder*).

Des *Contributor Licence Agreements (CLA)* sont de plus en plus courants.

Réciproquement, si vous êtes responsable d'un projet logiciel, il est nécessaire de garder trace de tous les contributeurs, leur statut, et tous les contrats, conventions, financements, ... qui concernent le logiciel.

Faut-il des cessions de droits, des accords signés de licence ?

⚠️ droit anglosaxon \neq droit français, où toute cession de droits moraux ou de droits *d'œuvres futures* est invalide.

Références :

- <http://www.gnu.org/prep/maintain/maintain.html#Legal-Matters>
- <http://www.oss-watch.ac.uk/resources/cla>
- <https://www.projet-plume.org/ressource/diffuser-logiciel-recomm-juridiques-admin>

Si vous êtes ...

utilisateur d'un logiciel :

- avez-vous le droit d'utiliser, de modifier ?
- existe-il des logiciels libres répondant au besoin recherché ?

développeur de votre propre logiciel :

- donner des licences avant la diffusion du code
- attention aux briques incluses, leurs droits et licences

responsable d'un projet logiciel avec des collaborateurs :

- en plus : garder trace des contributeurs (avec statut)
- avoir une copie de tout contrat, convention...

contributeur à un logiciel :

- étudier les cessions de droits avant leur signature
- quelle loi s'applique (USA, France...) ?

Comprendre les RS : Article vs. Logiciel

Article vs. Logiciel : questions juridiques et de politique scientifique dans la production de logiciels

T. Gomez-Diaz (PLUME, 2011 et SIF, 2015), <https://hal.archives-ouvertes.fr/hal-01158010>

Aspects légaux		
	Article	Logiciel
Droit auteur	droits moraux, droits patrimoniaux	droits moraux réduits droits pat. dévolus à l'employeur
Œuvre	article	code source, code objet, doc., ...
Auteurs	signataires, même %	notion complexe, pb. légal , établir % de participation
Propriétaires	auteurs, même % cession des droits	tutelles en général, mais dépend du régime salarié , des contrats , ...
Dates	soumission, publication	matériel de conception, versions
Évolution	œuvre indépendante	œuvre indépendante ? il faut revoir auteurs, dates, lic., ...
Travaux préc.	références, citations	briques : compatibilité, héritage lic.
Diffusion	éditeur, web	web, forges, besoin de licence
Droits	lire, citer, ne pas copier	lire, ne pas utiliser , ..., besoin lic.
Licences	droits et obligations, CC (web)	droits et obligations, libres, propriétaires

C'est clair pour les articles. Il faut prendre des précautions pour les logiciels.

Procédure de diffusion des logiciels

S'adapte à chaque situation, valable pour les données.

- Choisir un nom, éviter les noms déjà utilisés, les marques.
- (*) Établir la liste des auteurs (avec % de participation), leurs affiliations.
- (*) Établir la liste des fonctionnalités principales.
- (*) Établir la liste des briques logicielles ou les données utilisées, avec licences.
- Choisir une licence, avec l'accord des auteurs et propriétaires des droits. Si possible : un accord signé. Attention à la compatibilité et héritage des licences.
- Choisir un site web, forge, dépôt pour la distribution. Indiquer les licences et les conditions d'utilisation, copie..., comment citer l'œuvre (voir Section 3).
- Créer et indiquer une adresse courriel de contact.
- (*) La traçabilité est importante, archiver en .tar.gz régulièrement.
- Informer la direction des laboratoires et les tutelles (si pas fait au point licence).
- Diffuser le logiciel et/ou les données.
- Informer la communauté cible, considérer les data ou les software papers.

(*) À revoir à chaque nouvelle version du logiciel.

T. Gomez-Diaz, (2014) Free software, Open source software, licenses. A short presentation including a procedure for RS...
<https://hal.archives-ouvertes.fr/hal-01062383v2>

T. Gomez-Diaz, PLUME (2010) Diffuser un logiciel de laboratoire : recommandations juridiques et administratives
<https://www.projet-plume.org/ressource/diffuser-logiciel-recomm-juridiques-admin>

Mettre en place une licence

La licence doit être mise en place **avant la diffusion** du logiciel.
Attention aux cahiers des charges et aux contrats (clauses PI, licences).

En-tête pour tous les fichiers :

- Nom du fichier, nom du logiciel
- Copyright (©, Droits patrimoniaux), année(s), p. morale ou physique aussi : tous droits réservés, quelques droits réservés
- Auteur(s), une adresse de contact
- Licence(s)
- Important : date de création, date de la dernière version
- Utile : format SPDX ou Open source cartouche, DOIs...

