Identification de clés pour le succès de projets de gestion informatisée de données environnementales à partir du logiciel Collec-Science

Eric Quinton [1], **Christine Plumejeaud-Perreau** [2], **Sylvie Damy** [3]

1. *INRAE - Unité de recherche Écosystèmes aquatiques et changements globaux*
   50, avenue de Verdun
   33612 CESTAS, France
   eric.quinton@inrae.fr

2. *Littoral Environnement et Sociétés, U.M.R. 7266*
   2 rue Olympe de Gouges
   17000 La Rochelle, France
   christine.plumejeaud-perreau@univ-lr.fr

3. *Université de Bourgogne Franche-Comté – U.M.R. 6249 – Laboratoire Chrono-environnement*
   16 route de Gray
   25030 Besançon cedex, France
   sylvie.damy@univ-fcomte.fr

RÉSUMÉ. *Dans le cadre d'une recherche qui s'ouvre aux pratiques de l'open-data pour la diffusion de données de qualité, il devient plus que jamais nécessaire de doter le monde académique des « bons » outils pour cet objectif ambitieux. Nous parlons de logiciels dédiés à la gestion qui peuvent être soit créés dans le laboratoire où ils sont utilisés, soit diffusés vers une communauté de chercheurs, ou encore pris « sur étagère » dans le catalogue des logiciels libres. A travers cette contribution qui s'appuie sur la grille d'analyse Business Model Canvas, nous tentons d'identifier les points clés de la réussite de tels projets. Nous soulignons notamment la nécessité d'une estimation réaliste des moyens requis pour la diffusion et l'appropriation de tels logiciels et des différents profils de personnels nécessaires à la réalisation de ces projets, avec un focus particulier sur le rôle de curateur / administrateur / animateur des données dans les laboratoires, la nécessité de proposer une offre d'hébergement et la structuration de l'appui.*

ABSTRACT. *In the context of research that is opening up to open-data practices for the dissemination of quality data, it becomes more necessary than ever to equip the academic world with the right tools for this ambitious objective. We are talking about software dedicated to management that can either be created in the laboratory where it is used, or distributed to a community of*

researchers, or taken off the shelf in the open-source software catalogue. Through this contribution, which is based on the Business Model Canvas analysis grid, we try to identify the key points for the success of such projects. In particular, we underline the need for a realistic estimation of the means required for the dissemination and appropriation of such software and the different profiles of personnel needed to carry out these projects, with a particular focus on the role of curator / administrator / data manager in the laboratories, the need to propose a hosting offer and the structuring of support.

MOTS-CLÉS : *Génie Logiciel; Cycle de vie de la donnée - du logiciel ; Administrateur des données*

KEYWORDS: *Software Engineering; Data Life Cycle - Software; Database administrator*

1. Introduction

En juillet 2018, la Ministre de l'Enseignement Supérieur, de la Recherche et de l'Innovation a présenté le plan national pour la science ouverte [1]. L'idée principale de ce plan est la diffusion sans entrave des publications et données issues de recherches financées sur fonds publics. Il propose de structurer et d'ouvrir les données de la recherche, en rendant obligatoire la diffusion ouverte de celles qui sont issues de programmes financés par appels à projets sur fonds publics, en créant la fonction d'administrateur des données et le réseau associé au sein des établissements. Ce plan crée également les conditions pour promouvoir l'adoption d'une politique de données ouvertes associées aux articles publiés par les chercheurs. Actuellement, si certaines disciplines telles que l'astronomie ont déjà une expérience reconnue dans la gestion des données, dans le domaine environnemental où les données ne sont pas forcément très volumineuses mais très hétérogènes (*The long tail data* [2]), la gestion des données reste bien souvent à la charge des structures de recherche telles que les laboratoires.

Les données environnementales sont produites dans un but précis mais peuvent être utilisées pour un autre. La problématique de la qualité, de l'interopérabilité et de la traçabilité de celles-ci est ainsi très forte. La production de logiciels pour les gérer est essentielle afin d'aider les chercheurs à maîtriser au mieux leurs données. Ils sont à forte valeur ajoutée, mais avec un public potentiellement restreint et très spécialisé. Les laboratoires cherchent à embaucher des développeurs pour concevoir de nouveaux outils, et de nombreuses offres sont proposées dans les réseaux spécialisés universitaires. Cela montre l'importance de ce besoin mais aussi le manque de ressources humaines dans ce domaine.

1. https://www.ouvrirlascience.fr/plan-national-pour-la-science-ouverte/
PLAN NATIONAL POUR LA SCIENCE OUVERTE

2. https://www.rd-alliance.org/groups/long-tail-research-data-ig/wiki/plenary-3-long-tail-research-data.html - RDA : Plenary 3 - the long tail of research data

Une alternative à cette création est l'utilisation de logiciels créés par d'autres structures. L'État français, en encourageant la publication des codes en *Open-Source* et en portant à la connaissance du public les différentes réalisations, permet à de nombreux laboratoires de s'appuyer sur des produits tiers à moindre coût. Toutefois, il est illusoire de penser que la simple installation d'un logiciel soit suffisante pour garantir son usage sur le long terme. Mendoza *et al*. (2010) ont identifié trois phases essentielles pour utiliser un logiciel dans ce contexte. La première consiste en une phase exploratoire, pour identifier les produits susceptibles de répondre au besoin. La seconde vise à le tester, voire à l'adapter. Ce n'est qu'à la fin de ces étapes qu'il pourra être utilisé en conditions réelles (*cf.* figure 1). Enfin, une fois que l'appropriation est terminée, le logiciel est utilisé en « routine ».

