

Formation :
“Les Plans de Gestion des Logiciels de la Recherche”

Questions et Réponses sur le Modèle PRESOFT et les Plans de Gestion

T. Gomez-Diaz (CNRS/LIGM), V2, 29 juin 2026



*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/>

Préambule. L'objectif de ce document est de reprendre une partie des questions et commentaires qui ont été échangés lors des formations :

- « **Les Plans de Gestion des Logiciels de la Recherche** »¹, 23-24 mai 2024,
 - « **Les logiciels de la recherche et leurs licences : trois visions sur un objet avec un atelier sur les Plans de gestion de logiciels** »², 1 et 8 juin 2026,
- qui ont eu lieu à l'Université Gustave Eiffel (UGE), Paris-Est.

Les objectifs de ces formations ont été :

- Comprendre le contexte de production et de diffusion des logiciels de la recherche
- Connaître le modèle de Plan de gestion des logiciels de la recherche PRESOFT
- Savoir effectuer son propre Plan de gestion de logiciels
- Mieux connaître le panorama des logiciels de la recherche en France et à l'international

Ces formations ont inclus un atelier où les participants ont rempli leur propre plan de gestion de logiciels en utilisant le modèle PRESOFT, qui est disponible sur HAL³ et sur Zenodo⁴ (en formats odt et pdf).

Tous les supports des formations et des liens vers des documents et des informations complémentaires sont sur les pages web mentionnées. La formation de 2026 a inclus le cours : « *Les logiciels de la recherche et leurs licences : trois visions sur un objet* »⁵ qui traite en profondeur les questions juridiques et de politique scientifique sur les logiciels de la recherche. Le modèle PRESOFT a une page web⁶ qui contient des informations dédiées aux Plans de Gestion de Logiciels, y compris de liens vers d'autres modèles et vers les publications et les recommandations connues sur le sujet.

Voici la liste des questions et des réponses.

Point faut-il remplir toutes ces questions ? Comme le montre la publication de Yves Vincent Grossmann, Giacomo Lanza, Katarzyna Biernacka, Tim Hasler, Kerstin Helbig (2024), *Software Management Plans – Current Concepts, Tools, and*

1 <http://igm.univ-mlv.fr/~teresa/presoft/2024FormationPlansGestionLogicielsRecherche/>

2 <http://ligm.univ-eiffel.fr/~teresa/presoft/2026FormationLogicielsRecherche/>

3 <https://hal.science/hal-01802565v1>

4 <https://doi.org/10.5281/zenodo.1405614>

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

6 <https://ligm.univ-eiffel.fr/~teresa/presoft/>

Application. Data Science Journal⁷, le modèle PRESOFT est un des plus complets des modèles existants, mais aussi l'un des plus longs. Les questions sont là pour aider aux équipes à réfléchir et à planifier la vie d'un logiciel de la recherche, à y établir des objectifs, et à vérifier régulièrement si les objectifs ont été atteints, ou bien s'il y a de nouveaux objectifs. Vous y apporterez les réponses que vous souhaitez, il est, désormais, *votre plan de gestion de votre logiciel*. Ce qui est important est d'avoir les bonnes questions au bon moment, de ne pas oublier des points importants, de savoir traiter les questions relatives, par exemple, au droit d'auteur ou aux licences, ou bien celles relatives à la diffusion ou à la gestion du logiciel.

Point DMP OPIDoR. Le modèle PRESOFT a été inclus dans la liste des Plans de Gestion de données et de logiciels du service DMP OPIDoR (CNRS/INIST) entre avril 2018 et mars 2026. Pour des raisons de sécurité et de maintenance, ce modèle n'y est plus disponible⁸, mais on peut consulter les plans de gestion réalisés avec ce modèle et qui ont été publiés par leurs responsables, voir, par exemple, le modèle *Research Software Management Plan for SPPAS - the automatic annotation and analyses of speech*, B. Bigi (2024)⁹.

Point Document bilingue et formats (odt, pdf, markdown). Le modèle PRESOFT est un modèle bilingue [FR/EN] pour réaliser un plan de gestion de logiciels de la recherche. Il est disponible sur HAL et sur Zenodo. Une version markdown du modèle bilingue a été faite en 2023 par P. Clabaut (UGE)^{10, 11}.