Et ajouter un fichier de licence (COPYING, LICENCE, README, ...) à l'ensemble des fichiers, avec le texte complet ou une URL.

En plus :

- Indiquer les briques logicielles utilisées et leurs licences.
- Indiquer clairement la licence (**et les auteurs**) dans la documentation, sur le site Web.

Plan

1 Philosophie, production d'une communauté

2 Aspects juridiques

3 Aspects de politique scientifique

- Science ouverte et logiciels en France
- Article vs. Logiciel : aspects de politique scientifique
- PRESOF : Software management Plan
- Évolution des pratiques d'évaluation de la recherche
 - Logiciel de la recherche (RS)
 - Auteur d'un logiciel de la recherche
 - Publication d'un logiciel de la recherche (RS papers)
 - Référence et citation
 - Protocole **CDUR** : évaluation de la recherche et RS

Science ouverte et logiciels en France

- Loi pour une République numérique (7 octobre 2016), article 30

Lorsqu'un écrit scientifique issu d'une activité de recherche financée au moins pour moitié...

- Le Décret n.2017-638 (27 avril 2017) donne la liste des licences applicables aux données, bases de données et logiciels, Site Etalab <https://www.data.gouv.fr/fr/licences>
- Les logiciels produits par les administrations sont passés en Open Source par défaut, cela comprend les logiciels développés par les chercheurs
Blog de Lionel Maurel (7 décembre 2017) :
<https://scinfolex.com/2017/12/08/les-logiciels-produits-par-les-administrations-sont-passes-en-open-source-par-...>
- Plan National pour la Science Ouverte (juillet 2018)
<https://www.enseignementsup-recherche.gouv.fr/cid132529/le-plan-national-pour-la-science-ouve-rte-les-resultats-...>

La France [...] soutiendra le développement et la conservation des logiciels, support indissociable des connaissances techniques et scientifiques de l'humanité tout entière.

-
- Site Etalab. Les licences applicables spécifiquement aux codes source de logiciels : obligation de choisir parmi ces licences lorsque on peut établir une licence de réutilisation
 - ▶ permissives : Apache-2.0, BSD-2-Clause, BSD-3-Clause, CECILL-B, MIT
 - ▶ obligation réciprocité : CECILL-2.1, CECILL-C, GPL-3.0, LGPL-3.0, AGPL-3.0, MPL-2.0

Comprendre les RS : Article vs. Logiciel

Article vs. Logiciel : questions juridiques et de politique scientifique dans la production de logiciels

T. Gomez-Diaz (PLUME, 2011 et SIF, 2015), <https://hal.archives-ouvertes.fr/hal-01158010>

Aspects relatifs à la politique scientifique		
	Article	Logiciel
Définition (L, T)	ok	à définir
Signature (C, T)	ok, déf. par tutelles	à définir (copyright) associer les laboratoires
Références (L, T)	HAL	PLUME
Liste des œuvres (L, T)	document à jour	document inconnu, PLUME peut être utile
Libre accès (C, L, T, CSI)	politique (+/-) ok, dépôt ok (HAL)	politique (lic.) à définir, dépôt à établir
Validation (C, L, T, CSI)	procédure <i>referee</i> , reproductibilité	à définir, validé au sens PLUME
Qualité/évaluation (C, L, T, CSI)	nb. citations	articles associés, attirer utilisateurs, contrats
Motivation (C, L, T, CSI)	recherche, article	recherche, pas le logiciel
Objet (C, L, T, CSI)	scientifique	3D : scientifique, potentiel de transf. de tech., obj. industriel

Seul point rouge pour les articles (reproductibilité) est lié à l'accès au logiciel associé.

PRESOFT : Software management Plan

- PRESOFT :
Preservation for REsearch SOFTware ou pérennisation de logiciels de la recherche
- Projet IN2P3, 2017-18
- IN2P3 (IdGC, CC-IN2P3), LIGM
- Plans de gestion de Logiciels ou Software management Plans (SMP)
- Modèle de Plan de Gestion de RS V3.2
T. Gomez-Diaz, G. Romier,
4 avril 2018, CC-BY-SA et CC-BY v4.0
<https://hal.archives-ouvertes.fr/hal-01802565/>
- Sur DMP OPIDoR (INIST)
<https://dmp.opidor.fr/>
- Document bilingue
- Outil d'une politique scientifique

PRESOFT : Preservation for REsearch SOFTWARE

Pérennisation de logiciels de la recherche

Vincent Breton Teresa Gomez-Diaz Geneviève Romier
 CNRS/IdGC CNRS/LIGM CNRS/CC-IN2P3

Objectif : étudier l'implémentation de plans de gestion de logiciels dans les unités de recherche afin d'améliorer leur pérennisation.

