

Synthèse journée thématique Production et Valorisation des Logiciels

Conseil scientifique INS2I

19 Sept. 2016

Résumé

Rédigée par le Groupe Thématique "Production et valorisation des logiciels" composé de : Anne-Catherine Letournel (animatrice), Christian Barillot, Isabelle Queinnec, Guy Gogniat, François Chaumette

Table des matières

| | | |
|----------|---|----------|
| 1 | Introduction | 2 |
| 2 | Les intervenants de la journée | 2 |
| 3 | Les trois formes d'un logiciel de laboratoire | 3 |
| 4 | La propriété intellectuelle et les attendus des tutelles | 3 |
| 5 | L'évaluation de la recherche | 4 |
| 6 | Conclusions | 5 |
| | Fiches pratiques | 6 |
| A | Comment se former aux outils d'hébergement de code ? | 6 |
| B | Fiche d'auto-évaluation du logiciel | 8 |
| C | Caractérisation simplifiée d'un logiciel | 9 |

1 Introduction

Lors du conseil du 9 mai 2016, la direction de l'INS2I a fait part d'une demande au CSI au sujet du transfert technologique, de la valorisation et de l'évaluation de la production logicielle. Il s'agit de faire un état des lieux de ce qui est fait dans les laboratoires de l'INS2I et ce qui serait à faire autour de la valorisation et du logiciel. En effet, la DIRE s'occupe de la valorisation plutôt focalisée sur les sciences du vivant et l'ingénierie, mais possède une expérience faible en valorisation logicielle. L'institut souhaite utiliser au mieux les moyens fournis par les écosystèmes locaux (SATT par exemple) ou par tout autre moyen extérieur aux unités, mais pouvant être complémentaire. Il demande au CSI des préconisations dans ce sens. Egalement, l'INS2I pense que les dépôts APP sont un bon indicateur de la production logicielle de ses laboratoires et souhaite mettre en place une politique incitative en la matière. Le faible nombre de dépôts APP ne reflète pas la richesse de la production logicielle et ne permet pas non plus de l'appréhender.

Il est intéressant de rappeler certains éléments de réflexion du mandat du précédent CSI sur la question de la recherche partenariale. En 2013, le CSI s'est penché sur les conditions de caractérisation de la recherche partenariale, comme nécessité économique, comme allers-retours constructifs entre théorie et applications, comme indicateur de l'adéquation du diplôme de docteur aux besoins de recrutement des entreprises. Un constat avait été fait sur les cultures différentes dans le monde industriel où l'on parle de 'produits' et dans le monde académique où l'on parle de 'problèmes'. Cela avait amené à proposer une lecture en deux axes pour la création d'une oeuvre logicielle par un laboratoire : le market-pull, en réponse à une demande exprimée par le milieu industriel ou le techno-push, pour valoriser des résultats dont les entreprises ignorent encore l'existence. Le market-pull faisant plutôt appel à des structures externes aux laboratoires mais associées au monde académique (comme les SATT) et le techno-push se basant lui, sur des moyens et méthodologies à mettre en oeuvre au sein des labos, sans alourdir les processus de production des logiciels.

Entre le logiciel réponse à une problématique scientifique et le logiciel produit sur étagère se situe la maturation. Les activités de maturation doivent donc être encouragées et soutenues dans les laboratoires pour inciter aux bons réflexes de qualité logicielle dès le démarrage des projets et pour aider les équipes à mettre en valeur leurs productions logicielles lorsque cela a un sens. Dans tous les cas, une forte motivation du chercheur est souvent indispensable à la réussite d'un transfert technologique, ce qui sous-entend pour le chercheur, d'une part la prise en compte d'un cadre législatif assez spécifique et souvent méconnu et d'autre part un investissement temporel important.

En conclusion, pour aller plus loin, la réflexion suggérerait de considérer comment les laboratoires traitaient ces questions aujourd'hui et avec quels moyens de soutien. Les initiatives des sections 6 et 7 affichant clairement la prise en compte de la production logicielle dans l'évaluation des dossiers des chercheurs sont un bon début, mais encore très insuffisant pour un chercheur, en comparaison du travail de publication scientifique.

2 Les intervenants de la journée

Afin de connaître comment sont traitées les questions de valorisation du logiciel dans les laboratoires de l'INS2I, le groupe de travail a organisé une enquête en une vingtaine de questions, envoyée à toutes les unités CNRS rattachées à l'INS2I, pour obtenir un premier aperçu de l'éventail des solutions mises en oeuvre. Les résultats ont été présentés en ouverture de la journée thématique [1].