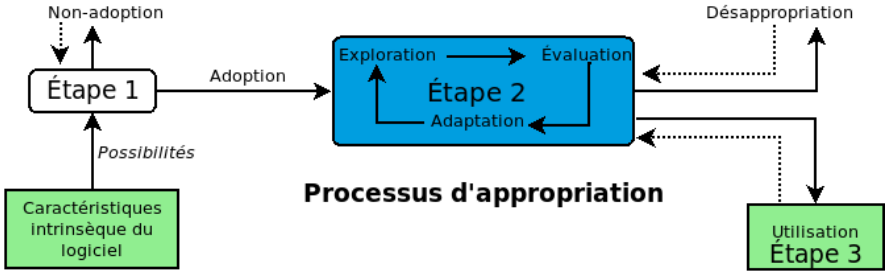

Figure 1. Processus d'appropriation des logiciels (d'après Mendoza et al. *(2010)).*

À chaque étape, différents freins peuvent amener à abandonner soit le choix du logiciel initial, soit le projet lui-même. Les raisons évoquées ont été étudiées : participation insuffisante de l'utilisateur final, mauvaise spécification des besoins, mauvaise estimation des coûts, relations entre les intervenants, fonctionnement de l'équipe de projet, politique de l'entreprise en contradiction avec le projet, etc. (Eberendu, 2015).

Le logiciel Collec-science [3] qui a été créé pour gérer les échantillons prélevés dans le cadre des campagnes scientifiques, a, dès le début du projet, été conçu pour être diffusé auprès d'autres laboratoires. Ainsi de nombreux travaux préparatoires ont été menés en partenariat avec le réseau des Zones Ateliers [4], notamment pour le choix de certaines technologies (choix des types de codes-barres, des matériels, identification des besoins complémentaires, etc.)(Quinton *et al*., 2018). Il a été déployé notamment dans le cadre des Zones-Ateliers. De nombreuses actions ont été nécessaires pour faciliter son déploiement et son appropriation par les différentes unités de recherche intéressées. Un groupe de travail a été constitué et a été largement supervisé par une personne du groupe (Plumejeaud-Perreau *et al*., 2019). Fin 2019, on peut estimer à un vingtaine le nombre de laboratoires de recherche qui soit l'utilisent en routine (étape 3, *cf.* figure 1), soit l'évaluent (étape 2). Cette diffusion n'aurait pas pu voir le jour

3. https://www.collec-science.org

4. Réseau interdisciplinaire de laboratoires français sur l'environnement et les socio-écosystèmes https://inee.cnrs.fr/fr/zones-ateliers

sans la mobilisation d'un certain nombre d'acteurs et la réalisation d'actions (formation, création d'un site web, etc.). Celles-ci ont été mises en place au fur et à mesure des besoins ressentis par l'équipe de diffusion et en fonction des retours des différents utilisateurs potentiels. Leur identification devrait permettre de mieux cerner les conditions qui facilitent l'appropriation d'un logiciel dans un contexte de laboratoire de recherche.

Pour identifier les facteurs de réussite de la diffusion du logiciel Collec-Science, nous avons cherché à répondre à la question suivante : quels sont les acteurs à mobiliser et les tâches à réaliser ? Pour cela, nous avons utilisé un outil largement répandu dans le monde des affaires, le *Business Model Plan* (Osterwalder, Pigneur, 2011). Basé sur un schéma composé de neuf cases, il s'adapte bien à l'étude que nous voulions mener. Nous expliciterons ce choix dans la partie *méthodologie*. Trois cas d'utilisation ont été étudiés : d'une part, la réalisation du logiciel pour un usage interne, d'autre part, sa mise en place dans un laboratoire qui ne l'a pas conçu, et enfin, sa diffusion à d'autres partenaires.

2. Méthodologie

2.1. Business Model Canvas

Pour identifier le processus de conception et de mise en œuvre d'un logiciel, nous avons choisi de nous appuyer sur le *Business Model Canvas*, créé par Alexander Osterwalder (Osterwalder, 2004). Ce modèle est plutôt dédié au monde économique. Cette modélisation des processus n'est pas la première proposée, mais son approche synthétique, basée sur un tableau composé de 9 cases, comme le montre la figure 2, permet d'appréhender et de représenter sur une seule page toutes les tâches, les coûts et les bénéfices attendus d'un projet. Il a d'ailleurs été utilisé dans le cadre de développements open-source de produits matériels (Fjeldsted *et al*., 2012). Luoma *et al*. (2012), dans une étude sur le modèle économique des plates-formes de type SAAS (*Software as a service*), considère que ce modèle est un des plus utilisés pour analyser les sociétés en technologie de l'information. Sa flexibilité nous a permis de l'adapter facilement au contexte de notre étude. Il est d'autre part assez simple d'utilisation, très synthétique et surtout nous permet de bien montrer aux directions des structures de recherche ce qu'il nous faut en termes de ressources humaines pour mener à bien de tels projets. Le tableau 1 récapitule l'ensemble des rubriques du canevas et la signification que nous lui avons donné.

Dans la pratique, le tableau est construit en positionnant des Post-it© sur un canevas affiché au mur , qui est retranscrit ensuite dans un format numérique.

2.2. Les études de cas

La démarche pour mettre en œuvre un logiciel n'est pas tout à fait la même selon qu'il est créé en interne, ou pris « sur étagère ». Dans le premier cas, il faut disposer

Tableau 1. Liste des rubriques disponibles dans un Business Model Canvas et signification dans le contexte de l'étude

Rubrique	Description
Offre	Produit créé et production de valeur attendue
Partenaires clés	Principaux partenaires mobilisés
Activités clés	Principales activités à mettre en œuvre pour produire l'offre
Ressources clés	Principales ressources (humaines ou non) à mobiliser pour réaliser les activités clés
Segments de clientèle	Cible à qui est destinée l'offre
Canaux de distribution	Moyens à mettre en place pour faire connaître l'offre
Relation client	Mécanismes permettant au segment de clientèle de faire remonter ses besoins ou de lui apporter les compléments nécessaires (formation, par exemple)
Structure des coûts	Coûts financiers ou humains à mobiliser pour arriver à l'objectif
Sources de revenus	Gains soit financiers, soit en efficacité, soit en réputation atteints par la diffusion de l'offre

des ressources internes pour la définition des besoins et le codage, mais cela permet d'obtenir une solution proche de la demande initiale. De plus, la formation des utilisateurs est aisée, le développeur étant disponible pour répondre aux questions. Dans le second cas, si l'étape du développement est ignorée, d'autres tâches apparaissent. Il faut s'approprier la structure et les buts du logiciel, le déployer, le configurer, se former et former les utilisateurs finaux, etc. Les ressources à mobiliser sont différentes mais toutes aussi indispensables pour la réussite du projet. Enfin, quand un laboratoire décide de proposer son logiciel à la communauté, il ne suffit pas de le publier pour qu'il soit utilisé. La protection du code, le choix de la licence de diffusion n'en sont que la première étape. Il faut ensuite rédiger des documentations, voire créer des sites vitrines, avoir une stratégie de communication, faciliter le déploiement, etc. Pour ces trois cas, connaître l'ensemble des acteurs à mobiliser et des tâches à exécuter devient nécessaire, pour d'une part faciliter la prise de décision initiale, et d'autre part dimensionner correctement les moyens à allouer.