Le projet PRESOFT

- Projet IN2P3
- Dates : janvier 2017 - décembre 2018
- Partenaires : deux unités IN2P3 (IdGC, CC-IN2P3) et un laboratoire (LIGM)

Objectifs scientifiques

- Développer des modèles et méthodes adaptés aux besoins de la communauté de la recherche
- Étudier la faisabilité de plans de gestion
- Étudier les bénéfices, les bénéfices, les contraintes d'adoption
- Étudier l'impact
- Publier les travaux réalisés

Étapes

- Janvier 2017 - mai 2017 : premières livraisons des modèles et procédures
- Juin 2017 - décembre 2017 : validation des plans de gestion de logiciels
- recensement, étude et utilisation de modèles existants
- 2018 : Étudier l'impact des livraisons et du projet

Motivation

Plans de Gestion de Logiciels (DMP)

- Définir un cadre de travail pour la recherche
- Définir des protocoles de gestion de la recherche
- Définir des protocoles de gestion de la recherche
- Définir des protocoles de gestion de la recherche
- Exemple : le programme cadre de recherche européenne Horizon 2020

Software Management Plan (SMP)

- Document centré sur le logiciel
- Documente les détails de la recherche, y compris les données
- Favorise la communication et la collaboration

Plan de gestion de logiciel ou SMP

- Document utilisé pour la pérennisation de logiciel
- Facile de mise à jour pour gérer le cycle de vie du logiciel
- Document utilisé pour la recherche
- Facilite la mise à jour des données de la recherche
- Permet d'identifier les données, les acteurs, les responsabilités
- Permet de gérer la production, l'entretien
- Permet de contribuer avec un plan de développement

Bénéfices du SMP

- Une vue d'ensemble de chaque logiciel
- Une vue de la vie de chaque logiciel
- Une vue de la recherche
- Une vue de la recherche
- Une vue de la recherche
- Une vue de la recherche
- Une vue de la recherche
- Une vue de la recherche

Procédure

- Les étapes suivent le cycle de vie du logiciel
- Mise à jour de l'SMP à chaque étape
- Remplacer les données pertinentes
- Mettre à jour le plan de développement

Modèle PRESOFT de SMP

- Questions liées au logiciel
- Questions liées au projet
- Questions liées au projet
- Questions liées au projet

Références

Article en ligne : [opencade \(plan de gestion de la recherche\) dans la production de logiciels](#), T. Gomez-Diaz, 2018 - [Bibliothèque de la recherche en informatique et en automatique](#), vol. 178, no. 178

Une autre référence : [Open source software, Version 3. About generalization and generalization for research software and data documentation](#), T. Gomez-Diaz, septembre 2016, présenté à la Conférence ISSE 2016

Research Software Sustainability Project on Knowledge Exchange website, février 2016

The Software Sustainability Institute. Checklist for a Software Management Plan, 2016

Mots clés : logiciel de la recherche, pérennisation, préservation, accès libre, reproductibilité, science ouverte.

Présentation du Modèle de SMP PRESOFT V3.2

Le modèle est précédé d'une présentation qui indique comment l'utiliser.

0. Titre avec le nom du logiciel, suivi de : dates, auteur(s), affiliation(s)
 1. Métadonnées / Metadata
 2. Contexte du logiciel / Software context
 - 2.1 Historique / History
 - 2.2 Projet (s) lié (s) au logiciel / Project(s) related to the software
 - 2.3 Questions légales et politique de diffusion /
Legal issues and distribution policy
 3. Caractéristiques du logiciel / Software features
 - 3.1 Objectifs scientifiques / Scientific goals
 - 3.2 Objectifs d'utilisation et de diffusion / Usage and distribution objectives
 - 3.3 Caractéristiques techniques / Technical features
 4. Organisation de l'équipe / Team organisation
 5. Organisation du développement / Development organisation
 6. Organisation de la diffusion / Distribution organisation
 7. Gestion du plan de gestion / SMP management
- Références / References

Évolution des pratiques d'évaluation de la recherche

La production de logiciels n'est pas correctement prise en compte actuellement dans l'évaluation de la recherche, pourtant cette production peut prendre une place importante. Il est nécessaire de **faire évoluer les pratiques d'évaluation**, et cela dans le cadre des évolutions de la Science ouverte.