Les liens avec les autres partenaires publics et acteurs régionaux de l'innovation sont à étudier pour ne pas être dans les laboratoires en opposition avec les partenaires, mais plutôt en complément. L'institut ne cherche pas à remplacer ce que font les SATT, la DIRE, BPI Banque Publique d'Investissement et OSEO, les pôles de compétitivité, les incubateurs etc ... Sur ce sujet, Marie-Anne St Jalmes, de la SATT Ouest Valorisation, est invitée pour présenter le rôle de la SATT dans le cycle complet de production et valorisation des logiciels [2].

L'évaluation de la production logicielle est au coeur de la question car elle semble mésestimée dans le système actuel. Si les chercheurs ne voient pas le résultat de leurs investissements dans le logiciel pris en compte, ils ne sont pas encouragés à y consacrer de l'énergie. Xavier Leroy, cher-

cheur Inria, membre de la commission d'évaluation Inria, est invité pour expliquer comment cette commission évalue la production logicielle lors des recrutements et promotion de ses chercheurs. Il présente également trois représentations possibles d'un logiciel : dispositif expérimental, autre forme de publication, objet de valorisation [3].

Enfin, pour le monde extérieur, le nombre de dépôts APP est un élément visible pour du patrimoine immatériel du CNRS. Malgré une réalité évidente de l'importance du transfert technologique dans la discipline des sciences de l'informations et de leurs interactions - le quart des startups du CNRS sont issues d'unités de l'INS2I d'après une enquête CNRS citée par W.Perruqueti- le dépôt APP est très peu pratiqué. L'institut souhaite mettre en place une politique incitative et demande l'avis du CSI. Magali Fitzgibbon est également invitée lors de cette journée thématique et son exposé [4] abordera cet outil à travers la notion plus générale d'actif du patrimoine immatériel que constitue le logiciel.

3 Les trois formes d'un logiciel de laboratoire

On propose la caractérisation simplifiée suivante pour tout logiciel de laboratoire :

- Lorsque le logiciel conserve une visibilité interne aux auteurs, c'est un prototype de test d'idées nouvelles (niveau 1, pas de choix de licence).
- Lorsque le logiciel est transféré gratuitement vers la communauté, la société ou l'enseignement, c'est une autre forme de publications que les publications scientifiques (niveau 2, plutôt une licence libre).
- Lorsqu'on parle de transfert technologique vers une application industrielle, c'est ce qu'on appelle communément "valorisation industrielle" (niveau 3, plutôt une licence propriétaire).

Cette classification n'est pas une échelle de maturation, le logiciel pouvant passer du niveau 1 au niveau 3 sans passer par le niveau 2, ou même exister partiellement avec une licence libre et partiellement avec une licence propriétaire.

Les solutions et les moyens de soutien à la production logicielle à privilégier sont très dépendants des trois catégories ci-dessus. Par exemple : accompagner et guider les ressources propres en ingénierie logicielle au niveau 1, fournir des ressources dédiées et pérennes (ingénieurs développement) pour constituer un soutien significatif de l'effort de développement au niveau 2, utiliser les dispositifs de maturation extérieurs aux laboratoires pour obtenir des ressources CDD au niveau 3.

Les plateformes matérielles portant du logiciel (type Equipex) ont déjà été évoquées lors d'une journée thématique dédiée en 2015 [5]. Elles sont un cas particulier de logiciel et sont concernées à ce titre par les préconisations évoquées dans la présente étude.

Les plateformes open-source collaboratives à visibilité mondiale (type logiciel de calcul SAGE ou COQ) sont un cas spécifique car le cadre juridique dépasse celui du CNRS. Les spécificités de valorisation des contributions à ce type de plateformes n'ont pas été évoquées lors de la journée.

L'identification des trois catégories décrites ci-dessus pourrait constituer une première étape dans un processus de cartographie du patrimoine logiciel des unités de l'INS2I.

4 La propriété intellectuelle et les attendus des tutelles

Les points d'intérêt suivants sont intrinsèques à l'objet 'logiciel' et sont incontournables pour mieux maîtriser un patrimoine logiciel :

- Le logiciel est un objet collaboratif par définition à cause des avantages et de la facilité de ré-utilisation du code.
- L'identification des parts respectives de contribution des différents auteurs ne fait pas partie implicitement du cycle de vie du logiciel.
- Dès qu'il est diffusé auprès d'un tiers, le logiciel a juridiquement besoin d'une licence. Une nouvelle fois, l'attribution d'une licence ne fait pas partie implicitement du cycle de vie du logiciel.
- Le brevet est rarement utilisé pour un logiciel, mais s'il l'est, il restreint la diffusion des technologies et ralentit le développement du logiciel.