Nous proposons l'étude de la conception et de la mise en place d'un logiciel de gestion de données dans trois cas différents, à l'aide des *Business Model Canvas*. La première modélisation traitera du développement et de l'utilisation du logiciel *Otolithe* au sein du laboratoire EABX [5]. La seconde s'attachera à identifier les acteurs et les tâches impliqués dans l'installation et l'utilisation du logiciel *Collec-Science* dans le laboratoire de Chrono-Environnement. Enfin, la dernière présentera l'ensemble des

5. laboratoire Écosystèmes Aquatiques et Changements Globaux – INRAE – CESTAS (Gironde)

tâches et des acteurs impliqués à la fois dans la conception et la diffusion du logiciel *Collec-Science*.

Le logiciel Collec-Science a, dès le début de sa conception, été prévu pour être diffusé dans d'autres laboratoires : s'appuyer sur celui-ci pour présenter les tâches correspondant au premier cas d'utilisation aurait soit été théorique, soit aurait présenté un biais (contacts nombreux avec d'autres laboratoires potentiellement intéressés).

3. Modélisation des cas d'utilisation

3.1. Création du logiciel et utilisation en interne dans le laboratoire

Le *Business Model Canvas* présenté en figure 2 correspond au logiciel Otolithe, un outil développé et déployé au sein du laboratoire EABX pour faciliter la lecture des stries de croissance sur les pièces calcifiées de poissons (otolithes, écailles, sections de rayons de nageoire). Les deux logiciels, utilisés comme supports dans cet article, sont très proches à la fois sur le plan technique (mêmes langages de programmation, mêmes outils, mêmes moteurs de bases de données) et sur le plan de la réalisation : ils ont été écrits par le même développeur, dans le même laboratoire. On peut considérer que les résultats obtenus pour Otolithe sont assez équivalents à ceux qui auraient été acquis dans le cas où Collec-Science n'aurait pas été destiné à être diffusé.

Business Model Canvas – Logiciel Otolithe

Partenaires clés	*Activités clés* • Fourniture des moyens (A) • Développement du logiciel Otolithe (B) • Administration de la plate-forme de production, sauvegarde (C)	*Offre (proposition de valeur)* • Lecture des pièces calcifiées : o gagner du temps o Fiabiliser la lecture o gagner en traçabilité	*Relation client* • Formation et appui vers les administrateurs métiers (B) • Formation des utilisateurs (E) • Assistance aux utilisateurs (E) • Gestion de tickets pour les demandes d'évolution et les bugs (B, E)	*Segments de clientèle* • (D) : Scientifiques et techniciens procédant aux lectures des pièces • (E) : techniciens et scientifiques préparant les photos et procédant à la saisie des données nécessaires à la lecture
	Ressources clés • (A) : E. Rochard – Directeur de l'UR – Chef de projet • (B) : E. Quinton – Admin BDD – développeur • (C) :J. Foury – Admin système		*Canaux de distribution* • Communication interne (A)	
Structure des coûts • Coûts de main d'œuvre		*Sources de revenus* • Gain de temps dans les activités de recherche • Fiabilisation des données récoltées		

Figure 2. Business Model Canvas de l'application Otolithe.

Le développement a été mis en place sous l'égide du directeur d'unité. Une fois l'application prête, elle a été mise en production par l'administrateur système. L'appli-

cation a été présentée aux utilisateurs lors de réunions, des formations ont été organisées pour les aider à la prendre en main. Un système de tickets, associé à la plate-forme de gestion de code, leur a été ouvert pour qu'ils puissent faire part des bogues rencontrés ou des évolutions qu'ils souhaitaient. Les gains de productivité pour les équipes techniques et pour les scientifiques chargés de la lecture des pièces calcifiées, la fiabilisation des données (stockage maîtrisé notamment) ont largement surpassé le temps d'écriture du logiciel.

3.2. Mise en place d'un logiciel conçu par un autre laboratoire

Le *Business Model Canvas* (figure 3) a été élaboré en reprenant les différentes tâches et les différents intervenants impliqués dans la mise en place du logiciel Collec-Science au sein du laboratoire de Chrono-environnement de l'université de Bourgogne-Franche Comté.