Point Estimation de coûts et Financements. Il n'est pas nécessaire d'être trop précis, mais cela peut être utile de savoir dire, dans la question *A cette date (à préciser), quel est le coût estimé du développement du logiciel ?* de la section 2.1, qu'un logiciel a pris 6 mois/personne entre 2002 et 2004, coût qui peut être révisé ou suivi lors des nouvelles éditions du plan. La section 2.2 aidera également à y indiquer les financements reçus d'un ou plusieurs projets tout au long de la vie du logiciel. Si le logiciel n'a pas reçu de financement par projet, la section 2.2 du modèle PRESOFT restera vide, ou bien elle peut être utilisée pour y indiquer un objectif de demande de financement futur. Si le logiciel a reçu le financement de plusieurs projets, la section 2.2 peut être dupliquée pour y intégrer ces informations. Par exemple la section 2.2.1 donnera des informations sur un financement ANR et la section 2.2.2 donnera les informations sur un autre financement. Cette section 2.2 peut aussi être utilisée pour gérer les informations des stages (rémunérés ou pas) pour garder trace des informations financières ou des ressources.

7 <https://doi.org/10.5334/dsj-2024-043>

8 Similairement, le modèle des plans de gestion du Software Sustainability Institute (SSI) n'est plus disponible sur DMP OPIDoR.

9 https://dmp.opidor.fr/public_plans?page=1&search=Bigi

10 https://gitlab.univ-eiffel.fr/paul.clabaut/dmp_opidor_markdown/-/blob/main/README.md?ref_type=heads&plain=1

11 https://gitlab.univ-eiffel.fr/paul.clabaut/dmp_opidor_markdown/-/blob/main/README.md

Point Estimation des apports. La question de la section 2.1 *quel est le coût estimé du développement du logiciel ?* peut être complétée avec la question *quels sont les apports du logiciel ?* En effet, le logiciel a pu apporter des financements, des collaborations, des nouveaux projets, des nouvelles questions scientifiques et des résultats publiés. Savoir évaluer ces apports, ou bien essayer de prévoir certains objectifs (scientifiques ou autres) avec le logiciel fait partie des réflexions à mener.

Point Traduction, par exemple de C++ à Python. L'auteur de la première version est aussi auteur de la deuxième version, et il faut un accord (ou bien une licence de logiciel libre) pour vérifier le contexte légal qui permet de faire cette traduction. Les auteurs de la deuxième version devront associer un certain % des droits d'auteur aux auteurs de la première version. Voir la présentation « *Brève introduction aux questions juridiques : droit d'auteur et licences* »¹² pour plus d'information sur les questions juridiques.

Point Équipe scientifique et équipe de développement. Rôles. Il arrive souvent qu'un logiciel est développé et maintenu par une seule personne qui accumule tous les rôles et responsabilités, ou bien une petite équipe qui concentre l'activité scientifique et l'activité de développement sans y faire distinction. Mais il y a aussi des développements avec des équipes bien organisées, avec des rôles différents, avec peut-être une gouvernance, une direction scientifique, des responsables du développement, de la documentation, etc. Et le plan permet d'identifier les différents rôles, un fonctionnement déjà en cours qui peut être amélioré, ou bien une organisation à mettre en place. Par exemple, un logiciel développé par une seule personne peut évoluer rapidement vers un développement avec une équipe qui intègre plusieurs personnes, avec des rôles différents, ou bien rester la responsabilité d'une seule personne pendant des longues années. Le plan permet le suivi de ces informations : elles peuvent être très simples ou bien complexes en fonction des évolutions du logiciel, de l'équipe de développement, de l'équipe scientifique. Le plan s'adapte aux situations diverses facilement.

Point Historique et vie du logiciel. Un plan de gestion de logiciels de la recherche peut se commencer à n'importe quel moment de la vie du logiciel, mais, évidemment, c'est plus facile de commencer le plan dans les étapes initiales du logiciel et une partie importante de questions restera sans réponse à ce moment-là. Elles auront, peut-être, une réponse au fur et à mesure des évolutions du logiciel. D'un autre côté, si le plan commence alors que l'historique du logiciel est trop complexe, il est possible qu'une bonne partie de l'information sur la vie du logiciel reste difficile de reconstituer, le plan est un lieu pour indiquer les informations connues et on peut y indiquer qu'il y a des informations manquantes correspondantes à une certaine période. Le plan est un document qui aide à centraliser certaines informations et d'assurer qu'elles ne seront pas « *perdues* ».