La méconnaissance par les établissements employeurs des auteurs de leurs parts respectives est un frein à la maturation. Les solutions pour l'identification des auteurs passent par de l'outillage

adapté (forges de type Gitlab par exemple) dès le démarrage d'un projet ou de l'outillage adapté en post-traitement, pour identifier les composants ré-utilisés. **L'utilisation de tels outils doit être encouragée au moment de la production des logiciels.**

Pour aider les développeurs qui manquent de repères à s'initier aux bons outils, nous proposons en annexe A, une représentation simplifiée de l'écosystème des outils et forges d'hébergement de code. Des pistes sont indiquées pour aider à sélectionner les bons outils et démarrer avec l'utilisation d'un gestionnaire de versions logicielles.

Les questions de licences sont à traiter au cas par cas avec les auteurs, leurs employeurs et surtout, les acteurs locaux de l'innovation. Il n'y a pas de licence idéale ou par défaut qui puisse être préconisée dans l'absolu : licence propriétaire ou open-source ? Dans quels cas envisager une concession de licence ? Comment mettre en place une rétribution dans le cadre d'une collaboration avec un partenaire industriel ? La protection par brevet a-t-elle un sens pour mon projet ? **Avant sa diffusion, un logiciel doit forcément faire l'objet d'une réflexion sur la question de la licence et de sa protection.**

Autre constat : un découplage apparaît entre la gestion administrative et juridique du logiciel (par les SATT et les DR) et sa gestion plus scientifique (par l'institut et même les sections), qui conduit à une gestion morcelée et incomplète de la production et de la valorisation des logiciels. **Il est souhaitable d'encourager les interactions entre ces acteurs.**

5 L'évaluation de la recherche

- C'est le nerf de la guerre.
- il faut revaloriser la production logicielle.
- Il est difficile de prendre en compte tous les types de production logicielle.

Pour ses logiciels, Inria utilise un modèle d'évaluation multi-critères assez abouti basé sur l'audience, l'originalité, la maturité, l'évolution - maintenance et la distribution - licensing. Chacune de ces 5 catégories propose quatre à cinq degrés de réalisation parmi lesquels l'auteur choisit celui qui convient le mieux. Pour compléter, la contribution personnelle de l'auteur est caractérisée à la fois qualitativement (design-architecture / coding-debugging / maintenance-support) et quantitativement (pas impliqué / contributeur occasionnel / contributeur régulier) [6].

En annexe B, le modèle complet d'Inria est disponible. Il peut être repris tel quel dans le cadre de toute enquête ou la production logicielle est concernée. Il convient bien en particulier pour tout auteur déjà familier de la production et de la diffusion de logiciels.

La section 6 cite directement le référentiel Inria dans ses critères pour l'évaluation, le recrutement et la promotion des chercheurs et y ajoute des conditions précises notamment sur la disponibilité du logiciel à des fins de test [7].

La section 7 prend en compte la production de prototype et de logiciels à la fois au niveau de l'évaluation des contributions scientifiques mais aussi dans un critère spécifique de valorisation basé sur "l'impact technologique et/ou économique des activités de recherche sur contrat ou de consultation, dépôts de brevets, participation à des créations d'entreprises" [8]. Par ailleurs, la section 7 fournit des conseils pour la présentation de la production logicielle [9].

Les sections sont les relais naturels d'expression par les auteurs de la production logicielle des unités du CNRS et sont en première ligne pour favoriser cette expression au moment des dossiers de promotion notamment. Par ailleurs, s'adressant au public large et hétérogène de tous les candidats aux concours, elles sont aussi intéressées par cette information au moment du recrutement. Les sections 6 et 7 constatent malheureusement un faible taux de dossiers mettant clairement en valeur la production logicielle. Le modèle d'évaluation proposé par Inria, assez sophistiqué, peut décourager des auteurs occasionnels ou novices. C'est pourquoi le groupe de travail propose une grille alternative, plus épurée.

Suite aux réflexions générées par la journée thématique, et à l'analyse des grilles d'évaluations existantes, la proposition d'un modèle simple de qualification des logiciels, similaire à celui proposé par la section 7, spécifiant 3 niveaux de maturité, détaillés à la section 3 a l'avantage de couvrir la quasi-totalité de la production logicielle scientifique. Rapide à renseigner, il inciterait les auteurs à

citer davantage leurs logiciels quel que soit leur niveau de maturité et à les mettre en avant dans leur CV. **L'objectif de cette grille est d'être largement diffusée et utilisée dans tout formulaire institutionnel (INS2I, sections 6 ou 7, CID51 ou même CNRS tous instituts confondus etc ...), là où l'information est pertinente : crac, candidature concours, enquête, première étape dans une démarche de dépôt APP etc ...**

Ainsi, nous proposons, en annexe C, une grille très simple de caractérisation applicable à tout logiciel. Cette grille, plus sommaire que celle d'Inria, permet de cerner la catégorie de logiciel en donnant une première idée de sa maturité et de sa qualité.