Business Model Canvas – Appropriation du Logiciel Collec-Science

Partenaires clés • (A) : E. Quinton – EABX IRSTEA • (B) : C. PLumejeaud UMR LIENSS/ RZA	*Activités clés* • Analyse des besoins (A,B,C,D,E) • Appui en développement (E) • Administration de la plate-forme de production (F) • Identification des personnes concernées par le logiciel au sein du laboratoire (E, C, D) • Animation auprès des techniciens en charge de la gestion des échantillons (E) • Définition des modèles d'échantillons (E, G1 et G2) *Ressources clés* • (C) : F. Raoul – directeur adjoint CE • (D) : S. Damy – responsable de l'axe transversal Données - Chef de projet • (E) : A. Maindron – développeur et animateur du projet • (F) : JD. Tissot et CH. Falconnet - Admin système • (G) : Ingénieurs / Techniciens gérant les échantillons (G1 : référents, G2 : utilisateurs)	*Offre (proposition de valeur)* Gestion des échantillons du laboratoire • D'un point de vue informatique : o centralisation des informations o accès facilité o gain de temps • D'un point de vue description des échantillons : o organisation o homogénéisation	*Relation client* • Formation des utilisateurs (E) • Assistance aux utilisateurs (E, G1, H et I) • Proposition de demandes d'évolution du logiciel en fonction des besoins des utilisateurs (E) *Canaux de distribution* • Communication interne (C,D,E) • Réunions organisées par l'animateur du projet (E) avec les techniciens (G1) et les utilisateurs finaux en consultation (G2, H, I)	*Segments de clientèle* • Ingénieurs/ Techniciens gérant les échantillons (stockage physique, répartition des échantillons dans les lieux de stockages, ...) (G) • (H) : Scientifiques ayant besoin de pouvoir rechercher facilement des échantillons • (I) : Gestionnaire du laboratoire pour la gestion des lieux de stockage (intervention en cas de panne, ...)
Structure des coûts • Coûts de personnels : o 6 mois d'Ingénieur d'études (E) o 10 % E/C sur 6 mois (D), 5 % E/C sur 6 mois (C) o 5 % à 15 % ingénieurs/techniciens sur 6 mois		*Sources de revenus* • Gestion des échantillons de qualité - Appui à la recherche : o optimisation du temps de gestion des échantillons o mutualisation et fiabilisation des données concernant les échantillons • Publications scientifiques (posters, articles, communications)		

Figure 3. Business Model Canvas correspondant à la mise en place de Collec-Science dans le laboratoire de Chrono-Environnement.

Compte-tenu de la nature des activités du laboratoire et de l'organisation des espaces de stockage, l'informatisation de la gestion des échantillons imposait une vision globale, et donc la mobilisation de l'ensemble des acteurs dès les premières étapes. Le projet a pu être porté par un animateur recruté pendant six mois, qui a en outre

collaboré au développement du logiciel et l'a fait évoluer en fonction des besoins spécifiques locaux. Le recours à des référents externes (le développeur et l'animatrice des Zones-Ateliers pour la gestion des échantillons) a été nécessaire pour faciliter la compréhension des concepts manipulés par le logiciel et appréhender sereinement son paramétrage. Bien que cela n'apparaisse pas dans le *Business Model Plan*, la cheffe de projet souhaiterait pouvoir basculer l'hébergement de l'application vers une plate-forme externalisée, les ressources informatiques au sein de son laboratoire rendant la gestion technique de celle-ci peu pérenne.

3.3. Diffusion du logiciel Collec-Science

Le troisième cas d'utilisation que nous avons étudié correspond à l'ensemble des tâches à réaliser pour diffuser le logiciel auprès d'autres laboratoires. Dès la publication des premières versions, nous avons ressenti le besoin de mettre en place des outils de communication et d'animation de la communauté des utilisateurs. En nous appuyant sur le livre « Logiciels et objets libres. Animer une communauté autour d'un projet libre » (Ribas *et al.*, 2016), nous avons mis en place un site web, des plates-formes de démonstration, des listes de diffusion, défini des règles d'organisation du développement, etc.

Malgré un appui fort réalisé par le réseau des Zones Ateliers, différents freins ont été identifiés, notamment en ce qui concerne la mise en place de la plate-forme technique d'hébergement du logiciel. Celui-ci nécessite la mise en place d'un serveur web Apache [6], des couches logicielles associées (PHP, JAVA, etc.) et d'un serveur de bases de données Postgresql [7]. Les premières versions nécessitaient la réalisation d'une suite d'opérations manuelles pour installer tous les composants, et peu de laboratoires disposaient en interne des ressources nécessaires pour réaliser ce travail, soit par manque de temps, soit par manque de compétences. Des scripts permettant une installation quasi-automatique du logiciel ont été mis au point. Deux stratégies différentes ont été mises en œuvre. La première permet de déployer une image dans un container Docker [8], la seconde déploie automatiquement les composants nécessaires dans un serveur Linux Debian [9]. La mise en place de ces scripts n'a pas permis de répondre à toutes les attentes : certains laboratoires n'avaient pas la capacité technique ou humaine suffisante pour déployer la solution en interne. Dans le cadre de l'animation des Zones Ateliers, une machine virtuelle a été louée auprès d'INRAE pour héberger les instances des laboratoires, membres d'une Zone Atelier, qui le souhaitaient.

L'ensemble des acteurs mobilisés et des tâches réalisées pour assurer le développement, la diffusion et la maintenance du logiciel sont récapitulés dans la figure 4.

6. http://httpd.apache.org/
7. https://www.postgresql.org/
8. https://github.com/jancelin/docker-collec
9. https://github.com/Irstea/collec/raw/master/install/deploy_new_instance.sh

Business Model Canvas – Logiciel Collec-Science

Partenaires clés

- (A) : C. PLumejeaud UMR LIENSS/RZA (S. Cipières, H. Linyer, O. Copi)
- (B) : S. Damy – UMR Chrono-environnement/ZA AJ (A. Maindron)
- (C) : W. Heintz – UMR Dynafor/ZA Pygar
- (D) : I. Billy, A. Caillo – UMR EPOC
- (E) : C. Pignol – UMR EDYTEM : ZA Alpes

Activités clés

- Définition du contour du projet (F, G)
- Développement du logiciel (G, A, B)
- Tests matériels : imprimantes, douchettes, étiquettes (A, D, E)
- Hébergement mutualisé (A, C)
- Documentation et vidéos (A, G)
- Gestion des instances de bases de données (A, C, I)
- Dépôt du code auprès de l'APP, choix de la licence (G)

Ressources clés

- (F) : E. Rochard – directeur UR EABX IRSTEA : Chef de projet
- (G) : E. Quinton – EABX IRSTEA : Admin BDD – développeur – responsable du développement
- (H) : G. Lambert -IRSTEA Bordeaux : web designer
- (I) : J. Foury – IRSTEA Bordeaux - Admin système

Offre (proposition de valeur)