Gomez-Diaz T. and Recio T., F1000Research 2019, 8 :1353

On the evaluation of research software : the CDUR procedure

[version 1 ; peer review : 1 approved, 1 approved with reservations] (5 août 2019)

[version 2 ; peer review : 2 approved] (26 novembre 2019), <https://doi.org/10.12688/f1000research.19994.2>

Les concepts étudiés avant de proposer la procédure **CDUR** :

- Logiciel de la recherche (RS)
- Auteur d'un logiciel de la recherche
- Publication d'un logiciel de la recherche (RS papers)
- Référence et citation

Note : [*nb*] indique la référence dans l'article (en V2) avec ce *nb*.

Concept : logiciel de la recherche (RS) (1/2)

- [2] (1994) Partha D, David PA : Toward a new economics of science

*there may be **important positive spillovers** across projects in the form of “learning effects” [...] including the development of computer software for performing data processing, storage, retrieval and network transmission.*

- [16] (2011) Kelly D : An Analysis of Process Characteristics for Dev. Scientific Soft.

Scientific software is defined by (1) it is developed to answer a scientific question ; (2) it relies on the close involvement of an scientific expert ; and (3) it provides data to be examined by the person who will answer that question ...

- [12] (2011) TGD : Article vs. Logiciel : questions juridiques et de politique scientifique...
(2009) TGD : Guide laboratoire pour recenser ses développements logiciels (PLUME)

logiciel du laboratoire tout programme utile pour faire avancer la recherche, qui a été produit avec la participation d'un membre du laboratoire. Il arrive souvent que des publications de recherche soient associées.

- [18] (2012) Sletholt MT, Hannay JE, et al. : What Do We Know about Scientific Software Development's Agile Practices ?

software developed by scientists for scientists

- [19] (2016) Hettrick S : Research Software Sustainability

Research software is developed within academia and used for the purposes of research : to generate, process and analyse results. This includes a broad range of software, programs written by researchers for their own use.

- [10] (2018) NASA Committee : Open Source Software Policy Options for NASA Earth and Space Sciences

Research software – that is, the software that researchers develop to aid their science...

Concept : logiciel de la recherche (RS) (2/2)

- [16] (2011) Kelly D : An Analysis of Process Characteristics for Dev. Scientific Soft.
 - ▶ **exclut** ce qui peut être inclus dans d'autres définitions :
[...] control software whose main functioning involves the interaction with other software and hardware ; user interface software [...]; and any generalized tool that scientists may use in support of developing and executing their software, but does not of itself answer a scientific question.
 - ▶ l'importance de l'**exactitude** :
If the software gives the wrong answer, all other qualities become irrelevant
- [12] (2011) TGD : Article vs. Logiciel : questions juridiques et de politique scientifique...
les définitions ne tiennent pas compte de l'**état du RS** :
"en projet", "fini", diffusé, qualité, portée, taille, documenté, maintenu, utilisé uniquement par une équipe pour réaliser une publication, ou dans plusieurs laboratoires

Conclusion de la définition de RS :

- ce qui est fait : code (ensemble de fichiers, bien identifié),
- qui le fait : auteur(s), mais aussi contributeur ou expert scientifique,
- pourquoi faire : recherche, science, ie. article(s) associés,
- **important** : qualité des résultats scientifiques produits.

Concept : auteur d'un logiciel de la recherche

Que veut dire auteur d'un RS ?

[12] (2011) TGD : Article vs. Logiciel : questions juridiques et de politique scientifique...

- concept légal : écriture du code

l'auteur écrit le code

- concept scientifique : apport d'expertise

sans l'expert scientifique le code n'existerait pas

- il peut y avoir d'autres contributions

documentation, correction de bugs, test, maintenance...

Définition d'auteur d'un RS (article sur **CDUR**) :

- sélection de 3 rôles (limites peuvent être floues) :
 - ▶ (i) responsable du RS,
 - ▶ (ii) contributeur principal ou important (écriture du code),
 - ▶ (iii) contributeur mineur (écriture du code ou autre participation).

Les personnes qui n'écrivent pas du code peuvent avoir un % de participation décidé par l'équipe.

Concept : publication d'un RS (RS papers)

L'article étudie certaines publications dédiées aux RS, présentent les logiciels, avec des procédures de *software peer review*.