6 Conclusions

Ce document présente un certain nombre de préconisations, reprises ci-dessous :

1. Cartographier le patrimoine logiciel des unités de l'INS2I :
 - En se rapprochant des DR pour consolider les données existantes de dépôts APP.
 - En incitant l'utilisation de référentiels de caractérisation des logiciels dans le cadre de procédures existantes : Crac, évaluations menées par les sections, dépôts APP ...
2. Encourager l'utilisation d'outils d'hébergement de code.
3. Inciter à la réflexion sur le choix des licences logicielles avec les acteurs du transfert technologique locaux
4. Développer les contacts entre les producteurs de code dans les laboratoires et les promoteurs d'innovation dans les structures locales (DR ou SATT...)

Quels dispositifs peuvent être activés, promus par l'INS2I ?

- Mise en oeuvre de missions doctorales pour aider aux dépôts APP dans les labos.
- Plans d'actions coordonnées sur un an ou deux avec communication ciblée auprès des labos : avant/en cours/après : cartographie des logiciels ou incitation à l'utilisation de forges ou éléments de réflexion sur le choix des licences....
- Création et animation d'un cercle interne à l'institut constitué de membres issus des laboratoires, en particulier de l'activité "appui/soutien logiciel à la recherche", lorsqu'elle existe, pour formaliser un socle commun.
- Nomination de correspondants valorisation, développeurs logiciels internes au laboratoire, chargés de relayer et de diffuser ce type d'actions sur le terrain, au plus près des acteurs.

Fiches pratiques

A Comment se former aux outils d'hébergement de code ?

Les deux tableaux 1 et 2 donnent quelques clés pour positionner la problématique de l'hébergement des codes sources, qui est la première question à se poser pour produire un code potentiellement valorisable.

Pour acquérir une formation dans ce domaine, quatre types de ressources sont possibles :

- Utiliser la documentation native du logiciel en ligne.
- Utiliser un tutorial plus pédagogique. OpenClassroom en propose une grande variété par exemple.
- S'adresser à la communauté locale des développeurs logiciels, s'il elle existe à proximité. Le réseau DevLog fédère des réseaux plus locaux qui proposent régulièrement des journées d'initiation à ces outils : devlog.cnrs.fr.
- Faire appel à des sociétés de formation qui dispensent des prestations payantes sur catalogue.

| | | | | | |
|--------------------------|---|--|--|--|-----------|
| Gestionnaire de version | CVS, un des plus anciens | svn | Team Foundation Server | git | mercurial |
| typologie | le stockage des dépôts se fait sur un seul serveur centralisé | | Solution propriétaire Microsoft | architecture des serveurs de dépôt répartie. Clonage aisé des dépôts | |
| documentation officielle | | svnbook.red-bean.com | www.visualstudio.com/fr/tfs | git-scm.com/docs | |
| Tutoriels open-classroom | | gerez-vos-projets-a-l-aide-du-gestionnaire-de-versions-subversion | | gerer-son-code-avec-git-et-github | |

TABLE 1 – Table des principaux gestionnaires de version

| | | | | | | |
|--|--|--|--|--|--|--|
| Site web de la forge | gforge.inria.fr | sourcesup.renater.fr | sourceforge.net | forge git de laboratoire | github.com | bitbucket.org |
| gestionnaire de la forge | Inria | GIP Renater : communauté académique française | Geeknet (US) | à installer soi-même et gérer chez soi | GitHub (US) | Atlassian (Australia) |
| Forge utilisée proposant aussi wikis, bug tracking, forums ... | Gforge | FusionForge | SourceForge | GitLab | | |
| gestionnaires de version sous-jacents supportés par la forge | svn, git... | svn, git... | svn, git... | git only | git only | git, mercurial |
| particularités | seul un agent inria peut créer un dépôt | offre de service Renater | inscription et charte d'utilisation | à définir soi-même | inscription libre | |
| dépôts publics ou privés? | les deux sont possibles | les deux sont possibles | public, pour logiciels open-source, attention publiciels | | gratuit si publics | |

TABLE 2 – Table des principales forges logicielles

Fiche d'autoévaluation du logiciel (modèle Inria)

Software Name :

Summary:

| | | | | | | | | | |
|---|----|----|----|-----|---|----|----|----|-----|
| A | SO | SM | EM | SDL | / | DA | CD | MS | TPM |
|---|----|----|----|-----|---|----|----|----|-----|

B.1 - Caractériser le logiciel

Critères A (Audience)

- A-1 Internal prototype.
- A-2 Used by people in the team or close to the team.
- A-3 Ambitious software, usable by people outside the team.
- A-4 Large audience, used by people outside the team.
- A-5 Wide audience, large user's community.