- Logiciel Collec-Science :
  - o Fiabiliser le stockage des échantillons
  - o Assurer la traçabilité
- Guide d'achat du matériel et des consommables (imprimantes, étiquettes, douchettes, tablettes)
- Préconisations pour l'organisation de la gestion d'échantillons

Relation client

- Formation et appui vers les administrateurs métiers (A, G)
- Listes de diffusion (A, G)
- Gestion de tickets pour les demandes d'évolution et les bugs (G)
- Assistance de 3ème niveau (G)
- Assistance de 2nd niveau (A, G)

Canaux de distribution

- Réseau des zones ateliers (A)
- RBDD (A, G)
- Forge Github (G)
- Installation automatisée docker (A)
- site web vitrine (A, G, H)
- application de démonstration (A, G)

Segments de clientèle

- laboratoires de recherche gérant des échantillons stockés *ex situ*
- Zones Ateliers

Structure des coûts

- Coûts de personnels : 3 ans 40% IR LIENSS, 10 mois IE RZA, 6 mois IE Chrono-environnement, 4 mois stage RZA, 5 mois sur 3 ans d'ingénieur IRSTEA Bordeaux, 1 mois création du site web
- Frais de mission : participation à des congrès (IRSTEA, RZA), démonstrations (Chambéry, Besançon, Brest, Rennes, Grenoble, Nantes, Montpellier, Strasbourg, etc.)
- Achat de matériel de test (RZA, EPOC, IRSTEA)
- Frais de location d'un serveur mutualisé auprès de l'Inra (RZA)

(en souligné, les coûts internes au laboratoire EABX)

Sources de revenus

- qualité améliorée de la traçabilité des données associées
- optimisation du rangement, de la gestion des containers (frigos, armoires) et des stocks (péremption des échantillons, facilité de recherche, etc.)
- publications scientifiques (posters, articles, communications)
- financement de la recherche (qualité des propositions dans les appels d'offres)
- renommée

Figure 4. Business Model Canvas correspondant à la création et à la diffusion du logiciel Collec-Science

Compte-tenu de la complexité des interactions entre les différents intervenants et les actions menées, nous avons représenté dans la figure 5 les principales activités réalisées.

Ce diagramme met en exergue trois rôles essentiels : le développeur, le chef de projet pour la diffusion, et le responsable métier (ou le chef de projet) de l'organisme déployant le logiciel. Ces rôles sont génériques : ils peuvent être portés par une ou plusieurs personnes différentes. Le développeur s'occupe à la fois du codage, de la gestion des versions, du traitement des demandes d'améliorations ou des signalements de dysfonctionnements, de la rédaction de la documentation technique. Le chef de projet pour la diffusion prend en charge toute la partie communication, appui, formation, rédaction de la documentation à destination des utilisateurs, etc. Le responsable métier de l'organisme déployant le logiciel a un rôle de chef de projet pour sa mise en œuvre dans son laboratoire, en allouant ou en obtenant les moyens techniques, humains et

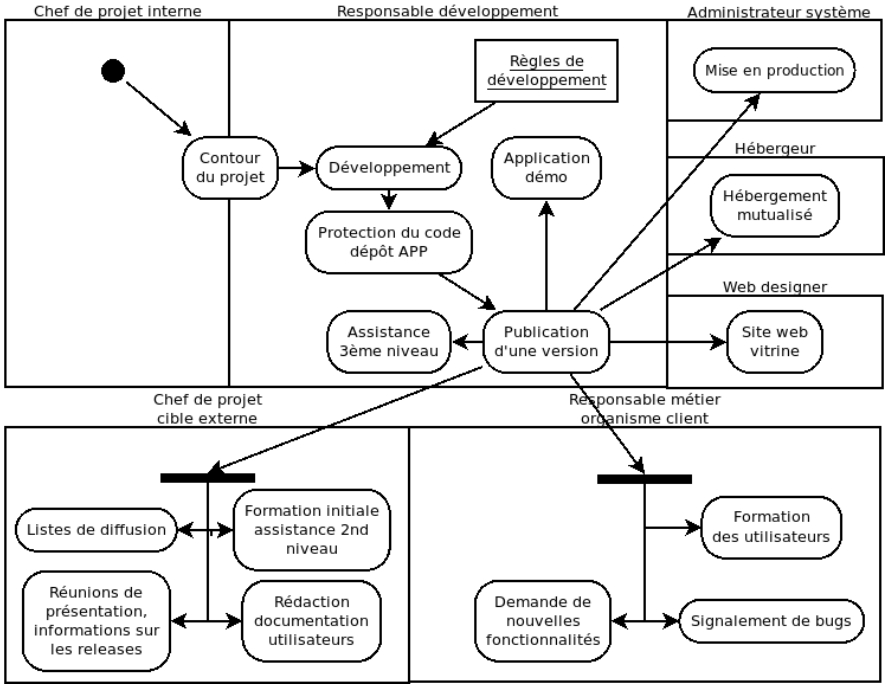

Figure 5. Diagramme d'activité représentant les acteurs impliqués et les tâches menées pour la création et la diffusion du logiciel Collec-Science

financiers nécessaires, en supervisant l'ensemble des opérations et en s'assurant de la qualité des données injectées dans le système.

4. Discussion

4.1. Identification des rôles à mobiliser

La notion de rôle est importante pour identifier clairement la position d'une personne dans une organisation. Plusieurs types de rôles peuvent être identifiés : axés sur les tâches, sur les relations ou sur l'autonomie (Zhu *et al.*, 2006). Nous nous attachons ici à identifier des rôles axés sur des tâches relevant du même domaine ou de même responsabilité. Quel que soit le mode de mise en œuvre d'un logiciel, qu'il soit développé en interne ou récupéré depuis un autre organisme, trois rôles sont absolument indispensables : un chef de projet, qui allouera les moyens adéquats et délimitera le projet, un administrateur de données / curateur de données / animateur du projet, qui le fera vivre au quotidien, et un administrateur des systèmes d'informations, qui s'occupera de gérer la plate-forme technique. Si cette dernière fonction ne peut être mobilisée, le recours à une offre d'hébergement devient indispensable.