- Journal of Open Research Software (JORS)
- The Journal of Open Source Software (JOSS)
- Research Ideas and Outcomes (RIO)
- Software Impacts
- SoftwareX
- et aussi : (2010) Image Processing On Line Journal (IPOL)

Voir la liste de N. Chue Hong, Software Sustainability Institute (SSI)

<https://www.software.ac.uk/resources/guides/which-journals-should-i-publish-my-software>



Projet PLUME (France, 2006-2013) <https://projet-plume.org/> :

- publication de fiches RELIER (description courte de RS), avec lien vers les publications associées, 358 RS (fr), 116 RS (en)
- publication de fiches de *logiciel validé par la communauté de l'ESR au sens de PLUME* et mis en production sur au moins 3 sites, stats : 96 RS sur 406
- pas de *software peer review*
- classification thématique, mots clés, interface de recherche

Concepts : référence et citation

[39] (2013) Pontille D, Torny D : La manufacture de l'évaluation scientifique ...

[...] la différence entre référence et citation : l'acte de référence relève d'un auteur donné alors que la citation est une nouvelle propriété, éventuellement calculable, du texte source. Selon P. Wouters (1999), ce renversement a radicalement modifié les pratiques de référencement et littéralement créé une nouvelle "culture de la citation"

Une référence **fixe** nom, auteur, date, et identifie le RS en tant qu'objet scientifique.

L'article retient 3 types de référence :

- RS paper, avec *software peer review*,
- un article de recherche "classique" qui décrit le logiciel,
- une "référence" : auteurs(s), nom du RS, description courte, version, date, url.

À noter que :

- Il peut avoir plusieurs références associées à un RS.
- Identifications plus complètes : metadonnées, CITATION files...
- Software Citation Group, Software Citation Implementation Working Group.

Protocole **CDUR** : évaluation de la recherche et RS

L'article propose le protocole **CDUR** pour prendre en compte la production des logiciels de la recherche dans l'évaluation. Il est flexible pour s'adapter aux différents contextes d'évaluation, tout au long de la vie scientifique.

Il y a quatre étapes :

- (C) Citation** mesure la bonne identification du RS en tant que production de la recherche, implique nom, auteurs, version, dates... bien établies
point légal : auteurs, affiliations, participation
- (D) Dissemination** toute diffusion a ses objectifs et un public cible, ce point regarde les bonnes pratiques suivies
point Science ouverte, légal : licences
- (U) Use** point dédié aux aspects purement "logiciel" **du RS** : produit des résultats corrects, facilite la réutilisation
point reproductibilité, validation des résultats obtenus
- (R) Research** point dédié aux aspects purement "recherche" : qualité du travail scientifique, publications associées au RS, impact
point impact de la recherche

Flexibilité d'application : chaque ensemble décideurs/comité d'évaluation établi **sa propre procédure**, adaptée aux objectifs fixés et au contexte d'évaluation.

Conclusion

Nous avons étudié la complexité des logiciels de la recherche.
Leur production dans les milieux scientifiques fait intervenir trois aspects :

- 1 Philosophie, communauté
 - ▶ **RS** : objet produit par une communauté
 - ▶ **Licence** : expression des valeurs de la communauté
- 2 Juridique
 - ▶ **RS** : œuvre protégée par le droit d'auteur
 - ▶ **Licence** : contrat, donne droits et obligations
- 3 Politique scientifique
 - ▶ **RS** : production scientifique
 - ▶ **Licence** : outil de politique scientifique

Revision des questions

Répondre à des questions :

- détailler la notion de fork
- comment collaborer, comment se séparer
- l'aspect légal et les caractéristiques des différentes licences
- compatibilité des licences
- avantages/inconvénients d'une politique propriétaire ou open source
- **choix de licence : critères à prendre en compte**
- **Autres ?**

Références

- Le droit lié aux logiciels, par Valérie Hospital (DAJ, CNRS), 2011 -
https://www.projet-plume.org/files/20110615_rencontrescnrs_droitlogiciels_hospital.pdf
- Framabook Option Libre. Du bon usage des licences libres, B. Jean, 2011 -
<http://framabook.org/option-libre-du-bon-usage-des-licences-libres>
- Thème PLUME : patrimoine logiciel d'un laboratoire, 2009-2013 -
<https://www.projet-plume.org/patrimoine-logiciel-laboratoire>
- Cours Logiciels libres, T. Gomez-Diaz, 2012 -
<https://projet-plume.org/ressource/cours-logiciels-libres-fevrier-2012>
- European IPR Helpdesk Bulletin Issue (26) on Software protection, 2017 -
<https://www.iprhelpdesk.eu/news/european-ipr-helpdesk-bulletin-issue-26>
- Le Projet PLUME et le paysage actuel des RS dans la science ouverte, T. Gomez-Diaz, 2019 -
<https://hal-cnrs.archives-ouvertes.fr/hal-02069359>