Critères SO (Software originality)

- SO-1 None.
- SO-2 Minor contributions to existing software, reusing known ideas.
- SO-3 Original software reusing known ideas and introducing new ideas.
- SO-4 Original software implementing a fair number of original ideas.

Critères SM Critères SO (Software maturity)

- SM-1 Demos work, loose documentation.
- SM-2 Basic usage works, terse documentation.
- SM-3 Well-developed software, good documentation, reasonable software engineering.
- SM-4 Major software project, strong software engineering.
- SM-5 High-assurance software, certified by an evaluation agency or formally verified.

Critères EM (Evolution and Maintenance)

- EM-1 No real future plans.
- EM-2 Basic maintenance to keep the software alive.
- EM-3 Good quality middle-term maintenance.
- EM-4 Well-defined and implemented plans for future maintenance and evolution.

Critères SDL (Software distribution and licensing)

- SDL-1 None.
- SDL-2 Privately distributed within the close community.
- SDL-3 Distributed to industrial partners in a contractual setting.
- SDL-4 Public source or binary distribution on the Web.
- SDL-5 External packaging and distribution, as part of a popular open source distribution or a commercially-distributed product.

B.2- Caractériser la contribution de l'auteur

Design and Architecture (DA), Specify if you are :

- DA-1 Not involved
- DA-2 Nn occasional contributor a
- DA-3 Regular contributor
- DA-4 A main contributor

Coding and Debugging (CD), Specify if you are :

- CD-1 Not involved
- CD-2 An occasional contributor
- CD-3 A regular contributor
- CD-4 A main contributor

Maintenance and Support (MS), Specify if you are :

- MS-1 Not involved
- MS-2 An occasional contributor
- MS-3 A regular contributor
- MS-4 A main contributor

Team/Project Management (TPM), Specify if you are :

- TPM-1 Not involved
- TPM-2 An occasional contributor
- TPM-3 A regular contributor
- TPM-4 A main contributor

B.3 Complements d'information

| | | |
|---|--|--|
| Software's web site (if available): | | |
| Objective of the software | <i>Domain addressed and precise technical goals</i> | |
| | <i>Type of the software (autonomous, part of a library, part of a larger system)</i> | |
| | <i>Performance goals, functionalities/algorithms, dissemination, etc</i> | |
| Users community : type (research, education, commercial/industrial, large open source community, etc.), size, how the community is animated, etc. | | |
| Your assessment of the software's impact in your research community and your user community. | | |
| Description of the state of the art, placement of the software w.r.t. the competition. | | |
| Size of the software (lines of code), languages used, size of the development team, development effort, list of patents and main publications related to the software. | | |
| Your contribution to the software (as a team or as an individual). | | |
| Any meaningful additional information | | |

C Caractérisation simplifiée d'un logiciel

La fiche, unique pour un logiciel rassemble les informations suivantes :

- Intitulé du logiciel
- Problématique résolue
- Page web / documentation
- Licence
- Date du dernier dépôt APP
- Auteurs
 - Nom - Prénom
 - Institution
 - Participation et pourcentage : conception, développement...
- Date de création
- Historique des versions
- Contexte de réalisation : projet de recherche académique, contrat industriel, collaboration locale, nationale, internationale ...

Références

- [1] AC. Letournel. un état des lieux, sept 2016.
- [2] MA. Saint Jalmes. Production et valorisation des logiciels, sept 2016.
- [3] X. Leroy. L'évaluation du logiciel à Inria, sept 2016.
- [4] M. Fitzgibbon. Atelier Thématique "Transfert et Valorisation Logicielle", sept 2016.
- [5] M. Beaudouin-Lafon. Plateformes d'envergures en Sciences de l'information, nov 2015.
- [6] Inria Evaluation Committee. Criteria for software self-assessment, 2015.
- [7] Section 6 du comité national du CNRS. Critères pour l'évaluation, le recrutement, la promotion des chercheurs, dec 2016.
- [8] Section 7 du comité national du CNRS. Critères d'évaluation de la section 07 proposés pour la mandature 2016 - 2021, dec 2016.
- [9] Section 7 du comité national du CNRS. Conseils de la Section 7 du CoNRS pour la présentation des productions logicielles, oct 2016.