Dans l'ensemble des laboratoires que nous avons accompagnés, le rôle du **curateur de données** nous a semblé prépondérant. Celui-ci doit disposer de compétences variées dans les domaines de la gestion de l'organisation (structuration et mise en place de collaborations, de politiques internes, etc.), du service aux utilisateurs (accès et réutilisation des données, formation, etc.) et de la gestion des données proprement dites ou de la technologie (Tammaro *et al*., 2014). Sa connaissance profonde de tous les aspects liés aux données en font un interlocuteur essentiel pour mettre en place une gestion des données, et particulièrement dans le domaine des collections, celui que nous avons traité. Toutefois, cette fonction est peu présente dans les unités de recherche. En novembre 2019, sur 39 postes ouverts au concours du CNRS pour la filière *informatique, statistiques et calcul scientifique*, un seul concernait la fonction d'administrateur de bases de données, contre 15 en ingénierie logicielle et 22 en calcul scientifique. Pourtant, les besoins dans ce domaine augmentent régulièrement, notamment avec les problématiques liées à l'*Open Data*, la mise en place de plans de gestion de données qui sont de plus en plus demandés dans le cadre des projets de recherche H2020 européens [10], ou pour répondre aux exigences du protocole de Nagoya [11].

Pour la diffusion d'un logiciel, deux rôles deviennent prédominants : le développeur et le chef de projet de diffusion. Le **développeur** doit non seulement créer le logiciel, mais également assurer son suivi avec la gestion des tickets, la rédaction de la documentation, l'assistance de troisième niveau, la mise en place de scripts de mise à jour, etc. Le **chef de projet de diffusion** a un rôle majeur en matière de formation des chefs de projets locaux, d'assistance de second niveau, de communication, etc. Son rôle se rapproche de celui des revendeurs de solutions dans le contexte du PLM (*Product Lifecycle Management*). Restuccia *et al*. (2016) ont identifié quatre rôles que peuvent jouer ces intermédiaires : informateurs de problèmes, conseillers en solutions, metteurs en œuvre de solutions, et gestionnaires de solutions. Dans la plupart des cas, le chef de projet de diffusion a eu un rôle de conseil, mais a pu aider parfois à implémenter la solution. Un recrutement de quelques mois d'un ingénieur dédié à cet aspect a d'ailleurs été réalisé par le réseau des Zones Ateliers.

Dans le tableau 2, nous avons cherché à identifier les différents rôles à mobiliser, en fonction du cas d'utilisation : écriture en interne d'un logiciel, création et diffusion d'un logiciel vers des structures tierces, et appropriation d'un logiciel fourni par une autre structure.

Plusieurs rôles peuvent être assumés par une même personne, certains pourraient être répartis au sein d'une équipe ou d'un collectif. Dans l'appui que nous avons apporté lors du déploiement de Collec-Science, nous avons identifié que l'absence d'une

10. http://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-data-management/data-management_en.htm

11. Protocole de Nagoya sur *l'accès aux ressources génétiques et le partage juste et équitable des avantages découlant de leur utilisation relatif à la convention sur la diversité biologique*, publié en 2012 (https://www.cbd.int/abs/doc/protocol/nagoya-protocol-fr.pdf)

Tableau 2. Rôles identifiés lors de la mise en œuvre d'un logiciel de gestion de données

Rôle	Description	Interne	Diffusion	Appropriation
Directeur	Ancrage du projet dans la pratique du laboratoire	X	X	X
Chef de projet	Pilotage du projet	X	X	X
Administrateur des données	Responsable de la qualité des données, animateur du projet	X	X	X
Responsable du développement	Supervision, mise en production des nouvelles versions et tâches liées à la protection du logiciel		X	
Développeur	Ajout de fonctionnalités, corrections	X	X	(X)
Administrateur système	Maintien en condition opérationnelle des plates-formes techniques	X	X	X
Hébergeur	Gestion de la plate-forme d'hébergement des instances des laboratoires clients		X	
Chef de projet diffusion	Toutes tâches liées à la mise en œuvre du logiciel dans des structures tierces		X	
Web designer	Conception du site web vitrine		X	
Responsables applicatifs	Utilisateurs avancés du logiciel	X		X
Utilisateurs	Utilisateurs du logiciel	X		X

personne pour tenir un des rôles était un facteur important de retard dans la mise en œuvre du projet, voire pouvait conduire à son échec.

4.2. Identification des tâches principales pour déployer un logiciel

Une fois que le laboratoire a décidé de mettre en œuvre un logiciel dont il n'a pas géré le développement, le projet commence en général par la mise à disposition des moyens techniques, humains et financiers nécessaires, puis par une communication vers les équipes locales, sous des formes variées. Le déploiement du logiciel proprement dit intègre des tâches de paramétrage, de formation des utilisateurs, d'assistance. Outre celles-ci, la gestion de la plate-forme technique apparaît comme une réelle problématique, à la fois pour sa mise en place, son administration (réalisation des sauvegardes, etc.) ou pour appliquer les mises à jour. Dans la pratique, peu de laboratoires disposent des ressources techniques suffisantes pour gérer ce type de plates-formes, ainsi des **solutions de mutualisation** deviennent quasiment incontournables. Mais, comme pour les solutions de type *Cloud* dont les risques ont été largement étudiés

(Neumann, 2014), des précautions doivent être prises pour garantir la sécurité des données et la réversibilité de l'hébergement (garantie de sauvegarde, base de données dédiée susceptible d'être récupérée intégralement, *p. e.*).

4.3. Identification des tâches principales impliquées dans la diffusion du logiciel Collec-Science

Outre les classiques opérations de communication (site web, interventions dans différents séminaires, etc.), nous avons identifié plusieurs tâches essentielles à la diffusion du logiciel, dans deux domaines principaux : l'appui aux chefs de projets locaux et la mise en place de solutions facilitant la diffusion technique.

Les chefs de projets locaux ont, dans la plupart des cas, réclamé un appui personnalisé, que nous avons apporté soit par l'intermédiaire de formations, soit par des échanges directs. La mise au point de **supports de communication** différents, comme des vidéos explicitant certains points d'utilisation particuliers, a été également appréciée.