¹² http://igm.univ-mlv.fr/~teresa/presoft/2024FormationPlansGestionLogicielsRecherche/presentations/20240523_RSMP_juridique_TGD.pdf

Point Citation. Le modèle propose dans sa section 6.1 la question de la citation du logiciel. Il est important d'associer une formule de citation au logiciel de la recherche, surtout s'il est prévu une diffusion vers un public large. La présentation « *Les logiciels de la recherche dans le cadre de la Science ouverte...* »¹³ donne des indications et des références sur comment traiter ce point. Si vous souhaitez utiliser les Software Citation Files¹⁴ de S. Druskat, vous pouvez, par exemple, intégrer le fichier .cff dans la section 6.1.

Point Standards. Il peut être important de prendre en considération les standards utilisés ou bien à utiliser, et d'y indiquer qu'il faut se renseigner sur les standards ouverts existants, ou bien ceux développés dans une certaine communauté. Ces informations sont à gérer dans la section 3.3 du modèle PRESOFT.

Point Flexibilité. Le plan est un document flexible qui peut s'initier à n'importe quel moment de la vie du logiciel et qui s'adapte aux besoins de l'équipe, pour y renseigner les informations qu'on considère les plus importantes, pour établir des nouveaux objectifs. Le plan s'adapte à vos objectifs, au traitement que vous souhaitez faire de ce document. Il est, désormais, *votre plan de gestion de votre logiciel.*

Point Révisions. Le plan est un document à réviser régulièrement, par exemple une fois par an. Une nouvelle version du plan peut se faire lorsqu'il y a un nouveau financement, un nouveau membre de l'équipe, un départ, une nouvelle version importante du logiciel... ou bien lorsque l'équipe décide qu'il n'est pas prévu des nouvelles versions. Le plan sera révisé à nouveau si cette décision change.

Point Aide à la décision. Le plan peut être utilisé pour y renseigner des informations, des stats (downloads...), qui peuvent faciliter des décisions comme mettre plus ou moins de ressources pour continuer ou pas le développement, assurer la maintenance, pour montrer à la direction du laboratoire (ou aux tutelles) le sérieux d'un projet, pour y inscrire des informations qui vont faciliter la demande des nouveaux financements. Le plan peut garder une trace des décisions prises, ou bien de celles qui sont à considérer lors des éditions suivantes du plan.

Point Gestion des informations du droit d'auteur. Le plan est, entre autres, un outil pour faire le suivi des différents participants, auteurs et/ou contributeurs, pour réfléchir au besoin/envie d'avoir des participations externes à l'équipe, existantes ou à venir, et pour réfléchir sur la gestion des collaborations scientifiques, au développement. Cela peut être important de se poser ces questions. Parfois il n'aura pas une réponse précise à un certain moment, mais la prochaine version du plan pourra peut-être intégrer des nouvelles réponses ou des nouveaux objectifs.

13 http://igm.univ-mlv.fr/~teresa/presoft/2024FormationPlansGestionLogicielsRecherche/presentations/20240523_RSMP_RS_TGD.pdf

14 <https://citation-file-format.github.io/>

Point Gestion des versions du plan. Le plan peut être diffusé avec la documentation du logiciel, dans sa forme complète ou en partie, selon les informations qui sont considérées comme publiques ou sensibles. Les versions peuvent être gérées de la même façon que pour n'importe quel autre fichier qui fait partie du logiciel.

Point Informations publiques et/ou privées. Le plan peut contenir des informations sensibles, sur le financement, des données personnelles, sur les planifications de l'activité scientifique ou du développement, avec des informations stratégiques. Il faut réfléchir sur les parties qui seront publiques ou sur celles qui resteront privées.

Point Images. Les versions odt ou markdown vous permettront l'intégration des images.

Remerciements. Les échanges avec les participants et les intervenants des deux formations mentionnées ont été très riches et formateurs, pour les participants mais aussi pour les intervenants. Certaines questions reviennent régulièrement (les licences...), d'autres nous ont aidés à réfléchir et à mieux saisir certains aspects relatifs à cet objet complexe, les logiciels de la recherche, et aux plans de gestion qui peuvent aider aux équipes qui produisent ces objets. Je remercie chaleureusement toutes les personnes qui ont participé à ces échanges et qui les ont faits possibles. En particulier la formation de 2026 a reçu le financement du CNRS via une Action nationale du réseau DevLOG. Les deux formations ont été possibles grâce aux financements et au support de l'Université Gustave Eiffel et du Laboratoire d'informatique Gaspard-Monge.