Concernant la diffusion technique, la fourniture de **scripts d'installation quasi-automatiques** a été demandée très tôt. La mise en place d'une **plate-forme d'hébergement mutualisée** a été un facteur de facilitation fort.

Enfin, l'animation de la communauté des utilisateurs prend de plus en plus d'importance au fil du temps. En 2019, soit trois ans après la sortie de la première version du logiciel Collec-Science, un comité de pilotage du logiciel a été mis en place. Il est constitué de représentants de divers laboratoires soit en phase d'appropriation, soit en cours d'utilisation. Ce comité, organisé en groupes de travail sur différentes thématiques, doit faciliter l'organisation de formations, les échanges sur les pratiques et besoins des utilisateurs. Regroupant des personnes de différents laboratoires (répartis sur toute la France), il s'intéresse aussi à l'obtention de financements pour l'animation, le suivi du logiciel et des actions plus ponctuelles ou plus ciblées telle que la location de la plate-forme mutualisée.

4.4. Pérennisation du projet

La diffusion d'un logiciel Open-Source s'appuie sur deux rôles essentiels : le développeur et le chef de projet diffusion. La question de la pérennisation de ces deux rôles se pose. S'il est relativement facile de les mettre en place au début du projet, ils doivent s'inscrire dans la stratégie des structures qui les portent. Les investissements consentis pour diffuser le logiciel doivent pouvoir induire des bénéfices, immatériels ou non. Nous en avons identifié deux types. Le premier concerne l'amélioration du logiciel suite aux demandes des divers utilisateurs externes, les corrections de bogues ou les nouvelles fonctionnalités implémentées profitant directement au laboratoire d'ori-

gine [12]. Le second est un impact en terme de notoriété, soit par la publication d'articles ou la participation à des congrès, soit par la reconnaissance du laboratoire ou des personnes qui portent le projet, ce qui peut avoir un effet indirect pour l'obtention de programmes de recherche. Mais cela reste insuffisant. Dès lors que le logiciel se diffuse, d'autres moyens doivent être trouvés. En effet, le laboratoire d'origine vise l'innovation par la recherche et consacre ses forces à sa mission première et non pas à assumer la maintenance et la diffusion d'un logiciel. Lisein *et al*. (2009) ont étudié plusieurs sociétés éditant des logiciels Open-Source, et la question de leur pérennisation est au cœur de leurs préoccupations. Une des sociétés étudiées fonctionnait uniquement avec des subventions et du mécénat, et sa survie n'était pas assurée sur le long terme. La longévité d'un projet applicatif dans le domaine de la recherche reste donc une question encore délicate.

5. Conclusions et perspectives

L'utilisation d'un logiciel de gestion de données de recherche, conçu en interne ou par un autre laboratoire est un processus qui s'inscrit dans le mouvement pour la science ouverte. Elle permet de répondre aux besoins d'ouverture et de partage des données. Sa diffusion et sa réutilisation permettent des économies de moyens et d'échelle, mais cela impose la réalisation de tâches spécifiques et la mobilisation de profils particuliers, tant pour les concepteurs de l'application que pour ceux qui le mettent en œuvre. L'utilisation de la grille d'analyse *Business Model Canvas* nous a permis de définir clairement les coûts et les gains associés à ces processus, et d'identifier les rôles mobilisés. Cet exercice assez peu habituel dans la milieu de la recherche nous a permis de mettre en avant, vis à vis de nos structures d'accueil, l'apport et l'intérêt de ces logiciels qui, dans le contexte actuel d'ouverture et partage des données de la recherche, sont devenus indispensables, , mais aussi leurs coûts en termes de ressources humaines notamment.

Dans ce travail, nous avons identifié des rôles clés qui facilitent la mise en œuvre d'une informatisation de gestion de données : le chef de projet, le curateur de données / animateur du projet, l'administrateur des systèmes d'information. Quand le logiciel est diffusé, un nouveau rôle prépondérant émerge : le chef de projet pour la diffusion.

La diffusion d'un logiciel comme Collec-Science en *Open-Source* ne peut se suffire d'une simple publication du code dans une plate-forme ouverte. Elle impose la mise en place d'une plate-forme d'hébergement mutualisée, de la communication sur des supports divers, et des actions d'appui vers les animateurs locaux, soit sous forme de formations spécifiques, soit sous forme d'appui personnalisé. Si, jusqu'à présent, nous avons pu répondre aux besoins des laboratoires tiers dans leur mise en place du logiciel Collec-Science, nous nous interrogeons quant à la pérennisation de cet appui,

12. La dernière version de Collec-Science intègre une gestion de méta-données et une représentation cartographique de l'emplacement des échantillons d'une collection, deux demandes issues des laboratoires partenaires, et qui ont trouvé une utilisation immédiate dans les projets du laboratoire d'origine.

au moins à moyen terme, avec deux questions principales : comment garantir un hébergement de qualité, et comment apporter l'aide de second niveau ? Pour les logiciels commerciaux, ces tâches sont souvent prises en charge par les revendeurs, qui peuvent proposer des niveaux d'intervention plus ou moins importants (Restuccia *et al*., 2016). Dans le contexte de l'Open-Source, la gestion d'une communauté peut être une des réponses, mais celle-ci ne peut fonctionner qu'avec la présence d'acteurs et d'animateurs (Dupont *et al*., 2017), qu'il faut coordonner. L'attribution des responsabilités et le parrainage (avec les financements qui vont avec) sont des facteurs essentiels pour assurer la réussite d'un projet (Viseur, 2013).

Il n'est pas interdit d'envisager un recours à des structures privées pour gérer ces aspects. Si le choix d'une licence *Open-Source* pour diffuser le logiciel est impératif dans le contexte de la recherche publique (Quinton *et al*., 2018), diverses stratégies ont été proposées pour monnayer et rendre financièrement intéressant les logiciels (Shahrivar *et al*., 2018 ; Osterwalder, 2004 ; Osterwalder, Pigneur, 2011). Certaines pistes pourraient s'appliquer à nos cas de figure : facturation de l'hébergement pour les structures qui ne peuvent déployer la solution en interne, prestation d'appui au démarrage comprenant l'analyse des besoins propres et la configuration initiale, abonnement pour un accès à une maintenance de second niveau pour les responsables métiers, voire de premier niveau pour les utilisateurs, etc. Toutefois, la volumétrie des déploiements reste trop faible pour qu'une structure puisse envisager de ne travailler qu'avec un seul logiciel : c'est tout un écosystème d'appui qu'il faudrait mettre en place autour de différents produits.

Une autre solution serait de s'appuyer sur une unité de service d'un établissement de recherche assurant les missions d'hébergement et d'aide à la configuration. Pour Collec-Science, la plate-forme mutualisée mise en place à INRAE, et financée par un budget alloué par le réseau des Zones-Ateliers, rentre partiellement dans ce schéma.

Bibliographie

Dupont L., Gabriel A., Camargo M., Guidat C. (2017, juin). Collaborative Innovation Projects Engaging open communities: a Case Study on Emerging Challenges. In *23rd ICE/IEEE International Conference on Technology, Management and Innovation.* Madeira Island, Portugal. Consulté sur https://dx.doi.org/10.1109/ICE.2017.8280002

Eberendu A. C. (2015). *Evaluation of Software Project Failure and Abandonment in Tertiary Institutions in Nigeria.* IISTE. Consulté sur https://citeseerx.ist.psu.edu/viewdoc/download ?doi=10.1.1.909.3042&rep=rep1&type=pdf

Fjeldsted A., Adalsteinsdottir G., Howard T., McAloone T. (2012, janvier). Open Source Development of Tangible Products - from a business perspective. Consulté sur https://www.researchgate.net/publication/261365095_Open_Source_Development_of _Tangible_Products_-_from_a_business_perspective

Lisein O., Pichault F., Desmecht J. (2009). Les business models des sociétés de services actives dans le secteur Open Source. *Systemes d'information management*, vol. Volume 14, n° 2, p. 7–38. Consulté sur https://www.cairn.info/revue-systemes-d-information-et -management-2009-2-page-7.htm (Publisher: ESKA)

Luoma E., Rönkkö M., Tyrväinen P. (2012). Current Software-as-a-Service Business Models: Evidence from Finland. In M. A. Cusumano, B. Iyer, N. Venkatraman (Eds.), *Software Business*, p. 181–194. Berlin, Heidelberg, Springer Berlin Heidelberg.

Mendoza A., Carroll J., Stern L. (2010, mars). Software appropriation over time: from adoption to stabilization and beyond. *Australasian Journal of Information Systems*, vol. 16, n° 2. Consulté sur https://journal.acs.org.au/index.php/ajis/article/view/507

Neumann P. G. (2014, septembre). Risks and myths of cloud computing and cloud storage. *Commun. ACM*, vol. 57, n° 10, p. 25–27. Consulté sur http://doi.acm.org/10.1145/2661049

Osterwalder A. (2004). The Business Model Ontology – A Proposition in a Design Science Approach. *Doctorat en Informatique de Gestion. Ecole des Hautes Etudes Commerciales de l'Université de Lausanne.*. Consulté sur https://www.researchgate.net/publication/33681401_The_Business_Model_Ontology_-_A_Proposition_in_a_Design_Science_Approach

Osterwalder A., Pigneur Y. (2011). *Business Model nouvelle génération*. Pearson France. Consulté sur https://www.pearson.fr/FR/book/?GCOI=27440100730760

Plumejeaud-Perreau, Quinton E., Pignol C., Linyer H., Ancelin J., Cipière S. *et al*. (2019, août). Towards better traceability of field sampling data. *Computers & Geosciences*, vol. 129, p. 82–91. Consulté sur https://doi.org/10.1016/j.cageo.2019.04.009

Quinton E., Plumejeaud-Perreau C., Linyer H., Ancelin J., Pignol C., Cipière S. *et al*. (2018, mai). Sample management for scientific research with Collec-Science. In *INFORSID*, p. 41-61. Nantes, France. Consulté sur https://hal.archives-ouvertes.fr/hal-01825250 (INFORSID, Nantes, FRA, 28-/05/2018 - 31/05/2018)

Restuccia M., Brentani U. de, Legoux R., Ouellet J.-F. (2016). Product life-cycle management and distributor contribution to new product development. *Journal of Product Innovation Management*, vol. 33, n° 1, p. 69-89. Consulté sur https://onlinelibrary.wiley.com/doi/abs/10.1111/jpim.12261

Ribas S., Guillaud P., Ubeda S. (2016). *Logiciels et objets libres. animer une communauté autour d?un projet libre* (Framasoft, Ed.). Consulté sur https://framabook.org/logiciels-et-objets-libres/

Shahrivar S., Elahi S., Hassanzadeh A., Montazer G. (2018, novembre). A business model for commercial open source software: A systematic literature review. *Information and Software Technology*, vol. 103, p. 202–214. Consulté sur https://doi.org/10.1016/j.infsof.2018.06.018

Tammaro A. M., Ross S., Casarosa V. (2014). Research Data Curator: the competencies gap. In *BOBCATSSS 2014 Proceedings.* Consulté sur https://proceedings.bobcatsss2014.hb.se/article/view/327

Viseur R. (2013, juin). Identifying Success Factors for the Mozilla Project. In E. Petrinja, G. Succi, N. Ioini, A. Sillitti (Eds.), *9th Open Source Software (OSS)*, vol. AICT-404, p. 45-60. Koper-Capodistria, Slovenia, Springer. Consulté sur https://dx.doi.org/10.1007/978-3-642-38928-3_4 (Part 1: Full Papers - Innovation and Sustainability)

Zhu H., Zhou M., Seguin P. (2006, Nov). Supporting software development with roles. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, n° 6, p. 1110-1123. Consulté sur https://doi.org/10.1109/TSMCA.2006.883170